

Materi Bahasa Pemrograman

Ringkasan lintas bahasa: bahasa, runtime, toolchain, database, dan pola belajar.

Disertai 1000 soal latihan lintas ekosistem beserta kunci jawaban.

Isi dokumen
Materi ringkas disusun dari pembahasan sebelumnya, lalu diikuti paket latihan 1000 soal dan kunci jawaban.
Bahasa dan notasi dibuat sederhana, fokus pada rumus inti, konsep operasional, dan pola soal yang sering keluar.
Dokumen ini adalah bahan belajar mandiri, bukan naskah ujian resmi.

Bagian 1 - Peta besar ekosistem

- Daftar yang dibahas mencakup bahasa, runtime, package manager, build tool, dan database. Tidak semuanya adalah bahasa murni.

- Bahasa yang termasuk inti: Python, C, C++, Java, C#, Go, Swift, Kotlin, Ruby, Rust, JavaScript, TypeScript, R, Julia, Haskell, Scala, Clojure, Elixir, Erlang, OCaml, Scheme, PHP, Lua, Tcl, Nim, Crystal, dan assembly dengan NASM.

- Runtime dan tool yang perlu dibedakan: Node.js, .NET, Deno, Bun, npm, npx, Vite, Py launcher, serta SQLite sebagai database engine.

Fondasi universal di semua bahasa
Syntax dasar, variabel, tipe data, operator, kondisi, loop, fungsi, struktur data, string, file I/O, error handling, modularitas, testing, debugging, dan analisis performa.
Target pembelajaran terbaik: kuasai konsep yang sama lebih dulu, lalu terjemahkan ke sintaks tiap bahasa.

Bagian 2 - Ringkasan per keluarga bahasa

Bahasa inti untuk fondasi

- Python untuk produktivitas, automation, data, dan prototyping.
- C untuk memahami memori, pointer, dan kedekatan dengan mesin.
- SQL pada SQLite untuk data dan query. Bash untuk otomasi shell.

Bahasa industri umum

- JavaScript/Node.js dan TypeScript untuk backend, tooling, dan web modern.
- Java dan C#/.NET untuk sistem besar, API, desktop, dan enterprise.
- Go untuk layanan ringan dan concurrency yang sederhana.

Bahasa performa dan sistem

- C++ untuk performa tinggi dan kontrol yang fleksibel.
- Rust untuk keamanan memori plus performa native.
- Swift dan Kotlin untuk pengembangan modern di ekosistem Apple atau JVM/Android.

Bahasa data dan scientific

- R untuk statistik dan pelaporan. Julia untuk numerik dan komputasi ilmiah.
- Bahasa ini kuat ketika fokus utama adalah analisis data, model, atau eksperimen numerik.

Bahasa paradigma dan niche

- Haskell, Scala, Clojure, Elixir, Erlang, OCaml, dan Scheme melatih pola pikir functional dan data-oriented.
- Ruby, PHP, Lua, Tcl, Nim, Crystal, dan NASM berguna untuk domain tertentu, dari scripting hingga low-level.

Bagian 3 - Command run dan compile inti

Python: `py -3 main.py` | `python main.py`

C: `gcc main.c -o main.exe`

C++: `g++ main.cpp -o main.exe`

Java: `javac Main.java` lalu `java Main`

.NET: `dotnet new console -n App`, lalu `dotnet run`

Go: `go run main.go` | `go build`

Swift: `swift main.swift` | `swiftc main.swift -o main.exe`

Kotlin: `kotlinc Main.kt -include-runtime -d Main.jar` lalu `java -jar Main.jar`

Ruby: `ruby main.rb` | Rust: `cargo run` atau `rustc main.rs`

Node.js: `node app.js` | TypeScript: `tsc app.ts` lalu `node app.js`

Deno: `deno run main.ts` | Bun: `bun run main.ts`

R: `Rscript main.R` | Julia: `julia main.jl` | SQLite: `sqlite3 data.db`

Strategi belajar 30 / 60 / 90 hari
30 hari: kuasai Python, C, SQL, Bash, dan JavaScript dasar.
60 hari: tambah TypeScript, Java, C#/.NET, dan Go dengan mini project CRUD atau API sederhana.
90 hari: naik ke C++, Rust, Kotlin, Swift, R, Julia, lalu sampling bahasa functional dan niche.

Bagian 4 - Template belajar per bahasa

- Level 1: hello world, variabel, tipe data, kondisi, loop, fungsi, input-output.
- Level 2: string, list/map, file I/O, module/package, error handling, testing dasar.
- Level 3: OOP atau struct, generic/type system, concurrency/async, dependency management.
- Level 4: idiomatic style, debugging, profiling, build/release, arsitektur, dan library ecosystem.
- Mini project universal: kalkulator CLI, todo list file-based, parser teks, log parser, atau REST API sederhana.

Bagian 5 - 1000 soal bahasa pemrograman

Setiap soal berbentuk pilihan ganda dengan empat opsi. Gunakan bagian kunci jawaban di akhir dokumen untuk memeriksa hasil. Format pertanyaan dibuat singkat agar nyaman untuk latihan volume tinggi.

Subbagian: Python

1. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:
A. 10
B. 8
C. 6
D. 9
2. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:
A. 15
B. 17
C. 8
D. 16
3. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:
A. 25
B. 26
C. 10
D. 24
4. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:
A. 35
B. 36
C. 12
D. 37
5. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:
A. 50
B. 49
C. 14
D. 48
6. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:
A. 16
B. 63
C. 65
D. 64
7. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:
A. 5
B. 4
C. 4
D. 3
8. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:
A. 10
B. 9
C. 6
D. 8
9. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:
A. 16
B. 15
C. 17
D. 8
10. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:
A. 24
B. 10
C. 26

D. 25

11. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 12
- B. 37
- C. 36
- D. 35

12. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 50
- B. 14
- C. 48
- D. 49

13. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 16
- B. 64
- C. 63
- D. 65

14. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 5
- B. 3
- C. 4
- D. 4

15. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 6
- B. 8
- C. 10
- D. 9

16. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 17
- B. 15
- C. 16
- D. 8

17. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 10
- B. 26
- C. 24
- D. 25

18. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 35
- B. 36
- C. 12
- D. 37

19. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 49
- B. 50
- C. 48
- D. 14

20. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 16
- B. 65
- C. 64
- D. 63

21. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 5
- B. 4
- C. 3
- D. 4

22. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 6
- B. 8
- C. 9
- D. 10

23. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 15
- B. 17
- C. 16
- D. 8

24. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 25
- B. 10
- C. 26
- D. 24

25. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 12
- B. 37
- C. 35
- D. 36

26. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 48
- B. 14
- C. 50
- D. 49

27. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 64
- B. 65
- C. 16
- D. 63

28. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 4
- B. 3
- C. 4
- D. 5

29. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 10
- B. 8
- C. 9
- D. 6

30. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 15
- B. 8
- C. 16
- D. 17

31. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 24
- B. 26
- C. 10
- D. 25

32. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 35
- B. 12
- C. 37
- D. 36

33. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 49
- B. 50
- C. 48
- D. 14

34. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 65
- B. 63
- C. 16
- D. 64

35. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 3
- B. 4
- C. 5
- D. 4

36. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 9
- B. 6
- C. 8
- D. 10

37. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 17

- B. 15
- C. 8
- D. 16

38. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 25
- B. 26
- C. 24
- D. 10

39. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 37
- B. 36
- C. 35
- D. 12

40. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 50
- B. 14
- C. 48
- D. 49

41. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 65
- B. 63
- C. 64
- D. 16

42. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 5
- B. 4
- C. 3
- D. 4

43. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 9
- B. 8
- C. 6
- D. 10

44. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 16
- B. 8
- C. 17
- D. 15

45. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 25
- B. 10
- C. 26
- D. 24

46. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 12
- B. 35
- C. 36
- D. 37

47. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 14
- B. 49
- C. 48
- D. 50

48. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 16
- B. 63
- C. 64
- D. 65

49. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 4
- B. 3
- C. 5
- D. 4

50. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 8
- B. 10

- C. 9
- D. 6

51. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 16
- B. 15
- C. 8
- D. 17

52. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 24
- B. 25
- C. 26
- D. 10

53. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 35
- B. 37
- C. 12
- D. 36

54. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 48
- B. 50
- C. 14
- D. 49

55. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 16
- B. 63
- C. 64
- D. 65

56. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 4
- B. 4
- C. 5
- D. 3

57. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 6
- B. 9
- C. 10
- D. 8

58. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 8
- B. 15
- C. 17
- D. 16

59. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 24
- B. 10
- C. 25
- D. 26

60. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 37
- B. 35
- C. 12
- D. 36

61. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 50
- B. 48
- C. 14
- D. 49

62. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 63
- B. 16
- C. 64
- D. 65

63. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 5
- B. 4
- C. 4

D. 3

64. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 10
- B. 6
- C. 9
- D. 8

65. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 15
- B. 17
- C. 8
- D. 16

66. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 24
- B. 10
- C. 25
- D. 26

67. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 35
- B. 37
- C. 12
- D. 36

68. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 48
- B. 50
- C. 14
- D. 49

69. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 63
- B. 64
- C. 65
- D. 16

70. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 4
- B. 5
- C. 3
- D. 4

71. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 8
- B. 9
- C. 6
- D. 10

72. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 16
- B. 15
- C. 8
- D. 17

73. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 26
- B. 24
- C. 25
- D. 10

74. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 36
- B. 12
- C. 35
- D. 37

75. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 49
- B. 48
- C. 50
- D. 14

76. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 63
- B. 65
- C. 16
- D. 64

77. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 3
- B. 5
- C. 4
- D. 4

78. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 6
- B. 9
- C. 8
- D. 10

79. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 8
- B. 15
- C. 17
- D. 16

80. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 25
- B. 26
- C. 24
- D. 10

81. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 36
- B. 37
- C. 12
- D. 35

82. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 48
- B. 49
- C. 50
- D. 14

83. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 63
- B. 16
- C. 65
- D. 64

84. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 3
- B. 4
- C. 5
- D. 4

85. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 9
- B. 10
- C. 8
- D. 6

86. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 8
- B. 15
- C. 16
- D. 17

87. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 10
- B. 25
- C. 24
- D. 26

88. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 37
- B. 36
- C. 12
- D. 35

89. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 49
- B. 14
- C. 48
- D. 50

90. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 65

- B. 16
- C. 64
- D. 63

91. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 4
- B. 4
- C. 3
- D. 5

92. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 10
- B. 9
- C. 6
- D. 8

93. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 17
- B. 16
- C. 8
- D. 15

94. Dalam Python, hasil dari ekspresi $5 * 5$ adalah:

- A. 25
- B. 24
- C. 10
- D. 26

95. Dalam Python, hasil dari ekspresi $6 * 6$ adalah:

- A. 35
- B. 36
- C. 12
- D. 37

96. Dalam Python, hasil dari ekspresi $7 * 7$ adalah:

- A. 50
- B. 48
- C. 14
- D. 49

97. Dalam Python, hasil dari ekspresi $8 * 8$ adalah:

- A. 63
- B. 16
- C. 64
- D. 65

98. Dalam Python, hasil dari ekspresi $2 * 2$ adalah:

- A. 3
- B. 4
- C. 5
- D. 4

99. Dalam Python, hasil dari ekspresi $3 * 3$ adalah:

- A. 8
- B. 9
- C. 6
- D. 10

100. Dalam Python, hasil dari ekspresi $4 * 4$ adalah:

- A. 17
- B. 8
- C. 16
- D. 15

Subbagian: C

101. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main1.exe adalah:

- A. javac main.c
- B. gcc main.c -o main1.exe
- C. python main.c
- D. g++ main.c -o main1.exe

102. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main2.exe adalah:

- A. javac main.c
- B. g++ main.c -o main2.exe
- C. python main.c
- D. gcc main.c -o main2.exe

103. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main3.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main3.exe
- D. gcc main.c -o main3.exe

104. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main4.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main4.exe
- D. gcc main.c -o main4.exe

105. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main5.exe adalah:

- A. g++ main.c -o main5.exe
- B. javac main.c
- C. gcc main.c -o main5.exe
- D. python main.c

106. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main6.exe adalah:

- A. gcc main.c -o main6.exe
- B. g++ main.c -o main6.exe
- C. python main.c
- D. javac main.c

107. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main7.exe adalah:

- A. javac main.c
- B. g++ main.c -o main7.exe
- C. gcc main.c -o main7.exe
- D. python main.c

108. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main8.exe adalah:

- A. javac main.c
- B. gcc main.c -o main8.exe
- C. python main.c
- D. g++ main.c -o main8.exe

109. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main9.exe adalah:

- A. gcc main.c -o main9.exe
- B. python main.c
- C. g++ main.c -o main9.exe
- D. javac main.c

110. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main10.exe adalah:

- A. javac main.c
- B. python main.c
- C. gcc main.c -o main10.exe
- D. g++ main.c -o main10.exe

111. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main11.exe adalah:

- A. g++ main.c -o main11.exe
- B. gcc main.c -o main11.exe
- C. python main.c
- D. javac main.c

112. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main12.exe adalah:

- A. gcc main.c -o main12.exe
- B. javac main.c
- C. python main.c
- D. g++ main.c -o main12.exe

113. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main13.exe adalah:

- A. javac main.c
- B. gcc main.c -o main13.exe
- C. python main.c
- D. g++ main.c -o main13.exe

114. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main14.exe adalah:

- A. python main.c
- B. g++ main.c -o main14.exe
- C. gcc main.c -o main14.exe
- D. javac main.c

115. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main15.exe adalah:

- A. python main.c
- B. g++ main.c -o main15.exe
- C. javac main.c
- D. gcc main.c -o main15.exe

116. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main16.exe adalah:

- A. g++ main.c -o main16.exe

- B. javac main.c
- C. gcc main.c -o main16.exe
- D. python main.c

117. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main17.exe adalah:

- A. javac main.c
- B. gcc main.c -o main17.exe
- C. g++ main.c -o main17.exe
- D. python main.c

118. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main18.exe adalah:

- A. g++ main.c -o main18.exe
- B. javac main.c
- C. gcc main.c -o main18.exe
- D. python main.c

119. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main19.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main19.exe
- D. gcc main.c -o main19.exe

120. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main20.exe adalah:

- A. g++ main.c -o main20.exe
- B. javac main.c
- C. python main.c
- D. gcc main.c -o main20.exe

121. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main21.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main21.exe
- D. gcc main.c -o main21.exe

122. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main22.exe adalah:

- A. python main.c
- B. g++ main.c -o main22.exe
- C. gcc main.c -o main22.exe
- D. javac main.c

123. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main23.exe adalah:

- A. gcc main.c -o main23.exe
- B. python main.c
- C. g++ main.c -o main23.exe
- D. javac main.c

124. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main24.exe adalah:

- A. python main.c
- B. g++ main.c -o main24.exe
- C. gcc main.c -o main24.exe
- D. javac main.c

125. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main25.exe adalah:

- A. python main.c
- B. javac main.c
- C. gcc main.c -o main25.exe
- D. g++ main.c -o main25.exe

126. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main26.exe adalah:

- A. python main.c
- B. g++ main.c -o main26.exe
- C. javac main.c
- D. gcc main.c -o main26.exe

127. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main27.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main27.exe
- D. gcc main.c -o main27.exe

128. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main28.exe adalah:

- A. python main.c
- B. g++ main.c -o main28.exe
- C. javac main.c
- D. gcc main.c -o main28.exe

129. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main29.exe adalah:

- A. python main.c
- B. gcc main.c -o main29.exe

- C. javac main.c
- D. g++ main.c -o main29.exe

130. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main30.exe adalah:

- A. g++ main.c -o main30.exe
- B. javac main.c
- C. python main.c
- D. gcc main.c -o main30.exe

131. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main31.exe adalah:

- A. gcc main.c -o main31.exe
- B. g++ main.c -o main31.exe
- C. javac main.c
- D. python main.c

132. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main32.exe adalah:

- A. python main.c
- B. javac main.c
- C. g++ main.c -o main32.exe
- D. gcc main.c -o main32.exe

133. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main33.exe adalah:

- A. g++ main.c -o main33.exe
- B. python main.c
- C. gcc main.c -o main33.exe
- D. javac main.c

134. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main34.exe adalah:

- A. python main.c
- B. javac main.c
- C. g++ main.c -o main34.exe
- D. gcc main.c -o main34.exe

135. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main35.exe adalah:

- A. gcc main.c -o main35.exe
- B. python main.c
- C. javac main.c
- D. g++ main.c -o main35.exe

136. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main36.exe adalah:

- A. javac main.c
- B. gcc main.c -o main36.exe
- C. g++ main.c -o main36.exe
- D. python main.c

137. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main37.exe adalah:

- A. python main.c
- B. g++ main.c -o main37.exe
- C. javac main.c
- D. gcc main.c -o main37.exe

138. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main38.exe adalah:

- A. g++ main.c -o main38.exe
- B. python main.c
- C. gcc main.c -o main38.exe
- D. javac main.c

139. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main39.exe adalah:

- A. javac main.c
- B. g++ main.c -o main39.exe
- C. gcc main.c -o main39.exe
- D. python main.c

140. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main40.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main40.exe
- D. gcc main.c -o main40.exe

141. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main41.exe adalah:

- A. javac main.c
- B. python main.c
- C. gcc main.c -o main41.exe
- D. g++ main.c -o main41.exe

142. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main42.exe adalah:

- A. python main.c
- B. javac main.c
- C. gcc main.c -o main42.exe

D. g++ main.c -o main42.exe

143. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main43.exe adalah:

- A. python main.c
- B. javac main.c
- C. g++ main.c -o main43.exe
- D. gcc main.c -o main43.exe

144. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main44.exe adalah:

- A. gcc main.c -o main44.exe
- B. javac main.c
- C. g++ main.c -o main44.exe
- D. python main.c

145. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main45.exe adalah:

- A. gcc main.c -o main45.exe
- B. g++ main.c -o main45.exe
- C. python main.c
- D. javac main.c

146. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main46.exe adalah:

- A. javac main.c
- B. g++ main.c -o main46.exe
- C. python main.c
- D. gcc main.c -o main46.exe

147. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main47.exe adalah:

- A. gcc main.c -o main47.exe
- B. python main.c
- C. javac main.c
- D. g++ main.c -o main47.exe

148. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main48.exe adalah:

- A. javac main.c
- B. g++ main.c -o main48.exe
- C. gcc main.c -o main48.exe
- D. python main.c

149. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main49.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main49.exe
- D. gcc main.c -o main49.exe

150. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main50.exe adalah:

- A. python main.c
- B. gcc main.c -o main50.exe
- C. g++ main.c -o main50.exe
- D. javac main.c

151. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main51.exe adalah:

- A. g++ main.c -o main51.exe
- B. gcc main.c -o main51.exe
- C. javac main.c
- D. python main.c

152. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main52.exe adalah:

- A. javac main.c
- B. gcc main.c -o main52.exe
- C. python main.c
- D. g++ main.c -o main52.exe

153. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main53.exe adalah:

- A. javac main.c
- B. g++ main.c -o main53.exe
- C. python main.c
- D. gcc main.c -o main53.exe

154. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main54.exe adalah:

- A. g++ main.c -o main54.exe
- B. gcc main.c -o main54.exe
- C. python main.c
- D. javac main.c

155. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main55.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main55.exe
- D. gcc main.c -o main55.exe

156. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main56.exe adalah:

- A. g++ main.c -o main56.exe
- B. gcc main.c -o main56.exe
- C. python main.c
- D. javac main.c

157. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main57.exe adalah:

- A. python main.c
- B. gcc main.c -o main57.exe
- C. g++ main.c -o main57.exe
- D. javac main.c

158. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main58.exe adalah:

- A. g++ main.c -o main58.exe
- B. gcc main.c -o main58.exe
- C. python main.c
- D. javac main.c

159. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main59.exe adalah:

- A. python main.c
- B. javac main.c
- C. gcc main.c -o main59.exe
- D. g++ main.c -o main59.exe

160. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main60.exe adalah:

- A. javac main.c
- B. python main.c
- C. gcc main.c -o main60.exe
- D. g++ main.c -o main60.exe

161. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main61.exe adalah:

- A. javac main.c
- B. python main.c
- C. gcc main.c -o main61.exe
- D. g++ main.c -o main61.exe

162. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main62.exe adalah:

- A. python main.c
- B. g++ main.c -o main62.exe
- C. gcc main.c -o main62.exe
- D. javac main.c

163. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main63.exe adalah:

- A. javac main.c
- B. gcc main.c -o main63.exe
- C. python main.c
- D. g++ main.c -o main63.exe

164. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main64.exe adalah:

- A. javac main.c
- B. gcc main.c -o main64.exe
- C. g++ main.c -o main64.exe
- D. python main.c

165. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main65.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main65.exe
- D. gcc main.c -o main65.exe

166. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main66.exe adalah:

- A. gcc main.c -o main66.exe
- B. javac main.c
- C. g++ main.c -o main66.exe
- D. python main.c

167. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main67.exe adalah:

- A. javac main.c
- B. g++ main.c -o main67.exe
- C. python main.c
- D. gcc main.c -o main67.exe

168. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main68.exe adalah:

- A. python main.c
- B. javac main.c
- C. g++ main.c -o main68.exe
- D. gcc main.c -o main68.exe

169. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main69.exe adalah:

- A. javac main.c

- B. gcc main.c -o main69.exe
- C. python main.c
- D. g++ main.c -o main69.exe

170. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main70.exe adalah:

- A. g++ main.c -o main70.exe
- B. javac main.c
- C. python main.c
- D. gcc main.c -o main70.exe

171. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main71.exe adalah:

- A. gcc main.c -o main71.exe
- B. g++ main.c -o main71.exe
- C. javac main.c
- D. python main.c

172. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main72.exe adalah:

- A. gcc main.c -o main72.exe
- B. python main.c
- C. g++ main.c -o main72.exe
- D. javac main.c

173. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main73.exe adalah:

- A. gcc main.c -o main73.exe
- B. g++ main.c -o main73.exe
- C. python main.c
- D. javac main.c

174. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main74.exe adalah:

- A. python main.c
- B. g++ main.c -o main74.exe
- C. javac main.c
- D. gcc main.c -o main74.exe

175. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main75.exe adalah:

- A. python main.c
- B. g++ main.c -o main75.exe
- C. gcc main.c -o main75.exe
- D. javac main.c

176. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main76.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main76.exe
- D. gcc main.c -o main76.exe

177. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main77.exe adalah:

- A. python main.c
- B. javac main.c
- C. gcc main.c -o main77.exe
- D. g++ main.c -o main77.exe

178. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main78.exe adalah:

- A. javac main.c
- B. gcc main.c -o main78.exe
- C. g++ main.c -o main78.exe
- D. python main.c

179. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main79.exe adalah:

- A. gcc main.c -o main79.exe
- B. javac main.c
- C. g++ main.c -o main79.exe
- D. python main.c

180. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main80.exe adalah:

- A. javac main.c
- B. gcc main.c -o main80.exe
- C. g++ main.c -o main80.exe
- D. python main.c

181. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main81.exe adalah:

- A. python main.c
- B. javac main.c
- C. gcc main.c -o main81.exe
- D. g++ main.c -o main81.exe

182. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main82.exe adalah:

- A. gcc main.c -o main82.exe
- B. python main.c

- C. g++ main.c -o main82.exe
- D. javac main.c

183. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main83.exe adalah:

- A. g++ main.c -o main83.exe
- B. gcc main.c -o main83.exe
- C. python main.c
- D. javac main.c

184. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main84.exe adalah:

- A. python main.c
- B. gcc main.c -o main84.exe
- C. javac main.c
- D. g++ main.c -o main84.exe

185. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main85.exe adalah:

- A. gcc main.c -o main85.exe
- B. g++ main.c -o main85.exe
- C. python main.c
- D. javac main.c

186. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main86.exe adalah:

- A. g++ main.c -o main86.exe
- B. javac main.c
- C. gcc main.c -o main86.exe
- D. python main.c

187. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main87.exe adalah:

- A. javac main.c
- B. python main.c
- C. g++ main.c -o main87.exe
- D. gcc main.c -o main87.exe

188. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main88.exe adalah:

- A. javac main.c
- B. gcc main.c -o main88.exe
- C. python main.c
- D. g++ main.c -o main88.exe

189. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main89.exe adalah:

- A. javac main.c
- B. python main.c
- C. gcc main.c -o main89.exe
- D. g++ main.c -o main89.exe

190. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main90.exe adalah:

- A. g++ main.c -o main90.exe
- B. python main.c
- C. javac main.c
- D. gcc main.c -o main90.exe

191. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main91.exe adalah:

- A. python main.c
- B. g++ main.c -o main91.exe
- C. gcc main.c -o main91.exe
- D. javac main.c

192. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main92.exe adalah:

- A. gcc main.c -o main92.exe
- B. python main.c
- C. javac main.c
- D. g++ main.c -o main92.exe

193. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main93.exe adalah:

- A. gcc main.c -o main93.exe
- B. g++ main.c -o main93.exe
- C. python main.c
- D. javac main.c

194. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main94.exe adalah:

- A. g++ main.c -o main94.exe
- B. javac main.c
- C. python main.c
- D. gcc main.c -o main94.exe

195. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main95.exe adalah:

- A. gcc main.c -o main95.exe
- B. python main.c
- C. javac main.c

D. g++ main.c -o main95.exe

196. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main96.exe adalah:

- A. python main.c
- B. g++ main.c -o main96.exe
- C. gcc main.c -o main96.exe
- D. javac main.c

197. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main97.exe adalah:

- A. javac main.c
- B. gcc main.c -o main97.exe
- C. g++ main.c -o main97.exe
- D. python main.c

198. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main98.exe adalah:

- A. g++ main.c -o main98.exe
- B. gcc main.c -o main98.exe
- C. javac main.c
- D. python main.c

199. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main99.exe adalah:

- A. javac main.c
- B. g++ main.c -o main99.exe
- C. gcc main.c -o main99.exe
- D. python main.c

200. Perintah yang paling tepat untuk meng-compile file C bernama main.c menjadi main100.exe adalah:

- A. g++ main.c -o main100.exe
- B. python main.c
- C. javac main.c
- D. gcc main.c -o main100.exe

Subbagian: Java

201. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. run Main.java
- D. javac Main

202. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. node Main.java
- D. javac Main

203. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

204. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

205. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. node Main.java
- D. java Main

206. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. node Main.java
- D. run Main.java

207. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

208. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main

- C. javac Main
- D. node Main.java

209. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

210. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. node Main.java
- D. java Main

211. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. javac Main
- D. run Main.java

212. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. java Main
- D. node Main.java

213. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. node Main.java
- D. run Main.java

214. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. javac Main
- D. run Main.java

215. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. java Main
- D. javac Main

216. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. javac Main
- C. run Main.java
- D. java Main

217. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. node Main.java
- D. run Main.java

218. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. javac Main
- C. run Main.java
- D. java Main

219. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. javac Main
- D. run Main.java

220. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main
- C. javac Main
- D. node Main.java

221. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main
- C. node Main.java

D. javac Main

222. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

223. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. java Main
- D. run Main.java

224. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. javac Main
- C. java Main
- D. run Main.java

225. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. run Main.java
- D. javac Main

226. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. run Main.java
- D. javac Main

227. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. node Main.java
- D. java Main

228. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. run Main.java
- D. javac Main

229. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

230. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. node Main.java
- D. run Main.java

231. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. java Main
- D. node Main.java

232. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

233. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

234. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

235. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

236. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

237. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

238. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main
- C. node Main.java
- D. javac Main

239. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. javac Main
- D. java Main

240. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main
- C. node Main.java
- D. javac Main

241. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. javac Main
- D. java Main

242. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. java Main
- D. javac Main

243. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

244. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

245. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

246. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

247. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. run Main.java
- D. javac Main

248. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java

- B. node Main.java
- C. javac Main
- D. java Main

249. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

250. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

251. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. java Main
- D. run Main.java

252. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

253. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. java Main
- D. javac Main

254. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. javac Main
- D. run Main.java

255. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. javac Main
- D. run Main.java

256. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. javac Main
- D. run Main.java

257. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

258. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. node Main.java
- D. java Main

259. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. java Main
- D. javac Main

260. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

261. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java

- C. javac Main
- D. java Main

262. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

263. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

264. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

265. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

266. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. javac Main
- C. run Main.java
- D. java Main

267. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. run Main.java
- D. node Main.java

268. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. run Main.java
- D. javac Main

269. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. javac Main
- D. run Main.java

270. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. node Main.java
- D. run Main.java

271. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

272. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. java Main
- D. run Main.java

273. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

274. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. node Main.java

D. run Main.java

275. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. node Main.java
- D. run Main.java

276. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. java Main
- D. run Main.java

277. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. javac Main
- C. java Main
- D. node Main.java

278. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. java Main
- D. run Main.java

279. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. node Main.java
- C. run Main.java
- D. java Main

280. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. javac Main
- C. run Main.java
- D. java Main

281. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. java Main
- D. javac Main

282. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. javac Main
- D. java Main

283. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

284. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. java Main
- D. node Main.java

285. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

286. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. node Main.java
- D. run Main.java

287. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. run Main.java
- D. node Main.java

288. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. node Main.java
- D. java Main

289. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. run Main.java
- D. javac Main

290. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. javac Main
- D. run Main.java

291. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. node Main.java
- D. run Main.java

292. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. java Main
- C. javac Main
- D. run Main.java

293. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. node Main.java
- B. run Main.java
- C. javac Main
- D. java Main

294. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. run Main.java
- C. javac Main
- D. node Main.java

295. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. java Main
- C. javac Main
- D. node Main.java

296. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. run Main.java
- C. node Main.java
- D. java Main

297. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. run Main.java
- B. node Main.java
- C. java Main
- D. javac Main

298. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. javac Main
- C. run Main.java
- D. node Main.java

299. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. java Main
- B. node Main.java
- C. run Main.java
- D. javac Main

300. Untuk menjalankan program Java setelah file Main.java di-compile, perintah yang benar adalah:

- A. javac Main
- B. java Main
- C. node Main.java
- D. run Main.java

Subbagian: JavaScript and TypeScript

301. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. npm
- C. vite
- D. node

302. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. tsc app.js
- C. npm app.js
- D. node app.js

303. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. node
- C. vite
- D. tsc

304. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. bun build app.js
- C. node app.js
- D. tsc app.js

305. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. vite
- D. npm

306. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. node app.js
- C. tsc app.js
- D. bun build app.js

307. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. npm
- C. node
- D. vite

308. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. node app.js
- C. tsc app.js
- D. npm app.js

309. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. vite
- C. npm
- D. tsc

310. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. node app.js
- C. npm app.js
- D. bun build app.js

311. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. node
- D. tsc

312. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. tsc app.js
- C. bun build app.js
- D. npm app.js

313. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. vite
- C. npm
- D. tsc

314. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js

- B. node app.js
- C. npm app.js
- D. bun build app.js

315. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. node
- D. tsc

316. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. npm app.js
- C. node app.js
- D. bun build app.js

317. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. tsc
- C. node
- D. vite

318. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. node app.js
- C. tsc app.js
- D. bun build app.js

319. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. node
- D. tsc

320. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. bun build app.js
- C. tsc app.js
- D. node app.js

321. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. node
- C. npm
- D. tsc

322. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

323. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. tsc
- C. node
- D. npm

324. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. tsc app.js
- C. bun build app.js
- D. npm app.js

325. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. node
- C. npm
- D. vite

326. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. node app.js
- D. bun build app.js

327. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite

- C. tsc
- D. node

328. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. npm app.js
- C. bun build app.js
- D. tsc app.js

329. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. vite
- D. npm

330. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. npm app.js
- C. bun build app.js
- D. node app.js

331. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. vite
- C. npm
- D. node

332. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. bun build app.js
- C. npm app.js
- D. node app.js

333. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite
- C. tsc
- D. node

334. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. node app.js
- C. bun build app.js
- D. npm app.js

335. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. node
- C. tsc
- D. npm

336. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. bun build app.js
- C. tsc app.js
- D. node app.js

337. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. tsc
- C. node
- D. vite

338. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. node app.js
- D. bun build app.js

339. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. node
- C. npm
- D. vite

340. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. node app.js
- C. tsc app.js

D. npm app.js

341. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite
- C. tsc
- D. node

342. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. npm app.js
- C. bun build app.js
- D. tsc app.js

343. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. npm
- D. vite

344. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. npm app.js
- D. tsc app.js

345. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. tsc
- C. node
- D. npm

346. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. tsc app.js
- C. bun build app.js
- D. npm app.js

347. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. vite
- D. npm

348. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. bun build app.js
- C. node app.js
- D. npm app.js

349. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. vite
- D. npm

350. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

351. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. node
- D. tsc

352. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. npm app.js
- C. tsc app.js
- D. bun build app.js

353. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. npm
- C. tsc
- D. vite

354. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. bun build app.js
- C. npm app.js
- D. node app.js

355. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. vite
- C. npm
- D. tsc

356. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. node app.js
- C. bun build app.js
- D. npm app.js

357. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. node
- C. vite
- D. npm

358. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. bun build app.js
- C. node app.js
- D. npm app.js

359. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. tsc
- C. npm
- D. node

360. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. node app.js
- C. npm app.js
- D. tsc app.js

361. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. vite
- C. npm
- D. node

362. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

363. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. node
- D. tsc

364. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. npm app.js
- C. tsc app.js
- D. bun build app.js

365. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. node
- C. vite
- D. tsc

366. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

367. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc

- B. vite
- C. npm
- D. node

368. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. node app.js
- C. tsc app.js
- D. npm app.js

369. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. tsc
- C. node
- D. npm

370. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. bun build app.js
- D. node app.js

371. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite
- C. node
- D. tsc

372. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. tsc app.js
- C. npm app.js
- D. bun build app.js

373. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite
- C. tsc
- D. node

374. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. tsc app.js
- C. npm app.js
- D. node app.js

375. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. tsc
- D. node

376. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. node app.js
- C. bun build app.js
- D. tsc app.js

377. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. vite
- C. node
- D. npm

378. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. npm app.js
- C. tsc app.js
- D. node app.js

379. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. npm
- C. tsc
- D. node

380. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. tsc app.js

- C. bun build app.js
- D. npm app.js

381. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. vite
- C. node
- D. tsc

382. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

383. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. vite
- C. node
- D. npm

384. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. node app.js
- B. bun build app.js
- C. tsc app.js
- D. npm app.js

385. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. node
- C. vite
- D. tsc

386. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. node app.js
- D. bun build app.js

387. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. vite
- B. node
- C. npm
- D. tsc

388. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. npm app.js
- C. node app.js
- D. bun build app.js

389. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. tsc
- B. node
- C. vite
- D. npm

390. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. bun build app.js
- B. npm app.js
- C. tsc app.js
- D. node app.js

391. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. npm
- B. tsc
- C. node
- D. vite

392. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. node app.js
- C. bun build app.js
- D. tsc app.js

393. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. vite

D. npm

394. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. bun build app.js
- D. node app.js

395. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. npm
- C. vite
- D. tsc

396. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. node app.js
- D. bun build app.js

397. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. tsc
- C. npm
- D. vite

398. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. tsc app.js
- B. bun build app.js
- C. node app.js
- D. npm app.js

399. Agar file TypeScript app.ts menjadi JavaScript, tool yang dipakai adalah:

- A. node
- B. vite
- C. tsc
- D. npm

400. Perintah umum untuk menjalankan file JavaScript bernama app.js di Node.js adalah:

- A. npm app.js
- B. tsc app.js
- C. bun build app.js
- D. node app.js

Subbagian: .NET and C#

401. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. decimal[]
- D. bool

402. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. dotnet exec.cs
- C. csc run
- D. dotnet open

403. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. string
- D. decimal[]

404. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. dotnet open
- D. csc run

405. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. bool
- D. decimal[]

406. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run

- C. dotnet exec.cs
- D. dotnet run

407. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. decimal[]
- C. bool
- D. int

408. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. csc run
- D. dotnet exec.cs

409. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. string
- C. decimal[]
- D. bool

410. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet exec.cs
- C. dotnet open
- D. dotnet run

411. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. int
- C. string
- D. decimal[]

412. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet exec.cs
- C. dotnet run
- D. csc run

413. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. decimal[]
- C. bool
- D. string

414. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. csc run
- D. dotnet exec.cs

415. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. string
- C. decimal[]
- D. bool

416. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet exec.cs
- D. dotnet run

417. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool
- C. decimal[]
- D. int

418. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. dotnet exec.cs
- D. csc run

419. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. string
- C. int

D. bool

420. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. csc run
- D. dotnet open

421. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. bool
- C. int
- D. string

422. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet run
- D. dotnet exec.cs

423. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. int
- C. bool
- D. string

424. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. dotnet exec.cs
- D. csc run

425. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. decimal[]
- D. bool

426. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet run
- D. dotnet exec.cs

427. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. string
- D. decimal[]

428. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet exec.cs
- D. dotnet open

429. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. decimal[]
- C. string
- D. int

430. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. dotnet exec.cs
- C. csc run
- D. dotnet open

431. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. string
- D. decimal[]

432. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. csc run
- C. dotnet exec.cs
- D. dotnet open

433. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. int
- C. string
- D. bool

434. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. csc run
- C. dotnet open
- D. dotnet run

435. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. decimal[]
- C. bool
- D. string

436. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet open
- D. dotnet exec.cs

437. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. decimal[]
- C. string
- D. bool

438. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. dotnet exec.cs
- C. dotnet open
- D. csc run

439. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. bool
- D. decimal[]

440. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet exec.cs
- D. dotnet open

441. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. decimal[]
- C. int
- D. bool

442. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. csc run
- D. dotnet exec.cs

443. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool
- C. decimal[]
- D. int

444. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. dotnet exec.cs
- D. csc run

445. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. string
- C. bool
- D. int

446. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open

- B. dotnet run
- C. dotnet exec.cs
- D. csc run

447. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. int
- C. string
- D. bool

448. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet open
- C. dotnet run
- D. dotnet exec.cs

449. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. bool
- C. string
- D. int

450. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet exec.cs
- D. dotnet run

451. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. int
- C. bool
- D. string

452. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet exec.cs
- C. dotnet run
- D. csc run

453. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. decimal[]
- C. bool
- D. string

454. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet run
- D. dotnet exec.cs

455. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool
- C. int
- D. decimal[]

456. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet open
- D. dotnet exec.cs

457. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. decimal[]
- C. bool
- D. int

458. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. csc run
- C. dotnet open
- D. dotnet exec.cs

459. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool

- C. int
- D. decimal[]

460. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet open
- C. dotnet run
- D. dotnet exec.cs

461. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. decimal[]
- D. bool

462. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. dotnet open
- C. dotnet exec.cs
- D. csc run

463. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. decimal[]
- D. bool

464. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet open
- D. dotnet exec.cs

465. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. string
- D. decimal[]

466. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. dotnet open
- D. csc run

467. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. int
- C. decimal[]
- D. string

468. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. csc run
- C. dotnet run
- D. dotnet open

469. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. bool
- C. int
- D. string

470. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. csc run
- D. dotnet open

471. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. decimal[]
- D. string

472. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet exec.cs
- C. dotnet open

D. dotnet run

473. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool
- C. int
- D. decimal[]

474. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet exec.cs
- C. dotnet run
- D. csc run

475. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. bool
- C. decimal[]
- D. int

476. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet exec.cs
- D. dotnet open

477. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. bool
- C. int
- D. string

478. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet open
- C. dotnet run
- D. dotnet exec.cs

479. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. string
- C. bool
- D. int

480. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet open
- C. csc run
- D. dotnet run

481. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. int
- C. string
- D. decimal[]

482. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. csc run
- D. dotnet open

483. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. decimal[]
- C. bool
- D. string

484. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. csc run
- C. dotnet exec.cs
- D. dotnet open

485. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. string
- B. int
- C. bool
- D. decimal[]

486. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. dotnet exec.cs
- C. csc run
- D. dotnet open

487. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. bool
- C. int
- D. string

488. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet run
- B. csc run
- C. dotnet exec.cs
- D. dotnet open

489. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. decimal[]
- B. int
- C. bool
- D. string

490. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet exec.cs
- B. dotnet run
- C. csc run
- D. dotnet open

491. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. string
- C. decimal[]
- D. int

492. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet run
- C. dotnet exec.cs
- D. dotnet open

493. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. string
- C. bool
- D. decimal[]

494. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. csc run
- B. dotnet open
- C. dotnet run
- D. dotnet exec.cs

495. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int
- B. bool
- C. decimal[]
- D. string

496. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet run
- C. csc run
- D. dotnet exec.cs

497. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. bool
- B. string
- C. int
- D. decimal[]

498. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. dotnet exec.cs
- C. dotnet run
- D. csc run

499. Dalam C#, tipe data yang umum untuk bilangan bulat adalah:

- A. int

- B. string
- C. decimal[]
- D. bool

500. Di .NET CLI, perintah yang umum dipakai untuk menjalankan proyek console adalah:

- A. dotnet open
- B. csc run
- C. dotnet run
- D. dotnet exec.cs

Subbagian: Go and Rust

501. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

502. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. gcc main.go
- C. go build main.go
- D. go run main.go

503. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

504. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go build main.go
- C. gcc main.go
- D. go run main.go

505. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

506. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. go build main.go
- C. gcc main.go
- D. cargo run

507. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. rustup only
- C. cargo
- D. npm

508. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go build main.go
- C. go run main.go
- D. gcc main.go

509. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. rustup only
- D. npm

510. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. go build main.go
- C. gcc main.go
- D. cargo run

511. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. cargo
- C. gcc
- D. rustup only

512. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go run main.go
- C. gcc main.go
- D. go build main.go

513. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. rustup only
- C. npm
- D. cargo

514. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. go run main.go
- C. gcc main.go
- D. cargo run

515. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. cargo
- C. npm
- D. rustup only

516. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go build main.go
- C. gcc main.go
- D. go run main.go

517. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. gcc
- C. cargo
- D. npm

518. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go build main.go
- C. go run main.go
- D. cargo run

519. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

520. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. cargo run
- C. go build main.go
- D. gcc main.go

521. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. rustup only
- C. npm
- D. cargo

522. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. cargo run
- C. go run main.go
- D. gcc main.go

523. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. cargo
- D. gcc

524. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. cargo run
- D. go build main.go

525. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc

- B. npm
- C. rustup only
- D. cargo

526. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. cargo run
- C. gcc main.go
- D. go run main.go

527. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

528. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go run main.go
- C. go build main.go
- D. gcc main.go

529. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

530. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go build main.go
- C. cargo run
- D. go run main.go

531. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

532. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. cargo run
- D. go build main.go

533. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. cargo
- C. rustup only
- D. npm

534. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go build main.go
- C. go run main.go
- D. cargo run

535. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

536. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. go build main.go
- D. cargo run

537. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. rustup only
- C. npm
- D. gcc

538. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. cargo run

- C. go build main.go
- D. gcc main.go

539. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. npm
- D. rustup only

540. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. go run main.go
- C. cargo run
- D. gcc main.go

541. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. rustup only
- C. cargo
- D. npm

542. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. gcc main.go
- C. go build main.go
- D. go run main.go

543. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. npm
- D. rustup only

544. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. go run main.go
- C. gcc main.go
- D. cargo run

545. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. rustup only
- C. cargo
- D. gcc

546. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. cargo run
- C. go build main.go
- D. go run main.go

547. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

548. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. cargo run
- C. go build main.go
- D. gcc main.go

549. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. gcc
- D. rustup only

550. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. cargo run
- D. go build main.go

551. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. cargo
- C. gcc

D. npm

552. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. cargo run
- D. go build main.go

553. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. cargo
- D. gcc

554. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. gcc main.go
- C. go run main.go
- D. go build main.go

555. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. gcc
- C. cargo
- D. rustup only

556. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go build main.go
- C. cargo run
- D. go run main.go

557. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

558. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. cargo run
- C. go build main.go
- D. gcc main.go

559. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. rustup only
- C. gcc
- D. npm

560. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go build main.go
- C. go run main.go
- D. gcc main.go

561. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. gcc
- C. cargo
- D. npm

562. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go build main.go
- C. cargo run
- D. go run main.go

563. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. cargo
- C. gcc
- D. npm

564. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. go build main.go
- D. cargo run

565. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. rustup only
- C. cargo
- D. npm

566. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. go run main.go
- C. gcc main.go
- D. cargo run

567. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. npm
- C. gcc
- D. cargo

568. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. go run main.go
- C. cargo run
- D. gcc main.go

569. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

570. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. gcc main.go
- C. cargo run
- D. go build main.go

571. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. npm
- D. rustup only

572. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. go build main.go
- D. cargo run

573. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. cargo
- C. rustup only
- D. npm

574. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. go build main.go
- C. cargo run
- D. gcc main.go

575. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. npm
- C. rustup only
- D. gcc

576. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. cargo run
- C. go run main.go
- D. gcc main.go

577. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. gcc
- B. npm
- C. cargo
- D. rustup only

578. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go

- B. gcc main.go
- C. cargo run
- D. go build main.go

579. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. gcc
- C. cargo
- D. npm

580. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go run main.go
- C. gcc main.go
- D. go build main.go

581. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. npm
- D. rustup only

582. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. gcc main.go
- C. cargo run
- D. go build main.go

583. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. gcc
- C. rustup only
- D. npm

584. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. gcc main.go
- C. go build main.go
- D. go run main.go

585. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. gcc
- C. cargo
- D. rustup only

586. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. go run main.go
- C. go build main.go
- D. cargo run

587. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. rustup only
- C. gcc
- D. npm

588. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. gcc main.go
- C. go build main.go
- D. cargo run

589. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. cargo
- C. rustup only
- D. gcc

590. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. cargo run
- B. go build main.go
- C. go run main.go
- D. gcc main.go

591. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. cargo

- C. gcc
- D. rustup only

592. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. gcc main.go
- C. go build main.go
- D. cargo run

593. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. rustup only
- C. gcc
- D. npm

594. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. gcc main.go
- B. cargo run
- C. go build main.go
- D. go run main.go

595. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. rustup only
- B. cargo
- C. gcc
- D. npm

596. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. gcc main.go
- C. cargo run
- D. go build main.go

597. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. cargo
- B. rustup only
- C. gcc
- D. npm

598. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go build main.go
- B. cargo run
- C. gcc main.go
- D. go run main.go

599. Tool build yang paling idiomatik untuk proyek Rust adalah:

- A. npm
- B. rustup only
- C. cargo
- D. gcc

600. Untuk menjalankan file Go bernama main.go tanpa menyimpan binary permanen, perintahnya adalah:

- A. go run main.go
- B. go build main.go
- C. gcc main.go
- D. cargo run

Subbagian: SQL and SQLite

601. Perintah SQL untuk mengambil semua kolom dari tabel users_1 adalah:

- A. SELECT * FROM users_1;
- B. GET * users_1;
- C. READ users_1;
- D. SHOW ALL users_1;

602. Perintah SQL untuk mengambil semua kolom dari tabel users_2 adalah:

- A. SELECT * FROM users_2;
- B. READ users_2;
- C. GET * users_2;
- D. SHOW ALL users_2;

603. Perintah SQL untuk mengambil semua kolom dari tabel users_3 adalah:

- A. SHOW ALL users_3;
- B. SELECT * FROM users_3;
- C. GET * users_3;
- D. READ users_3;

604. Perintah SQL untuk mengambil semua kolom dari tabel users_4 adalah:

- A. READ users_4;

- B. SHOW ALL users_4;
- C. SELECT * FROM users_4;
- D. GET * users_4;

605. Perintah SQL untuk mengambil semua kolom dari tabel users_5 adalah:

- A. SELECT * FROM users_5;
- B. READ users_5;
- C. SHOW ALL users_5;
- D. GET * users_5;

606. Perintah SQL untuk mengambil semua kolom dari tabel users_6 adalah:

- A. GET * users_6;
- B. SHOW ALL users_6;
- C. READ users_6;
- D. SELECT * FROM users_6;

607. Perintah SQL untuk mengambil semua kolom dari tabel users_7 adalah:

- A. GET * users_7;
- B. SHOW ALL users_7;
- C. SELECT * FROM users_7;
- D. READ users_7;

608. Perintah SQL untuk mengambil semua kolom dari tabel users_8 adalah:

- A. SHOW ALL users_8;
- B. READ users_8;
- C. GET * users_8;
- D. SELECT * FROM users_8;

609. Perintah SQL untuk mengambil semua kolom dari tabel users_9 adalah:

- A. SHOW ALL users_9;
- B. READ users_9;
- C. GET * users_9;
- D. SELECT * FROM users_9;

610. Perintah SQL untuk mengambil semua kolom dari tabel users_10 adalah:

- A. SHOW ALL users_10;
- B. READ users_10;
- C. GET * users_10;
- D. SELECT * FROM users_10;

611. Perintah SQL untuk mengambil semua kolom dari tabel users_11 adalah:

- A. READ users_11;
- B. GET * users_11;
- C. SHOW ALL users_11;
- D. SELECT * FROM users_11;

612. Perintah SQL untuk mengambil semua kolom dari tabel users_12 adalah:

- A. GET * users_12;
- B. READ users_12;
- C. SELECT * FROM users_12;
- D. SHOW ALL users_12;

613. Perintah SQL untuk mengambil semua kolom dari tabel users_13 adalah:

- A. READ users_13;
- B. SELECT * FROM users_13;
- C. GET * users_13;
- D. SHOW ALL users_13;

614. Perintah SQL untuk mengambil semua kolom dari tabel users_14 adalah:

- A. READ users_14;
- B. SELECT * FROM users_14;
- C. SHOW ALL users_14;
- D. GET * users_14;

615. Perintah SQL untuk mengambil semua kolom dari tabel users_15 adalah:

- A. SHOW ALL users_15;
- B. GET * users_15;
- C. READ users_15;
- D. SELECT * FROM users_15;

616. Perintah SQL untuk mengambil semua kolom dari tabel users_16 adalah:

- A. SELECT * FROM users_16;
- B. READ users_16;
- C. GET * users_16;
- D. SHOW ALL users_16;

617. Perintah SQL untuk mengambil semua kolom dari tabel users_17 adalah:

- A. SELECT * FROM users_17;
- B. GET * users_17;

- C. READ users_17;
- D. SHOW ALL users_17;

618. Perintah SQL untuk mengambil semua kolom dari tabel users_18 adalah:

- A. READ users_18;
- B. SELECT * FROM users_18;
- C. SHOW ALL users_18;
- D. GET * users_18;

619. Perintah SQL untuk mengambil semua kolom dari tabel users_19 adalah:

- A. SELECT * FROM users_19;
- B. GET * users_19;
- C. SHOW ALL users_19;
- D. READ users_19;

620. Perintah SQL untuk mengambil semua kolom dari tabel users_20 adalah:

- A. SELECT * FROM users_20;
- B. READ users_20;
- C. GET * users_20;
- D. SHOW ALL users_20;

621. Perintah SQL untuk mengambil semua kolom dari tabel users_21 adalah:

- A. SHOW ALL users_21;
- B. READ users_21;
- C. SELECT * FROM users_21;
- D. GET * users_21;

622. Perintah SQL untuk mengambil semua kolom dari tabel users_22 adalah:

- A. SELECT * FROM users_22;
- B. READ users_22;
- C. GET * users_22;
- D. SHOW ALL users_22;

623. Perintah SQL untuk mengambil semua kolom dari tabel users_23 adalah:

- A. READ users_23;
- B. GET * users_23;
- C. SELECT * FROM users_23;
- D. SHOW ALL users_23;

624. Perintah SQL untuk mengambil semua kolom dari tabel users_24 adalah:

- A. GET * users_24;
- B. SHOW ALL users_24;
- C. READ users_24;
- D. SELECT * FROM users_24;

625. Perintah SQL untuk mengambil semua kolom dari tabel users_25 adalah:

- A. SELECT * FROM users_25;
- B. READ users_25;
- C. GET * users_25;
- D. SHOW ALL users_25;

626. Perintah SQL untuk mengambil semua kolom dari tabel users_26 adalah:

- A. READ users_26;
- B. GET * users_26;
- C. SHOW ALL users_26;
- D. SELECT * FROM users_26;

627. Perintah SQL untuk mengambil semua kolom dari tabel users_27 adalah:

- A. SHOW ALL users_27;
- B. SELECT * FROM users_27;
- C. GET * users_27;
- D. READ users_27;

628. Perintah SQL untuk mengambil semua kolom dari tabel users_28 adalah:

- A. SHOW ALL users_28;
- B. SELECT * FROM users_28;
- C. GET * users_28;
- D. READ users_28;

629. Perintah SQL untuk mengambil semua kolom dari tabel users_29 adalah:

- A. READ users_29;
- B. GET * users_29;
- C. SHOW ALL users_29;
- D. SELECT * FROM users_29;

630. Perintah SQL untuk mengambil semua kolom dari tabel users_30 adalah:

- A. SELECT * FROM users_30;
- B. SHOW ALL users_30;
- C. READ users_30;

D. GET * users_30;

631. Perintah SQL untuk mengambil semua kolom dari tabel users_31 adalah:

- A. GET * users_31;
- B. SHOW ALL users_31;
- C. SELECT * FROM users_31;
- D. READ users_31;

632. Perintah SQL untuk mengambil semua kolom dari tabel users_32 adalah:

- A. SHOW ALL users_32;
- B. SELECT * FROM users_32;
- C. GET * users_32;
- D. READ users_32;

633. Perintah SQL untuk mengambil semua kolom dari tabel users_33 adalah:

- A. READ users_33;
- B. GET * users_33;
- C. SELECT * FROM users_33;
- D. SHOW ALL users_33;

634. Perintah SQL untuk mengambil semua kolom dari tabel users_34 adalah:

- A. SELECT * FROM users_34;
- B. READ users_34;
- C. SHOW ALL users_34;
- D. GET * users_34;

635. Perintah SQL untuk mengambil semua kolom dari tabel users_35 adalah:

- A. SELECT * FROM users_35;
- B. GET * users_35;
- C. READ users_35;
- D. SHOW ALL users_35;

636. Perintah SQL untuk mengambil semua kolom dari tabel users_36 adalah:

- A. GET * users_36;
- B. SHOW ALL users_36;
- C. SELECT * FROM users_36;
- D. READ users_36;

637. Perintah SQL untuk mengambil semua kolom dari tabel users_37 adalah:

- A. SHOW ALL users_37;
- B. READ users_37;
- C. SELECT * FROM users_37;
- D. GET * users_37;

638. Perintah SQL untuk mengambil semua kolom dari tabel users_38 adalah:

- A. READ users_38;
- B. GET * users_38;
- C. SHOW ALL users_38;
- D. SELECT * FROM users_38;

639. Perintah SQL untuk mengambil semua kolom dari tabel users_39 adalah:

- A. SELECT * FROM users_39;
- B. SHOW ALL users_39;
- C. GET * users_39;
- D. READ users_39;

640. Perintah SQL untuk mengambil semua kolom dari tabel users_40 adalah:

- A. SHOW ALL users_40;
- B. SELECT * FROM users_40;
- C. READ users_40;
- D. GET * users_40;

641. Perintah SQL untuk mengambil semua kolom dari tabel users_41 adalah:

- A. READ users_41;
- B. SHOW ALL users_41;
- C. SELECT * FROM users_41;
- D. GET * users_41;

642. Perintah SQL untuk mengambil semua kolom dari tabel users_42 adalah:

- A. SHOW ALL users_42;
- B. READ users_42;
- C. SELECT * FROM users_42;
- D. GET * users_42;

643. Perintah SQL untuk mengambil semua kolom dari tabel users_43 adalah:

- A. READ users_43;
- B. GET * users_43;
- C. SELECT * FROM users_43;
- D. SHOW ALL users_43;

644. Perintah SQL untuk mengambil semua kolom dari tabel users_44 adalah:

- A. GET * users_44;
- B. SHOW ALL users_44;
- C. READ users_44;
- D. SELECT * FROM users_44;

645. Perintah SQL untuk mengambil semua kolom dari tabel users_45 adalah:

- A. SHOW ALL users_45;
- B. READ users_45;
- C. SELECT * FROM users_45;
- D. GET * users_45;

646. Perintah SQL untuk mengambil semua kolom dari tabel users_46 adalah:

- A. SELECT * FROM users_46;
- B. SHOW ALL users_46;
- C. READ users_46;
- D. GET * users_46;

647. Perintah SQL untuk mengambil semua kolom dari tabel users_47 adalah:

- A. SELECT * FROM users_47;
- B. READ users_47;
- C. GET * users_47;
- D. SHOW ALL users_47;

648. Perintah SQL untuk mengambil semua kolom dari tabel users_48 adalah:

- A. GET * users_48;
- B. SELECT * FROM users_48;
- C. SHOW ALL users_48;
- D. READ users_48;

649. Perintah SQL untuk mengambil semua kolom dari tabel users_49 adalah:

- A. GET * users_49;
- B. READ users_49;
- C. SELECT * FROM users_49;
- D. SHOW ALL users_49;

650. Perintah SQL untuk mengambil semua kolom dari tabel users_50 adalah:

- A. READ users_50;
- B. SHOW ALL users_50;
- C. GET * users_50;
- D. SELECT * FROM users_50;

651. Perintah SQL untuk mengambil semua kolom dari tabel users_51 adalah:

- A. READ users_51;
- B. GET * users_51;
- C. SHOW ALL users_51;
- D. SELECT * FROM users_51;

652. Perintah SQL untuk mengambil semua kolom dari tabel users_52 adalah:

- A. SHOW ALL users_52;
- B. READ users_52;
- C. SELECT * FROM users_52;
- D. GET * users_52;

653. Perintah SQL untuk mengambil semua kolom dari tabel users_53 adalah:

- A. GET * users_53;
- B. SHOW ALL users_53;
- C. SELECT * FROM users_53;
- D. READ users_53;

654. Perintah SQL untuk mengambil semua kolom dari tabel users_54 adalah:

- A. SELECT * FROM users_54;
- B. SHOW ALL users_54;
- C. READ users_54;
- D. GET * users_54;

655. Perintah SQL untuk mengambil semua kolom dari tabel users_55 adalah:

- A. READ users_55;
- B. SELECT * FROM users_55;
- C. SHOW ALL users_55;
- D. GET * users_55;

656. Perintah SQL untuk mengambil semua kolom dari tabel users_56 adalah:

- A. SHOW ALL users_56;
- B. SELECT * FROM users_56;
- C. READ users_56;
- D. GET * users_56;

657. Perintah SQL untuk mengambil semua kolom dari tabel users_57 adalah:

- A. SELECT * FROM users_57;

- B. SHOW ALL users_57;
- C. READ users_57;
- D. GET * users_57;

658. Perintah SQL untuk mengambil semua kolom dari tabel users_58 adalah:

- A. READ users_58;
- B. SELECT * FROM users_58;
- C. GET * users_58;
- D. SHOW ALL users_58;

659. Perintah SQL untuk mengambil semua kolom dari tabel users_59 adalah:

- A. GET * users_59;
- B. READ users_59;
- C. SHOW ALL users_59;
- D. SELECT * FROM users_59;

660. Perintah SQL untuk mengambil semua kolom dari tabel users_60 adalah:

- A. SELECT * FROM users_60;
- B. READ users_60;
- C. GET * users_60;
- D. SHOW ALL users_60;

661. Perintah SQL untuk mengambil semua kolom dari tabel users_61 adalah:

- A. SHOW ALL users_61;
- B. SELECT * FROM users_61;
- C. READ users_61;
- D. GET * users_61;

662. Perintah SQL untuk mengambil semua kolom dari tabel users_62 adalah:

- A. READ users_62;
- B. SHOW ALL users_62;
- C. SELECT * FROM users_62;
- D. GET * users_62;

663. Perintah SQL untuk mengambil semua kolom dari tabel users_63 adalah:

- A. SHOW ALL users_63;
- B. SELECT * FROM users_63;
- C. GET * users_63;
- D. READ users_63;

664. Perintah SQL untuk mengambil semua kolom dari tabel users_64 adalah:

- A. SHOW ALL users_64;
- B. SELECT * FROM users_64;
- C. READ users_64;
- D. GET * users_64;

665. Perintah SQL untuk mengambil semua kolom dari tabel users_65 adalah:

- A. GET * users_65;
- B. SHOW ALL users_65;
- C. SELECT * FROM users_65;
- D. READ users_65;

666. Perintah SQL untuk mengambil semua kolom dari tabel users_66 adalah:

- A. READ users_66;
- B. GET * users_66;
- C. SHOW ALL users_66;
- D. SELECT * FROM users_66;

667. Perintah SQL untuk mengambil semua kolom dari tabel users_67 adalah:

- A. SHOW ALL users_67;
- B. SELECT * FROM users_67;
- C. READ users_67;
- D. GET * users_67;

668. Perintah SQL untuk mengambil semua kolom dari tabel users_68 adalah:

- A. SHOW ALL users_68;
- B. READ users_68;
- C. SELECT * FROM users_68;
- D. GET * users_68;

669. Perintah SQL untuk mengambil semua kolom dari tabel users_69 adalah:

- A. READ users_69;
- B. GET * users_69;
- C. SHOW ALL users_69;
- D. SELECT * FROM users_69;

670. Perintah SQL untuk mengambil semua kolom dari tabel users_70 adalah:

- A. SHOW ALL users_70;
- B. GET * users_70;

- C. READ users_70;
- D. SELECT * FROM users_70;

671. Perintah SQL untuk mengambil semua kolom dari tabel users_71 adalah:

- A. READ users_71;
- B. SELECT * FROM users_71;
- C. SHOW ALL users_71;
- D. GET * users_71;

672. Perintah SQL untuk mengambil semua kolom dari tabel users_72 adalah:

- A. SHOW ALL users_72;
- B. SELECT * FROM users_72;
- C. READ users_72;
- D. GET * users_72;

673. Perintah SQL untuk mengambil semua kolom dari tabel users_73 adalah:

- A. SELECT * FROM users_73;
- B. GET * users_73;
- C. READ users_73;
- D. SHOW ALL users_73;

674. Perintah SQL untuk mengambil semua kolom dari tabel users_74 adalah:

- A. GET * users_74;
- B. SELECT * FROM users_74;
- C. SHOW ALL users_74;
- D. READ users_74;

675. Perintah SQL untuk mengambil semua kolom dari tabel users_75 adalah:

- A. READ users_75;
- B. GET * users_75;
- C. SHOW ALL users_75;
- D. SELECT * FROM users_75;

676. Perintah SQL untuk mengambil semua kolom dari tabel users_76 adalah:

- A. GET * users_76;
- B. SELECT * FROM users_76;
- C. READ users_76;
- D. SHOW ALL users_76;

677. Perintah SQL untuk mengambil semua kolom dari tabel users_77 adalah:

- A. READ users_77;
- B. GET * users_77;
- C. SELECT * FROM users_77;
- D. SHOW ALL users_77;

678. Perintah SQL untuk mengambil semua kolom dari tabel users_78 adalah:

- A. SELECT * FROM users_78;
- B. GET * users_78;
- C. SHOW ALL users_78;
- D. READ users_78;

679. Perintah SQL untuk mengambil semua kolom dari tabel users_79 adalah:

- A. GET * users_79;
- B. SHOW ALL users_79;
- C. SELECT * FROM users_79;
- D. READ users_79;

680. Perintah SQL untuk mengambil semua kolom dari tabel users_80 adalah:

- A. SELECT * FROM users_80;
- B. READ users_80;
- C. GET * users_80;
- D. SHOW ALL users_80;

681. Perintah SQL untuk mengambil semua kolom dari tabel users_81 adalah:

- A. GET * users_81;
- B. SELECT * FROM users_81;
- C. READ users_81;
- D. SHOW ALL users_81;

682. Perintah SQL untuk mengambil semua kolom dari tabel users_82 adalah:

- A. GET * users_82;
- B. READ users_82;
- C. SHOW ALL users_82;
- D. SELECT * FROM users_82;

683. Perintah SQL untuk mengambil semua kolom dari tabel users_83 adalah:

- A. SELECT * FROM users_83;
- B. SHOW ALL users_83;
- C. GET * users_83;

D. READ users_83;

684. Perintah SQL untuk mengambil semua kolom dari tabel users_84 adalah:

- A. READ users_84;
- B. SELECT * FROM users_84;
- C. GET * users_84;
- D. SHOW ALL users_84;

685. Perintah SQL untuk mengambil semua kolom dari tabel users_85 adalah:

- A. GET * users_85;
- B. SELECT * FROM users_85;
- C. SHOW ALL users_85;
- D. READ users_85;

686. Perintah SQL untuk mengambil semua kolom dari tabel users_86 adalah:

- A. READ users_86;
- B. GET * users_86;
- C. SHOW ALL users_86;
- D. SELECT * FROM users_86;

687. Perintah SQL untuk mengambil semua kolom dari tabel users_87 adalah:

- A. SHOW ALL users_87;
- B. SELECT * FROM users_87;
- C. READ users_87;
- D. GET * users_87;

688. Perintah SQL untuk mengambil semua kolom dari tabel users_88 adalah:

- A. SHOW ALL users_88;
- B. GET * users_88;
- C. SELECT * FROM users_88;
- D. READ users_88;

689. Perintah SQL untuk mengambil semua kolom dari tabel users_89 adalah:

- A. SELECT * FROM users_89;
- B. READ users_89;
- C. SHOW ALL users_89;
- D. GET * users_89;

690. Perintah SQL untuk mengambil semua kolom dari tabel users_90 adalah:

- A. SELECT * FROM users_90;
- B. READ users_90;
- C. GET * users_90;
- D. SHOW ALL users_90;

691. Perintah SQL untuk mengambil semua kolom dari tabel users_91 adalah:

- A. GET * users_91;
- B. SHOW ALL users_91;
- C. READ users_91;
- D. SELECT * FROM users_91;

692. Perintah SQL untuk mengambil semua kolom dari tabel users_92 adalah:

- A. GET * users_92;
- B. SHOW ALL users_92;
- C. READ users_92;
- D. SELECT * FROM users_92;

693. Perintah SQL untuk mengambil semua kolom dari tabel users_93 adalah:

- A. GET * users_93;
- B. SHOW ALL users_93;
- C. READ users_93;
- D. SELECT * FROM users_93;

694. Perintah SQL untuk mengambil semua kolom dari tabel users_94 adalah:

- A. GET * users_94;
- B. SHOW ALL users_94;
- C. SELECT * FROM users_94;
- D. READ users_94;

695. Perintah SQL untuk mengambil semua kolom dari tabel users_95 adalah:

- A. READ users_95;
- B. GET * users_95;
- C. SHOW ALL users_95;
- D. SELECT * FROM users_95;

696. Perintah SQL untuk mengambil semua kolom dari tabel users_96 adalah:

- A. SELECT * FROM users_96;
- B. GET * users_96;
- C. SHOW ALL users_96;
- D. READ users_96;

697. Perintah SQL untuk mengambil semua kolom dari tabel users_97 adalah:

- A. SHOW ALL users_97;
- B. READ users_97;
- C. GET * users_97;
- D. SELECT * FROM users_97;

698. Perintah SQL untuk mengambil semua kolom dari tabel users_98 adalah:

- A. GET * users_98;
- B. SHOW ALL users_98;
- C. READ users_98;
- D. SELECT * FROM users_98;

699. Perintah SQL untuk mengambil semua kolom dari tabel users_99 adalah:

- A. SHOW ALL users_99;
- B. GET * users_99;
- C. SELECT * FROM users_99;
- D. READ users_99;

700. Perintah SQL untuk mengambil semua kolom dari tabel users_100 adalah:

- A. READ users_100;
- B. SHOW ALL users_100;
- C. GET * users_100;
- D. SELECT * FROM users_100;

Subbagian: Functional Programming

701. Konsep yang paling erat dengan Haskell adalah:

- A. manual memory only
- B. immutability
- C. pointer arithmetic
- D. class inheritance only

702. Konsep yang paling erat dengan Scala adalah:

- A. no types
- B. JVM interoperability
- C. assembly syntax
- D. browser-only runtime

703. Konsep yang paling erat dengan Clojure adalah:

- A. manual linker scripts
- B. GPU shaders
- C. C header files
- D. Lisp-style syntax

704. Konsep yang paling erat dengan Elixir adalah:

- A. SQL-only execution
- B. concurrency on the BEAM VM
- C. C preprocessor macros only
- D. exclusive Windows GUI support

705. Konsep yang paling erat dengan Erlang adalah:

- A. template metaprogramming
- B. Excel macro runtime
- C. direct CSS rendering
- D. fault-tolerant actor model

706. Konsep yang paling erat dengan OCaml adalah:

- A. byte-only arithmetic
- B. shell piping as core syntax
- C. strong type inference
- D. HTML as native AST

707. Konsep yang paling erat dengan Scheme adalah:

- A. browser DOM mandatory
- B. symbolic and recursive programming
- C. spreadsheet formulas only
- D. only image editing

708. Konsep yang paling erat dengan Haskell adalah:

- A. manual memory only
- B. immutability
- C. class inheritance only
- D. pointer arithmetic

709. Konsep yang paling erat dengan Scala adalah:

- A. JVM interoperability
- B. no types
- C. assembly syntax
- D. browser-only runtime

710. Konsep yang paling erat dengan Clojure adalah:

- A. C header files
- B. manual linker scripts
- C. Lisp-style syntax
- D. GPU shaders

711. Konsep yang paling erat dengan Elixir adalah:

- A. concurrency on the BEAM VM
- B. SQL-only execution
- C. C preprocessor macros only
- D. exclusive Windows GUI support

712. Konsep yang paling erat dengan Erlang adalah:

- A. template metaprogramming
- B. fault-tolerant actor model
- C. Excel macro runtime
- D. direct CSS rendering

713. Konsep yang paling erat dengan OCaml adalah:

- A. strong type inference
- B. shell piping as core syntax
- C. byte-only arithmetic
- D. HTML as native AST

714. Konsep yang paling erat dengan Scheme adalah:

- A. symbolic and recursive programming
- B. browser DOM mandatory
- C. only image editing
- D. spreadsheet formulas only

715. Konsep yang paling erat dengan Haskell adalah:

- A. immutability
- B. manual memory only
- C. class inheritance only
- D. pointer arithmetic

716. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax
- B. JVM interoperability
- C. browser-only runtime
- D. no types

717. Konsep yang paling erat dengan Clojure adalah:

- A. Lisp-style syntax
- B. C header files
- C. manual linker scripts
- D. GPU shaders

718. Konsep yang paling erat dengan Elixir adalah:

- A. concurrency on the BEAM VM
- B. SQL-only execution
- C. exclusive Windows GUI support
- D. C preprocessor macros only

719. Konsep yang paling erat dengan Erlang adalah:

- A. template metaprogramming
- B. direct CSS rendering
- C. fault-tolerant actor model
- D. Excel macro runtime

720. Konsep yang paling erat dengan OCaml adalah:

- A. byte-only arithmetic
- B. strong type inference
- C. HTML as native AST
- D. shell piping as core syntax

721. Konsep yang paling erat dengan Scheme adalah:

- A. spreadsheet formulas only
- B. symbolic and recursive programming
- C. browser DOM mandatory
- D. only image editing

722. Konsep yang paling erat dengan Haskell adalah:

- A. manual memory only
- B. pointer arithmetic
- C. immutability
- D. class inheritance only

723. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax

- B. no types
- C. JVM interoperability
- D. browser-only runtime

724. Konsep yang paling erat dengan Clojure adalah:

- A. GPU shaders
- B. manual linker scripts
- C. Lisp-style syntax
- D. C header files

725. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. concurrency on the BEAM VM
- C. exclusive Windows GUI support
- D. SQL-only execution

726. Konsep yang paling erat dengan Erlang adalah:

- A. fault-tolerant actor model
- B. direct CSS rendering
- C. Excel macro runtime
- D. template metaprogramming

727. Konsep yang paling erat dengan OCaml adalah:

- A. shell piping as core syntax
- B. HTML as native AST
- C. strong type inference
- D. byte-only arithmetic

728. Konsep yang paling erat dengan Scheme adalah:

- A. browser DOM mandatory
- B. only image editing
- C. symbolic and recursive programming
- D. spreadsheet formulas only

729. Konsep yang paling erat dengan Haskell adalah:

- A. pointer arithmetic
- B. immutability
- C. class inheritance only
- D. manual memory only

730. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax
- B. JVM interoperability
- C. no types
- D. browser-only runtime

731. Konsep yang paling erat dengan Clojure adalah:

- A. C header files
- B. Lisp-style syntax
- C. manual linker scripts
- D. GPU shaders

732. Konsep yang paling erat dengan Elixir adalah:

- A. SQL-only execution
- B. exclusive Windows GUI support
- C. concurrency on the BEAM VM
- D. C preprocessor macros only

733. Konsep yang paling erat dengan Erlang adalah:

- A. direct CSS rendering
- B. template metaprogramming
- C. Excel macro runtime
- D. fault-tolerant actor model

734. Konsep yang paling erat dengan OCaml adalah:

- A. strong type inference
- B. HTML as native AST
- C. byte-only arithmetic
- D. shell piping as core syntax

735. Konsep yang paling erat dengan Scheme adalah:

- A. only image editing
- B. spreadsheet formulas only
- C. browser DOM mandatory
- D. symbolic and recursive programming

736. Konsep yang paling erat dengan Haskell adalah:

- A. class inheritance only
- B. manual memory only

- C. immutability
- D. pointer arithmetic

737. Konsep yang paling erat dengan Scala adalah:

- A. browser-only runtime
- B. assembly syntax
- C. no types
- D. JVM interoperability

738. Konsep yang paling erat dengan Clojure adalah:

- A. C header files
- B. Lisp-style syntax
- C. GPU shaders
- D. manual linker scripts

739. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. SQL-only execution
- C. concurrency on the BEAM VM
- D. exclusive Windows GUI support

740. Konsep yang paling erat dengan Erlang adalah:

- A. template metaprogramming
- B. fault-tolerant actor model
- C. direct CSS rendering
- D. Excel macro runtime

741. Konsep yang paling erat dengan OCaml adalah:

- A. strong type inference
- B. shell piping as core syntax
- C. HTML as native AST
- D. byte-only arithmetic

742. Konsep yang paling erat dengan Scheme adalah:

- A. only image editing
- B. spreadsheet formulas only
- C. symbolic and recursive programming
- D. browser DOM mandatory

743. Konsep yang paling erat dengan Haskell adalah:

- A. pointer arithmetic
- B. manual memory only
- C. immutability
- D. class inheritance only

744. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax
- B. no types
- C. browser-only runtime
- D. JVM interoperability

745. Konsep yang paling erat dengan Clojure adalah:

- A. GPU shaders
- B. C header files
- C. Lisp-style syntax
- D. manual linker scripts

746. Konsep yang paling erat dengan Elixir adalah:

- A. exclusive Windows GUI support
- B. concurrency on the BEAM VM
- C. C preprocessor macros only
- D. SQL-only execution

747. Konsep yang paling erat dengan Erlang adalah:

- A. direct CSS rendering
- B. fault-tolerant actor model
- C. Excel macro runtime
- D. template metaprogramming

748. Konsep yang paling erat dengan OCaml adalah:

- A. shell piping as core syntax
- B. strong type inference
- C. byte-only arithmetic
- D. HTML as native AST

749. Konsep yang paling erat dengan Scheme adalah:

- A. browser DOM mandatory
- B. spreadsheet formulas only
- C. symbolic and recursive programming

D. only image editing

750. Konsep yang paling erat dengan Haskell adalah:

- A. immutability
- B. pointer arithmetic
- C. manual memory only
- D. class inheritance only

751. Konsep yang paling erat dengan Scala adalah:

- A. browser-only runtime
- B. no types
- C. assembly syntax
- D. JVM interoperability

752. Konsep yang paling erat dengan Clojure adalah:

- A. Lisp-style syntax
- B. manual linker scripts
- C. C header files
- D. GPU shaders

753. Konsep yang paling erat dengan Elixir adalah:

- A. concurrency on the BEAM VM
- B. C preprocessor macros only
- C. SQL-only execution
- D. exclusive Windows GUI support

754. Konsep yang paling erat dengan Erlang adalah:

- A. direct CSS rendering
- B. Excel macro runtime
- C. template metaprogramming
- D. fault-tolerant actor model

755. Konsep yang paling erat dengan OCaml adalah:

- A. strong type inference
- B. byte-only arithmetic
- C. HTML as native AST
- D. shell piping as core syntax

756. Konsep yang paling erat dengan Scheme adalah:

- A. spreadsheet formulas only
- B. symbolic and recursive programming
- C. browser DOM mandatory
- D. only image editing

757. Konsep yang paling erat dengan Haskell adalah:

- A. class inheritance only
- B. pointer arithmetic
- C. immutability
- D. manual memory only

758. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax
- B. JVM interoperability
- C. browser-only runtime
- D. no types

759. Konsep yang paling erat dengan Clojure adalah:

- A. C header files
- B. GPU shaders
- C. manual linker scripts
- D. Lisp-style syntax

760. Konsep yang paling erat dengan Elixir adalah:

- A. SQL-only execution
- B. concurrency on the BEAM VM
- C. C preprocessor macros only
- D. exclusive Windows GUI support

761. Konsep yang paling erat dengan Erlang adalah:

- A. Excel macro runtime
- B. template metaprogramming
- C. fault-tolerant actor model
- D. direct CSS rendering

762. Konsep yang paling erat dengan OCaml adalah:

- A. HTML as native AST
- B. shell piping as core syntax
- C. byte-only arithmetic
- D. strong type inference

763. Konsep yang paling erat dengan Scheme adalah:

- A. spreadsheet formulas only
- B. only image editing
- C. symbolic and recursive programming
- D. browser DOM mandatory

764. Konsep yang paling erat dengan Haskell adalah:

- A. pointer arithmetic
- B. immutability
- C. class inheritance only
- D. manual memory only

765. Konsep yang paling erat dengan Scala adalah:

- A. JVM interoperability
- B. assembly syntax
- C. browser-only runtime
- D. no types

766. Konsep yang paling erat dengan Clojure adalah:

- A. C header files
- B. Lisp-style syntax
- C. manual linker scripts
- D. GPU shaders

767. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. SQL-only execution
- C. exclusive Windows GUI support
- D. concurrency on the BEAM VM

768. Konsep yang paling erat dengan Erlang adalah:

- A. template metaprogramming
- B. Excel macro runtime
- C. fault-tolerant actor model
- D. direct CSS rendering

769. Konsep yang paling erat dengan OCaml adalah:

- A. HTML as native AST
- B. byte-only arithmetic
- C. strong type inference
- D. shell piping as core syntax

770. Konsep yang paling erat dengan Scheme adalah:

- A. spreadsheet formulas only
- B. browser DOM mandatory
- C. only image editing
- D. symbolic and recursive programming

771. Konsep yang paling erat dengan Haskell adalah:

- A. pointer arithmetic
- B. class inheritance only
- C. manual memory only
- D. immutability

772. Konsep yang paling erat dengan Scala adalah:

- A. no types
- B. assembly syntax
- C. JVM interoperability
- D. browser-only runtime

773. Konsep yang paling erat dengan Clojure adalah:

- A. Lisp-style syntax
- B. C header files
- C. GPU shaders
- D. manual linker scripts

774. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. concurrency on the BEAM VM
- C. SQL-only execution
- D. exclusive Windows GUI support

775. Konsep yang paling erat dengan Erlang adalah:

- A. Excel macro runtime
- B. fault-tolerant actor model
- C. template metaprogramming
- D. direct CSS rendering

776. Konsep yang paling erat dengan OCaml adalah:

- A. shell piping as core syntax

- B. HTML as native AST
- C. strong type inference
- D. byte-only arithmetic

777. Konsep yang paling erat dengan Scheme adalah:

- A. only image editing
- B. spreadsheet formulas only
- C. browser DOM mandatory
- D. symbolic and recursive programming

778. Konsep yang paling erat dengan Haskell adalah:

- A. immutability
- B. pointer arithmetic
- C. manual memory only
- D. class inheritance only

779. Konsep yang paling erat dengan Scala adalah:

- A. no types
- B. JVM interoperability
- C. browser-only runtime
- D. assembly syntax

780. Konsep yang paling erat dengan Clojure adalah:

- A. GPU shaders
- B. manual linker scripts
- C. C header files
- D. Lisp-style syntax

781. Konsep yang paling erat dengan Elixir adalah:

- A. concurrency on the BEAM VM
- B. exclusive Windows GUI support
- C. SQL-only execution
- D. C preprocessor macros only

782. Konsep yang paling erat dengan Erlang adalah:

- A. fault-tolerant actor model
- B. Excel macro runtime
- C. direct CSS rendering
- D. template metaprogramming

783. Konsep yang paling erat dengan OCaml adalah:

- A. HTML as native AST
- B. byte-only arithmetic
- C. shell piping as core syntax
- D. strong type inference

784. Konsep yang paling erat dengan Scheme adalah:

- A. spreadsheet formulas only
- B. only image editing
- C. symbolic and recursive programming
- D. browser DOM mandatory

785. Konsep yang paling erat dengan Haskell adalah:

- A. class inheritance only
- B. pointer arithmetic
- C. immutability
- D. manual memory only

786. Konsep yang paling erat dengan Scala adalah:

- A. no types
- B. JVM interoperability
- C. browser-only runtime
- D. assembly syntax

787. Konsep yang paling erat dengan Clojure adalah:

- A. GPU shaders
- B. C header files
- C. manual linker scripts
- D. Lisp-style syntax

788. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. concurrency on the BEAM VM
- C. SQL-only execution
- D. exclusive Windows GUI support

789. Konsep yang paling erat dengan Erlang adalah:

- A. fault-tolerant actor model
- B. direct CSS rendering

- C. Excel macro runtime
- D. template metaprogramming

790. Konsep yang paling erat dengan OCaml adalah:

- A. HTML as native AST
- B. strong type inference
- C. byte-only arithmetic
- D. shell piping as core syntax

791. Konsep yang paling erat dengan Scheme adalah:

- A. browser DOM mandatory
- B. symbolic and recursive programming
- C. spreadsheet formulas only
- D. only image editing

792. Konsep yang paling erat dengan Haskell adalah:

- A. class inheritance only
- B. pointer arithmetic
- C. immutability
- D. manual memory only

793. Konsep yang paling erat dengan Scala adalah:

- A. browser-only runtime
- B. JVM interoperability
- C. assembly syntax
- D. no types

794. Konsep yang paling erat dengan Clojure adalah:

- A. manual linker scripts
- B. Lisp-style syntax
- C. GPU shaders
- D. C header files

795. Konsep yang paling erat dengan Elixir adalah:

- A. C preprocessor macros only
- B. exclusive Windows GUI support
- C. SQL-only execution
- D. concurrency on the BEAM VM

796. Konsep yang paling erat dengan Erlang adalah:

- A. Excel macro runtime
- B. fault-tolerant actor model
- C. direct CSS rendering
- D. template metaprogramming

797. Konsep yang paling erat dengan OCaml adalah:

- A. strong type inference
- B. HTML as native AST
- C. byte-only arithmetic
- D. shell piping as core syntax

798. Konsep yang paling erat dengan Scheme adalah:

- A. symbolic and recursive programming
- B. browser DOM mandatory
- C. only image editing
- D. spreadsheet formulas only

799. Konsep yang paling erat dengan Haskell adalah:

- A. pointer arithmetic
- B. class inheritance only
- C. immutability
- D. manual memory only

800. Konsep yang paling erat dengan Scala adalah:

- A. assembly syntax
- B. browser-only runtime
- C. JVM interoperability
- D. no types

Subbagian: Scripting Languages

801. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. expressive scripting and web apps
- C. firmware flashing protocol
- D. spreadsheet-only language

802. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol

- B. spreadsheet-only language
- C. server-side web scripting
- D. GPU-only programming

803. Deskripsi yang paling tepat untuk Lua adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. lightweight embeddable scripting
- D. spreadsheet-only language

804. Deskripsi yang paling tepat untuk Tcl adalah:

- A. command-oriented scripting
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

805. Deskripsi yang paling tepat untuk Bash adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. shell automation on Unix-like systems
- D. spreadsheet-only language

806. Deskripsi yang paling tepat untuk Nim adalah:

- A. spreadsheet-only language
- B. compiled language with Python-like feel
- C. GPU-only programming
- D. firmware flashing protocol

807. Deskripsi yang paling tepat untuk Crystal adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. Ruby-like syntax with native compilation
- D. GPU-only programming

808. Deskripsi yang paling tepat untuk Ruby adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. GPU-only programming
- D. expressive scripting and web apps

809. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol
- B. server-side web scripting
- C. GPU-only programming
- D. spreadsheet-only language

810. Deskripsi yang paling tepat untuk Lua adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. lightweight embeddable scripting
- D. GPU-only programming

811. Deskripsi yang paling tepat untuk Tcl adalah:

- A. command-oriented scripting
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. GPU-only programming

812. Deskripsi yang paling tepat untuk Bash adalah:

- A. firmware flashing protocol
- B. shell automation on Unix-like systems
- C. GPU-only programming
- D. spreadsheet-only language

813. Deskripsi yang paling tepat untuk Nim adalah:

- A. spreadsheet-only language
- B. compiled language with Python-like feel
- C. GPU-only programming
- D. firmware flashing protocol

814. Deskripsi yang paling tepat untuk Crystal adalah:

- A. Ruby-like syntax with native compilation
- B. GPU-only programming
- C. spreadsheet-only language
- D. firmware flashing protocol

815. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. expressive scripting and web apps

- C. spreadsheet-only language
- D. firmware flashing protocol

816. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. GPU-only programming
- D. server-side web scripting

817. Deskripsi yang paling tepat untuk Lua adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. firmware flashing protocol
- D. lightweight embeddable scripting

818. Deskripsi yang paling tepat untuk Tcl adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. spreadsheet-only language
- D. command-oriented scripting

819. Deskripsi yang paling tepat untuk Bash adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. firmware flashing protocol
- D. shell automation on Unix-like systems

820. Deskripsi yang paling tepat untuk Nim adalah:

- A. compiled language with Python-like feel
- B. spreadsheet-only language
- C. GPU-only programming
- D. firmware flashing protocol

821. Deskripsi yang paling tepat untuk Crystal adalah:

- A. firmware flashing protocol
- B. Ruby-like syntax with native compilation
- C. GPU-only programming
- D. spreadsheet-only language

822. Deskripsi yang paling tepat untuk Ruby adalah:

- A. spreadsheet-only language
- B. expressive scripting and web apps
- C. GPU-only programming
- D. firmware flashing protocol

823. Deskripsi yang paling tepat untuk PHP adalah:

- A. server-side web scripting
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. GPU-only programming

824. Deskripsi yang paling tepat untuk Lua adalah:

- A. GPU-only programming
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. lightweight embeddable scripting

825. Deskripsi yang paling tepat untuk Tcl adalah:

- A. firmware flashing protocol
- B. command-oriented scripting
- C. spreadsheet-only language
- D. GPU-only programming

826. Deskripsi yang paling tepat untuk Bash adalah:

- A. shell automation on Unix-like systems
- B. spreadsheet-only language
- C. GPU-only programming
- D. firmware flashing protocol

827. Deskripsi yang paling tepat untuk Nim adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. compiled language with Python-like feel
- D. firmware flashing protocol

828. Deskripsi yang paling tepat untuk Crystal adalah:

- A. GPU-only programming
- B. Ruby-like syntax with native compilation
- C. spreadsheet-only language

D. firmware flashing protocol

829. Deskripsi yang paling tepat untuk Ruby adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. expressive scripting and web apps
- D. firmware flashing protocol

830. Deskripsi yang paling tepat untuk PHP adalah:

- A. GPU-only programming
- B. server-side web scripting
- C. firmware flashing protocol
- D. spreadsheet-only language

831. Deskripsi yang paling tepat untuk Lua adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. lightweight embeddable scripting

832. Deskripsi yang paling tepat untuk Tcl adalah:

- A. command-oriented scripting
- B. GPU-only programming
- C. spreadsheet-only language
- D. firmware flashing protocol

833. Deskripsi yang paling tepat untuk Bash adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. spreadsheet-only language
- D. shell automation on Unix-like systems

834. Deskripsi yang paling tepat untuk Nim adalah:

- A. compiled language with Python-like feel
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. GPU-only programming

835. Deskripsi yang paling tepat untuk Crystal adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. firmware flashing protocol
- D. Ruby-like syntax with native compilation

836. Deskripsi yang paling tepat untuk Ruby adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. expressive scripting and web apps
- D. GPU-only programming

837. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol
- B. server-side web scripting
- C. spreadsheet-only language
- D. GPU-only programming

838. Deskripsi yang paling tepat untuk Lua adalah:

- A. lightweight embeddable scripting
- B. spreadsheet-only language
- C. GPU-only programming
- D. firmware flashing protocol

839. Deskripsi yang paling tepat untuk Tcl adalah:

- A. spreadsheet-only language
- B. command-oriented scripting
- C. GPU-only programming
- D. firmware flashing protocol

840. Deskripsi yang paling tepat untuk Bash adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. shell automation on Unix-like systems

841. Deskripsi yang paling tepat untuk Nim adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. spreadsheet-only language
- D. compiled language with Python-like feel

842. Deskripsi yang paling tepat untuk Crystal adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. GPU-only programming
- D. Ruby-like syntax with native compilation

843. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. expressive scripting and web apps
- C. spreadsheet-only language
- D. firmware flashing protocol

844. Deskripsi yang paling tepat untuk PHP adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. server-side web scripting
- D. firmware flashing protocol

845. Deskripsi yang paling tepat untuk Lua adalah:

- A. lightweight embeddable scripting
- B. spreadsheet-only language
- C. GPU-only programming
- D. firmware flashing protocol

846. Deskripsi yang paling tepat untuk Tcl adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. command-oriented scripting
- D. firmware flashing protocol

847. Deskripsi yang paling tepat untuk Bash adalah:

- A. shell automation on Unix-like systems
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

848. Deskripsi yang paling tepat untuk Nim adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. compiled language with Python-like feel
- D. GPU-only programming

849. Deskripsi yang paling tepat untuk Crystal adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. Ruby-like syntax with native compilation
- D. GPU-only programming

850. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. expressive scripting and web apps

851. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol
- B. server-side web scripting
- C. GPU-only programming
- D. spreadsheet-only language

852. Deskripsi yang paling tepat untuk Lua adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. firmware flashing protocol
- D. lightweight embeddable scripting

853. Deskripsi yang paling tepat untuk Tcl adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. command-oriented scripting
- D. GPU-only programming

854. Deskripsi yang paling tepat untuk Bash adalah:

- A. shell automation on Unix-like systems
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. GPU-only programming

855. Deskripsi yang paling tepat untuk Nim adalah:

- A. spreadsheet-only language

- B. GPU-only programming
- C. firmware flashing protocol
- D. compiled language with Python-like feel

856. Deskripsi yang paling tepat untuk Crystal adalah:

- A. Ruby-like syntax with native compilation
- B. firmware flashing protocol
- C. GPU-only programming
- D. spreadsheet-only language

857. Deskripsi yang paling tepat untuk Ruby adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. GPU-only programming
- D. expressive scripting and web apps

858. Deskripsi yang paling tepat untuk PHP adalah:

- A. firmware flashing protocol
- B. server-side web scripting
- C. GPU-only programming
- D. spreadsheet-only language

859. Deskripsi yang paling tepat untuk Lua adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. spreadsheet-only language
- D. lightweight embeddable scripting

860. Deskripsi yang paling tepat untuk Tcl adalah:

- A. command-oriented scripting
- B. GPU-only programming
- C. firmware flashing protocol
- D. spreadsheet-only language

861. Deskripsi yang paling tepat untuk Bash adalah:

- A. firmware flashing protocol
- B. shell automation on Unix-like systems
- C. spreadsheet-only language
- D. GPU-only programming

862. Deskripsi yang paling tepat untuk Nim adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. compiled language with Python-like feel

863. Deskripsi yang paling tepat untuk Crystal adalah:

- A. Ruby-like syntax with native compilation
- B. GPU-only programming
- C. spreadsheet-only language
- D. firmware flashing protocol

864. Deskripsi yang paling tepat untuk Ruby adalah:

- A. expressive scripting and web apps
- B. GPU-only programming
- C. spreadsheet-only language
- D. firmware flashing protocol

865. Deskripsi yang paling tepat untuk PHP adalah:

- A. server-side web scripting
- B. GPU-only programming
- C. firmware flashing protocol
- D. spreadsheet-only language

866. Deskripsi yang paling tepat untuk Lua adalah:

- A. lightweight embeddable scripting
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

867. Deskripsi yang paling tepat untuk Tcl adalah:

- A. spreadsheet-only language
- B. command-oriented scripting
- C. GPU-only programming
- D. firmware flashing protocol

868. Deskripsi yang paling tepat untuk Bash adalah:

- A. shell automation on Unix-like systems
- B. firmware flashing protocol

- C. spreadsheet-only language
- D. GPU-only programming

869. Deskripsi yang paling tepat untuk Nim adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. compiled language with Python-like feel
- D. GPU-only programming

870. Deskripsi yang paling tepat untuk Crystal adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. GPU-only programming
- D. Ruby-like syntax with native compilation

871. Deskripsi yang paling tepat untuk Ruby adalah:

- A. expressive scripting and web apps
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

872. Deskripsi yang paling tepat untuk PHP adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. server-side web scripting
- D. firmware flashing protocol

873. Deskripsi yang paling tepat untuk Lua adalah:

- A. GPU-only programming
- B. lightweight embeddable scripting
- C. firmware flashing protocol
- D. spreadsheet-only language

874. Deskripsi yang paling tepat untuk Tcl adalah:

- A. GPU-only programming
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. command-oriented scripting

875. Deskripsi yang paling tepat untuk Bash adalah:

- A. firmware flashing protocol
- B. shell automation on Unix-like systems
- C. GPU-only programming
- D. spreadsheet-only language

876. Deskripsi yang paling tepat untuk Nim adalah:

- A. compiled language with Python-like feel
- B. firmware flashing protocol
- C. GPU-only programming
- D. spreadsheet-only language

877. Deskripsi yang paling tepat untuk Crystal adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. firmware flashing protocol
- D. Ruby-like syntax with native compilation

878. Deskripsi yang paling tepat untuk Ruby adalah:

- A. firmware flashing protocol
- B. expressive scripting and web apps
- C. GPU-only programming
- D. spreadsheet-only language

879. Deskripsi yang paling tepat untuk PHP adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. server-side web scripting
- D. GPU-only programming

880. Deskripsi yang paling tepat untuk Lua adalah:

- A. lightweight embeddable scripting
- B. GPU-only programming
- C. firmware flashing protocol
- D. spreadsheet-only language

881. Deskripsi yang paling tepat untuk Tcl adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. command-oriented scripting

D. firmware flashing protocol

882. Deskripsi yang paling tepat untuk Bash adalah:

- A. spreadsheet-only language
- B. GPU-only programming
- C. shell automation on Unix-like systems
- D. firmware flashing protocol

883. Deskripsi yang paling tepat untuk Nim adalah:

- A. compiled language with Python-like feel
- B. firmware flashing protocol
- C. GPU-only programming
- D. spreadsheet-only language

884. Deskripsi yang paling tepat untuk Crystal adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. spreadsheet-only language
- D. Ruby-like syntax with native compilation

885. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. expressive scripting and web apps
- C. firmware flashing protocol
- D. spreadsheet-only language

886. Deskripsi yang paling tepat untuk PHP adalah:

- A. server-side web scripting
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

887. Deskripsi yang paling tepat untuk Lua adalah:

- A. GPU-only programming
- B. firmware flashing protocol
- C. lightweight embeddable scripting
- D. spreadsheet-only language

888. Deskripsi yang paling tepat untuk Tcl adalah:

- A. firmware flashing protocol
- B. GPU-only programming
- C. command-oriented scripting
- D. spreadsheet-only language

889. Deskripsi yang paling tepat untuk Bash adalah:

- A. GPU-only programming
- B. shell automation on Unix-like systems
- C. spreadsheet-only language
- D. firmware flashing protocol

890. Deskripsi yang paling tepat untuk Nim adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. compiled language with Python-like feel
- D. firmware flashing protocol

891. Deskripsi yang paling tepat untuk Crystal adalah:

- A. Ruby-like syntax with native compilation
- B. GPU-only programming
- C. firmware flashing protocol
- D. spreadsheet-only language

892. Deskripsi yang paling tepat untuk Ruby adalah:

- A. firmware flashing protocol
- B. spreadsheet-only language
- C. GPU-only programming
- D. expressive scripting and web apps

893. Deskripsi yang paling tepat untuk PHP adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. GPU-only programming
- D. server-side web scripting

894. Deskripsi yang paling tepat untuk Lua adalah:

- A. lightweight embeddable scripting
- B. firmware flashing protocol
- C. GPU-only programming
- D. spreadsheet-only language

895. Deskripsi yang paling tepat untuk Tcl adalah:

- A. command-oriented scripting
- B. firmware flashing protocol
- C. spreadsheet-only language
- D. GPU-only programming

896. Deskripsi yang paling tepat untuk Bash adalah:

- A. spreadsheet-only language
- B. firmware flashing protocol
- C. shell automation on Unix-like systems
- D. GPU-only programming

897. Deskripsi yang paling tepat untuk Nim adalah:

- A. GPU-only programming
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. compiled language with Python-like feel

898. Deskripsi yang paling tepat untuk Crystal adalah:

- A. GPU-only programming
- B. Ruby-like syntax with native compilation
- C. spreadsheet-only language
- D. firmware flashing protocol

899. Deskripsi yang paling tepat untuk Ruby adalah:

- A. GPU-only programming
- B. expressive scripting and web apps
- C. spreadsheet-only language
- D. firmware flashing protocol

900. Deskripsi yang paling tepat untuk PHP adalah:

- A. server-side web scripting
- B. spreadsheet-only language
- C. firmware flashing protocol
- D. GPU-only programming

Subbagian: Tooling

901. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

902. Package manager utama pada ekosistem Node.js adalah:

- A. cargo
- B. kotlinc
- C. pipx-only
- D. npm

903. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. npx
- C. vite --always
- D. node -g

904. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. Deno
- C. NASM
- D. JVM

905. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. GHC
- B. Rscript
- C. V8 source only
- D. Bun

906. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. sqlite3
- C. nasm
- D. Vite

907. Package manager utama pada ekosistem Node.js adalah:

- A. pipx-only
- B. cargo
- C. kotlinc
- D. npm

908. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. node -g
- C. vite --always
- D. npx

909. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. Deno
- C. JVM
- D. NASM

910. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. GHC
- C. Rscript
- D. V8 source only

911. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

912. Package manager utama pada ekosistem Node.js adalah:

- A. cargo
- B. kotlinc
- C. npm
- D. pipx-only

913. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. node -g
- B. python -m npx
- C. npx
- D. vite --always

914. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. NASM
- C. JVM
- D. Deno

915. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Rscript
- B. Bun
- C. V8 source only
- D. GHC

916. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. nasm
- B. Vite
- C. erl
- D. sqlite3

917. Package manager utama pada ekosistem Node.js adalah:

- A. pipx-only
- B. kotlinc
- C. npm
- D. cargo

918. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. npx
- B. node -g
- C. python -m npx
- D. vite --always

919. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. NASM
- B. SQLite
- C. JVM
- D. Deno

920. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. V8 source only
- B. Bun
- C. GHC
- D. Rscript

921. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. sqlite3

- B. erl
- C. nasm
- D. Vite

922. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. pipx-only
- C. cargo
- D. npm

923. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. npx
- B. python -m npx
- C. vite --always
- D. node -g

924. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. JVM
- C. NASM
- D. SQLite

925. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Rscript
- B. Bun
- C. GHC
- D. V8 source only

926. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. nasm
- B. Vite
- C. sqlite3
- D. erl

927. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. cargo
- C. pipx-only
- D. npm

928. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. vite --always
- C. node -g
- D. npx

929. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. Deno
- C. NASM
- D. JVM

930. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. GHC
- B. Rscript
- C. Bun
- D. V8 source only

931. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

932. Package manager utama pada ekosistem Node.js adalah:

- A. npm
- B. kotlinc
- C. cargo
- D. pipx-only

933. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. node -g
- B. vite --always
- C. npx
- D. python -m npx

934. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. NASM

- C. Deno
- D. JVM

935. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. GHC
- B. Bun
- C. V8 source only
- D. Rscript

936. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

937. Package manager utama pada ekosistem Node.js adalah:

- A. pipx-only
- B. kotlinc
- C. cargo
- D. npm

938. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. node -g
- B. npx
- C. python -m npx
- D. vite --always

939. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. NASM
- C. JVM
- D. SQLite

940. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. V8 source only
- C. GHC
- D. Rscript

941. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. sqlite3
- D. nasm

942. Package manager utama pada ekosistem Node.js adalah:

- A. npm
- B. pipx-only
- C. kotlinc
- D. cargo

943. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. vite --always
- B. python -m npx
- C. node -g
- D. npx

944. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. SQLite
- C. NASM
- D. JVM

945. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Rscript
- B. GHC
- C. Bun
- D. V8 source only

946. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. sqlite3
- B. nasm
- C. Vite
- D. erl

947. Package manager utama pada ekosistem Node.js adalah:

- A. cargo
- B. npm
- C. kotlinc

D. ppx-only

948. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. vite --always
- B. node -g
- C. npx
- D. python -m npx

949. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. NASM
- B. SQLite
- C. Deno
- D. JVM

950. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. GHC
- C. V8 source only
- D. Rscript

951. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. nasm
- B. sqlite3
- C. erl
- D. Vite

952. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. npm
- C. ppx-only
- D. cargo

953. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. npx
- C. vite --always
- D. node -g

954. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. NASM
- C. SQLite
- D. JVM

955. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. V8 source only
- C. GHC
- D. Rscript

956. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. sqlite3
- B. erl
- C. Vite
- D. nasm

957. Package manager utama pada ekosistem Node.js adalah:

- A. npm
- B. cargo
- C. kotlinc
- D. ppx-only

958. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. npx
- B. node -g
- C. vite --always
- D. python -m npx

959. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. NASM
- B. JVM
- C. SQLite
- D. Deno

960. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. GHC
- B. V8 source only
- C. Bun
- D. Rscript

961. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. sqlite3
- B. erl
- C. nasm
- D. Vite

962. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. cargo
- C. pipx-only
- D. npm

963. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. vite --always
- B. python -m npx
- C. node -g
- D. npx

964. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. SQLite
- C. JVM
- D. NASM

965. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. V8 source only
- C. Rscript
- D. GHC

966. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. Vite
- B. sqlite3
- C. nasm
- D. erl

967. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. npm
- C. pipx-only
- D. cargo

968. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. node -g
- B. vite --always
- C. npx
- D. python -m npx

969. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. SQLite
- C. NASM
- D. JVM

970. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. GHC
- B. Bun
- C. Rscript
- D. V8 source only

971. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. sqlite3
- C. Vite
- D. nasm

972. Package manager utama pada ekosistem Node.js adalah:

- A. pipx-only
- B. cargo
- C. kotlinc
- D. npm

973. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. node -g
- B. vite --always
- C. npx
- D. python -m npx

974. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno

- B. JVM
- C. SQLite
- D. NASM

975. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. V8 source only
- C. GHC
- D. Rscript

976. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. Vite
- B. sqlite3
- C. erl
- D. nasm

977. Package manager utama pada ekosistem Node.js adalah:

- A. npm
- B. pipx-only
- C. cargo
- D. kotlinc

978. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. vite --always
- B. python -m npx
- C. npx
- D. node -g

979. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. JVM
- B. NASM
- C. SQLite
- D. Deno

980. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. Rscript
- C. GHC
- D. V8 source only

981. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. Vite
- B. nasm
- C. sqlite3
- D. erl

982. Package manager utama pada ekosistem Node.js adalah:

- A. pipx-only
- B. cargo
- C. kotlinc
- D. npm

983. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. node -g
- C. npx
- D. vite --always

984. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. SQLite
- B. NASM
- C. JVM
- D. Deno

985. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. V8 source only
- B. Bun
- C. GHC
- D. Rscript

986. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. sqlite3
- B. nasm
- C. erl
- D. Vite

987. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. cargo

- C. pipx-only
- D. npm

988. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. vite --always
- B. python -m npx
- C. node -g
- D. npx

989. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. SQLite
- C. NASM
- D. JVM

990. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. Bun
- B. Rscript
- C. V8 source only
- D. GHC

991. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

992. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. cargo
- C. pipx-only
- D. npm

993. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. node -g
- C. npx
- D. vite --always

994. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. JVM
- B. Deno
- C. SQLite
- D. NASM

995. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. V8 source only
- B. Bun
- C. Rscript
- D. GHC

996. Tool yang dipakai untuk membuat project frontend cepat dengan template modern adalah:

- A. erl
- B. Vite
- C. nasm
- D. sqlite3

997. Package manager utama pada ekosistem Node.js adalah:

- A. kotlinc
- B. npm
- C. pipx-only
- D. cargo

998. Perintah yang sering dipakai untuk menjalankan package CLI tanpa install global permanen adalah:

- A. python -m npx
- B. npx
- C. node -g
- D. vite --always

999. Runtime modern yang dapat menjalankan TypeScript secara langsung dengan model permission adalah:

- A. Deno
- B. JVM
- C. NASM
- D. SQLite

1000. Runtime JavaScript/TypeScript yang menekankan kecepatan dan juga punya package manager terintegrasi adalah:

- A. V8 source only
- B. Rscript
- C. GHC

D. Bun

Bagian 6 - Kunci jawaban 1000 soal

Gunakan kunci ini setelah mengerjakan seluruh set. Format: nomor soal, huruf jawaban benar, lalu inti jawaban dalam kurung.

1. D (9); 2. D (16); 3. A (25); 4. B (36); 5. B (49); 6. D (64)
7. B (4); 8. B (9); 9. A (16); 10. D (25); 11. C (36); 12. D (49)
13. B (64); 14. C (4); 15. D (9); 16. C (16); 17. D (25); 18. B (36)
19. A (49); 20. C (64); 21. B (4); 22. C (9); 23. C (16); 24. A (25)
25. D (36); 26. D (49); 27. A (64); 28. A (4); 29. C (9); 30. C (16)
31. D (25); 32. D (36); 33. A (49); 34. D (64); 35. B (4); 36. A (9)
37. D (16); 38. A (25); 39. B (36); 40. D (49); 41. C (64); 42. B (4)
43. A (9); 44. A (16); 45. A (25); 46. C (36); 47. B (49); 48. C (64)
49. A (4); 50. C (9); 51. A (16); 52. B (25); 53. D (36); 54. D (49)
55. C (64); 56. A (4); 57. B (9); 58. D (16); 59. C (25); 60. D (36)
61. D (49); 62. C (64); 63. B (4); 64. C (9); 65. D (16); 66. C (25)
67. D (36); 68. D (49); 69. B (64); 70. A (4); 71. B (9); 72. A (16)
73. C (25); 74. A (36); 75. A (49); 76. D (64); 77. C (4); 78. B (9)
79. D (16); 80. A (25); 81. A (36); 82. B (49); 83. D (64); 84. B (4)
85. A (9); 86. C (16); 87. B (25); 88. B (36); 89. A (49); 90. C (64)
91. A (4); 92. B (9); 93. B (16); 94. A (25); 95. B (36); 96. D (49)
97. C (64); 98. B (4); 99. B (9); 100. C (16); 101. B (gcc main.c -o main1.exe); 102. D (gcc main.c -o main2.exe)
103. D (gcc main.c -o main3.exe); 104. D (gcc main.c -o main4.exe); 105. C (gcc main.c -o main5.exe); 106. A (gcc main.c -o main6.exe); 107. C (gcc main.c -o main7.exe); 108. B (gcc main.c -o main8.exe)
109. A (gcc main.c -o main9.exe); 110. C (gcc main.c -o main10.exe); 111. B (gcc main.c -o main11.exe); 112. A (gcc main.c -o main12.exe); 113. B (gcc main.c -o main13.exe); 114. C (gcc main.c -o main14.exe)
115. D (gcc main.c -o main15.exe); 116. C (gcc main.c -o main16.exe); 117. B (gcc main.c -o main17.exe); 118. C (gcc main.c -o main18.exe); 119. D (gcc main.c -o main19.exe); 120. D (gcc main.c -o main20.exe)
121. D (gcc main.c -o main21.exe); 122. C (gcc main.c -o main22.exe); 123. A (gcc main.c -o main23.exe); 124. C (gcc main.c -o main24.exe); 125. C (gcc main.c -o main25.exe); 126. D (gcc main.c -o main26.exe)
127. D (gcc main.c -o main27.exe); 128. D (gcc main.c -o main28.exe); 129. B (gcc main.c -o main29.exe); 130. D (gcc main.c -o main30.exe); 131. A (gcc main.c -o main31.exe); 132. D (gcc main.c -o main32.exe)
133. C (gcc main.c -o main33.exe); 134. D (gcc main.c -o main34.exe); 135. A (gcc main.c -o main35.exe); 136. B (gcc main.c -o main36.exe); 137. D (gcc main.c -o main37.exe); 138. C (gcc main.c -o main38.exe)
139. C (gcc main.c -o main39.exe); 140. D (gcc main.c -o main40.exe); 141. C (gcc main.c -o main41.exe); 142. C (gcc main.c -o main42.exe); 143. D (gcc main.c -o main43.exe); 144. A (gcc main.c -o main44.exe)
145. A (gcc main.c -o main45.exe); 146. D (gcc main.c -o main46.exe); 147. A (gcc main.c -o main47.exe); 148. C (gcc main.c -o main48.exe); 149. D (gcc main.c -o main49.exe); 150. B (gcc main.c -o main50.exe)
151. B (gcc main.c -o main51.exe); 152. B (gcc main.c -o main52.exe); 153. D (gcc main.c -o main53.exe); 154. B (gcc main.c -o main54.exe); 155. D (gcc main.c -o main55.exe); 156. B (gcc main.c -o main56.exe)
157. B (gcc main.c -o main57.exe); 158. B (gcc main.c -o main58.exe); 159. C (gcc main.c -o main59.exe); 160. C (gcc main.c -o main60.exe); 161. C (gcc main.c -o main61.exe); 162. C (gcc main.c -o main62.exe)
163. B (gcc main.c -o main63.exe); 164. B (gcc main.c -o main64.exe); 165. D (gcc main.c -o main65.exe); 166. A (gcc main.c -o main66.exe); 167. D (gcc main.c -o main67.exe); 168. D (gcc main.c -o main68.exe)
169. B (gcc main.c -o main69.exe); 170. D (gcc main.c -o main70.exe); 171. A (gcc main.c -o main71.exe); 172. A (gcc main.c -o main72.exe); 173. A (gcc main.c -o main73.exe); 174. D (gcc main.c -o main74.exe)
175. C (gcc main.c -o main75.exe); 176. D (gcc main.c -o main76.exe); 177. C (gcc main.c -o main77.exe); 178. B (gcc main.c -o main78.exe); 179. A (gcc main.c -o main79.exe); 180. B (gcc main.c -o main80.exe)
181. C (gcc main.c -o main81.exe); 182. A (gcc main.c -o main82.exe); 183. B (gcc main.c -o main83.exe); 184. B (gcc main.c -o main84.exe); 185. A (gcc main.c -o main85.exe); 186. C (gcc main.c -o main86.exe)
187. D (gcc main.c -o main87.exe); 188. B (gcc main.c -o main88.exe); 189. C (gcc main.c -o main89.exe); 190. D (gcc main.c -o main90.exe); 191. C (gcc main.c -o main91.exe); 192. A (gcc main.c -o main92.exe)
193. A (gcc main.c -o main93.exe); 194. D (gcc main.c -o main94.exe); 195. A (gcc main.c -o main95.exe); 196. C (gcc main.c -o main96.exe); 197. B (gcc main.c -o main97.exe); 198. B (gcc main.c -o main98.exe)

199. C (gcc main.c -o main99.exe); 200. D (gcc main.c -o main100.exe); 201. B (java Main); 202. A (java Main); 203. D (java Main); 204. C (java Main)

205. D (java Main); 206. A (java Main); 207. A (java Main); 208. B (java Main); 209. A (java Main); 210. D (java Main)

211. B (java Main); 212. C (java Main); 213. A (java Main); 214. A (java Main); 215. C (java Main); 216. D (java Main)

217. B (java Main); 218. D (java Main); 219. A (java Main); 220. B (java Main); 221. B (java Main); 222. A (java Main)

223. C (java Main); 224. C (java Main); 225. A (java Main); 226. A (java Main); 227. D (java Main); 228. A (java Main)

229. A (java Main); 230. A (java Main); 231. C (java Main); 232. D (java Main); 233. C (java Main); 234. A (java Main)

235. C (java Main); 236. C (java Main); 237. A (java Main); 238. B (java Main); 239. D (java Main); 240. B (java Main)

241. D (java Main); 242. C (java Main); 243. C (java Main); 244. A (java Main); 245. A (java Main); 246. C (java Main)

247. A (java Main); 248. D (java Main); 249. D (java Main); 250. C (java Main); 251. C (java Main); 252. A (java Main)

253. C (java Main); 254. A (java Main); 255. B (java Main); 256. A (java Main); 257. C (java Main); 258. D (java Main)

259. C (java Main); 260. C (java Main); 261. D (java Main); 262. D (java Main); 263. D (java Main); 264. D (java Main)

265. A (java Main); 266. D (java Main); 267. B (java Main); 268. B (java Main); 269. B (java Main); 270. B (java Main)

271. C (java Main); 272. C (java Main); 273. C (java Main); 274. B (java Main); 275. A (java Main); 276. C (java Main)

277. C (java Main); 278. C (java Main); 279. D (java Main); 280. D (java Main); 281. C (java Main); 282. D (java Main)

283. A (java Main); 284. C (java Main); 285. C (java Main); 286. A (java Main); 287. B (java Main); 288. D (java Main)

289. B (java Main); 290. A (java Main); 291. B (java Main); 292. B (java Main); 293. D (java Main); 294. A (java Main)

295. B (java Main); 296. D (java Main); 297. C (java Main); 298. A (java Main); 299. A (java Main); 300. B (java Main)

301. A (tsc); 302. D (node app.js); 303. D (tsc); 304. C (node app.js); 305. B (tsc); 306. B (node app.js)

307. A (tsc); 308. B (node app.js); 309. D (tsc); 310. B (node app.js); 311. D (tsc); 312. A (node app.js)

313. D (tsc); 314. B (node app.js); 315. D (tsc); 316. C (node app.js); 317. B (tsc); 318. B (node app.js)

319. D (tsc); 320. D (node app.js); 321. D (tsc); 322. A (node app.js); 323. B (tsc); 324. A (node app.js)

325. A (tsc); 326. C (node app.js); 327. C (tsc); 328. A (node app.js); 329. B (tsc); 330. D (node app.js)

331. A (tsc); 332. D (node app.js); 333. C (tsc); 334. B (node app.js); 335. C (tsc); 336. D (node app.js)

337. B (tsc); 338. C (node app.js); 339. A (tsc); 340. B (node app.js); 341. C (tsc); 342. A (node app.js)

343. B (tsc); 344. A (node app.js); 345. B (tsc); 346. A (node app.js); 347. B (tsc); 348. C (node app.js)

349. B (tsc); 350. A (node app.js); 351. D (tsc); 352. A (node app.js); 353. C (tsc); 354. D (node app.js)

355. D (tsc); 356. B (node app.js); 357. A (tsc); 358. C (node app.js); 359. B (tsc); 360. B (node app.js)

361. A (tsc); 362. A (node app.js); 363. D (tsc); 364. A (node app.js); 365. D (tsc); 366. A (node app.js)

367. A (tsc); 368. B (node app.js); 369. B (tsc); 370. D (node app.js); 371. D (tsc); 372. A (node app.js)

373. C (tsc); 374. D (node app.js); 375. C (tsc); 376. B (node app.js); 377. A (tsc); 378. D (node app.js)

379. C (tsc); 380. A (node app.js); 381. D (tsc); 382. A (node app.js); 383. A (tsc); 384. A (node app.js)

385. D (tsc); 386. C (node app.js); 387. D (tsc); 388. C (node app.js); 389. A (tsc); 390. D (node app.js)

391. B (tsc); 392. B (node app.js); 393. B (tsc); 394. D (node app.js); 395. D (tsc); 396. C (node app.js)

397. B (tsc); 398. C (node app.js); 399. C (tsc); 400. D (node app.js); 401. B (int); 402. A (dotnet run)

403. A (int); 404. B (dotnet run); 405. B (int); 406. D (dotnet run); 407. D (int); 408. B (dotnet run)

409. A (int); 410. D (dotnet run); 411. B (int); 412. C (dotnet run); 413. A (int); 414. B (dotnet run)

415. A (int); 416. D (dotnet run); 417. D (int); 418. B (dotnet run); 419. C (int); 420. B (dotnet run)

421. C (int); 422. C (dotnet run); 423. B (int); 424. B (dotnet run); 425. B (int); 426. C (dotnet run)

427. A (int); 428. B (dotnet run); 429. D (int); 430. A (dotnet run); 431. A (int); 432. A (dotnet run)

433. B (int); 434. D (dotnet run); 435. A (int); 436. B (dotnet run); 437. A (int); 438. A (dotnet run)

439. B (int); 440. B (dotnet run); 441. C (int); 442. B (dotnet run); 443. D (int); 444. B (dotnet run)

445. D (int); 446. B (dotnet run); 447. B (int); 448. C (dotnet run); 449. D (int); 450. D (dotnet run)

451. B (int); 452. C (dotnet run); 453. A (int); 454. C (dotnet run); 455. C (int); 456. B (dotnet run)

457. D (int); 458. A (dotnet run); 459. C (int); 460. C (dotnet run); 461. B (int); 462. A (dotnet run)

463. B (int); 464. B (dotnet run); 465. A (int); 466. B (dotnet run); 467. B (int); 468. C (dotnet run)

469. C (int); 470. B (dotnet run); 471. A (int); 472. D (dotnet run); 473. C (int); 474. C (dotnet run)

475. D (int); 476. B (dotnet run); 477. C (int); 478. C (dotnet run); 479. D (int); 480. D (dotnet run)

481. B (int); 482. B (dotnet run); 483. A (int); 484. A (dotnet run); 485. B (int); 486. A (dotnet run)

487. C (int); 488. A (dotnet run); 489. B (int); 490. B (dotnet run); 491. D (int); 492. B (dotnet run)

493. A (int); 494. C (dotnet run); 495. A (int); 496. B (dotnet run); 497. C (int); 498. C (dotnet run)

499. A (int); 500. C (dotnet run); 501. D (cargo); 502. D (go run main.go); 503. A (cargo); 504. D (go run main.go)

505. A (cargo); 506. A (go run main.go); 507. C (cargo); 508. C (go run main.go); 509. A (cargo); 510. A (go run main.go)

511. B (cargo); 512. B (go run main.go); 513. D (cargo); 514. B (go run main.go); 515. B (cargo); 516. D (go run main.go)

517. C (cargo); 518. C (go run main.go); 519. D (cargo); 520. A (go run main.go); 521. D (cargo); 522. C (go run main.go)

523. C (cargo); 524. B (go run main.go); 525. D (cargo); 526. D (go run main.go); 527. A (cargo); 528. B (go run main.go)

529. D (cargo); 530. D (go run main.go); 531. A (cargo); 532. B (go run main.go); 533. B (cargo); 534. C (go run main.go)

535. D (cargo); 536. B (go run main.go); 537. A (cargo); 538. A (go run main.go); 539. A (cargo); 540. B (go run main.go)

541. C (cargo); 542. D (go run main.go); 543. A (cargo); 544. B (go run main.go); 545. C (cargo); 546. D (go run main.go)

547. D (cargo); 548. A (go run main.go); 549. A (cargo); 550. B (go run main.go); 551. B (cargo); 552. B (go run main.go)

553. C (cargo); 554. C (go run main.go); 555. C (cargo); 556. D (go run main.go); 557. D (cargo); 558. A (go run main.go)

559. A (cargo); 560. C (go run main.go); 561. C (cargo); 562. D (go run main.go); 563. B (cargo); 564. B (go run main.go)

565. C (cargo); 566. B (go run main.go); 567. D (cargo); 568. B (go run main.go); 569. A (cargo); 570. A (go run main.go)

571. A (cargo); 572. B (go run main.go); 573. B (cargo); 574. A (go run main.go); 575. A (cargo); 576. C (go run main.go)

577. C (cargo); 578. A (go run main.go); 579. C (cargo); 580. B (go run main.go); 581. A (cargo); 582. A (go run main.go)

583. A (cargo); 584. D (go run main.go); 585. C (cargo); 586. B (go run main.go); 587. A (cargo); 588. A (go run main.go)

589. B (cargo); 590. C (go run main.go); 591. B (cargo); 592. A (go run main.go); 593. A (cargo); 594. D (go run main.go)

595. B (cargo); 596. A (go run main.go); 597. A (cargo); 598. D (go run main.go); 599. C (cargo); 600. A (go run main.go)

601. A (SELECT * FROM users_1;); 602. A (SELECT * FROM users_2;); 603. B (SELECT * FROM users_3;); 604. C (SELECT * FROM users_4;); 605. A (SELECT * FROM users_5;); 606. D (SELECT * FROM users_6;)

607. C (SELECT * FROM users_7;); 608. D (SELECT * FROM users_8;); 609. D (SELECT * FROM users_9;); 610. D (SELECT * FROM users_10;); 611. D (SELECT * FROM users_11;); 612. C (SELECT * FROM users_12;)

613. B (SELECT * FROM users_13;); 614. B (SELECT * FROM users_14;); 615. D (SELECT * FROM users_15;); 616. A (SELECT * FROM users_16;); 617. A (SELECT * FROM users_17;); 618. B (SELECT * FROM users_18;)

619. A (SELECT * FROM users_19;); 620. A (SELECT * FROM users_20;); 621. C (SELECT * FROM users_21;); 622. A (SELECT * FROM users_22;); 623. C (SELECT * FROM users_23;); 624. D (SELECT * FROM users_24;)

625. A (SELECT * FROM users_25;); 626. D (SELECT * FROM users_26;); 627. B (SELECT * FROM users_27;); 628. B (SELECT * FROM users_28;); 629. D (SELECT * FROM users_29;); 630. A (SELECT * FROM users_30;)

631. C (SELECT * FROM users_31;); 632. B (SELECT * FROM users_32;); 633. C (SELECT * FROM users_33;); 634. A (SELECT * FROM users_34;); 635. A (SELECT * FROM users_35;); 636. C (SELECT * FROM users_36;)

637. C (SELECT * FROM users_37;); 638. D (SELECT * FROM users_38;); 639. A (SELECT * FROM users_39;); 640. B (SELECT * FROM users_40;); 641. C (SELECT * FROM users_41;); 642. C (SELECT * FROM users_42;)

643. C (SELECT * FROM users_43;); 644. D (SELECT * FROM users_44;); 645. C (SELECT * FROM users_45;); 646. A (SELECT * FROM users_46;); 647. A (SELECT * FROM users_47;); 648. B (SELECT * FROM users_48;)

649. C (SELECT * FROM users_49;); 650. D (SELECT * FROM users_50;); 651. D (SELECT * FROM users_51;); 652. C (SELECT * FROM users_52;); 653. C (SELECT * FROM users_53;); 654. A (SELECT * FROM users_54;)

655. B (SELECT * FROM users_55;); 656. B (SELECT * FROM users_56;); 657. A (SELECT * FROM users_57;); 658. B (SELECT * FROM users_58;); 659. D (SELECT * FROM users_59;); 660. A (SELECT * FROM users_60;)

661. B (SELECT * FROM users_61;); 662. C (SELECT * FROM users_62;); 663. B (SELECT * FROM users_63;); 664. B (SELECT * FROM users_64;); 665. C (SELECT * FROM users_65;); 666. D (SELECT * FROM users_66;)

667. B (SELECT * FROM users_67;); 668. C (SELECT * FROM users_68;); 669. D (SELECT * FROM users_69;); 670. D (SELECT * FROM users_70;); 671. B (SELECT * FROM users_71;); 672. B (SELECT * FROM users_72;)

673. A (SELECT * FROM users_73;); 674. B (SELECT * FROM users_74;); 675. D (SELECT * FROM users_75;); 676. B (SELECT * FROM users_76;); 677. C (SELECT * FROM users_77;); 678. A (SELECT * FROM users_78;)

679. C (SELECT * FROM users_79;); 680. A (SELECT * FROM users_80;); 681. B (SELECT * FROM users_81;); 682. D (SELECT * FROM users_82;); 683. A (SELECT * FROM users_83;); 684. B (SELECT * FROM users_84;)

685. B (SELECT * FROM users_85;); 686. D (SELECT * FROM users_86;); 687. B (SELECT * FROM users_87;); 688. C (SELECT * FROM users_88;); 689. A (SELECT * FROM users_89;); 690. A (SELECT * FROM users_90;)

691. D (SELECT * FROM users_91;); 692. D (SELECT * FROM users_92;); 693. D (SELECT * FROM users_93;); 694. C (SELECT * FROM users_94;); 695. D (SELECT * FROM users_95;); 696. A (SELECT * FROM users_96;)

697. D (SELECT * FROM users_97;); 698. D (SELECT * FROM users_98;); 699. C (SELECT * FROM users_99;); 700. D (SELECT * FROM users_100;); 701. B (immutability); 702. B (JVM interoperability)

703. D (Lisp-style syntax); 704. B (concurrency on the BEAM VM); 705. D (fault-tolerant actor model); 706. C (strong type inference); 707. B (symbolic and recursive programming); 708. B (immutability)

709. A (JVM interoperability); 710. C (Lisp-style syntax); 711. A (concurrency on the BEAM VM); 712. B (fault-tolerant actor model); 713. A (strong type inference); 714. A (symbolic and recursive programming)

715. A (immutability); 716. B (JVM interoperability); 717. A (Lisp-style syntax); 718. A (concurrency on the BEAM VM); 719. C (fault-tolerant actor model); 720. B (strong type inference)

721. B (symbolic and recursive programming); 722. C (immutability); 723. C (JVM interoperability); 724. C (Lisp-style syntax); 725. B (concurrency on the BEAM VM); 726. A (fault-tolerant actor model)

727. C (strong type inference); 728. C (symbolic and recursive programming); 729. B (immutability); 730. B (JVM interoperability); 731. B (Lisp-style syntax); 732. C (concurrency on the BEAM VM)

733. D (fault-tolerant actor model); 734. A (strong type inference); 735. D (symbolic and recursive programming); 736. C (immutability); 737. D (JVM interoperability); 738. B (Lisp-style syntax)

739. C (concurrency on the BEAM VM); 740. B (fault-tolerant actor model); 741. A (strong type inference); 742. C (symbolic and recursive programming); 743. C (immutability); 744. D (JVM interoperability)

745. C (Lisp-style syntax); 746. B (concurrency on the BEAM VM); 747. B (fault-tolerant actor model); 748. B (strong type inference); 749. C (symbolic and recursive programming); 750. A (immutability)

751. D (JVM interoperability); 752. A (Lisp-style syntax); 753. A (concurrency on the BEAM VM); 754. D (fault-tolerant actor model); 755. A (strong type inference); 756. B (symbolic and recursive programming)

757. C (immutability); 758. B (JVM interoperability); 759. D (Lisp-style syntax); 760. B (concurrency on the BEAM VM); 761. C (fault-tolerant actor model); 762. D (strong type inference)

763. C (symbolic and recursive programming); 764. B (immutability); 765. A (JVM interoperability); 766. B (Lisp-style syntax); 767. D (concurrency on the BEAM VM); 768. C (fault-tolerant actor model)

769. C (strong type inference); 770. D (symbolic and recursive programming); 771. D (immutability); 772. C (JVM interoperability); 773. A (Lisp-style syntax); 774. B (concurrency on the BEAM VM)

775. B (fault-tolerant actor model); 776. C (strong type inference); 777. D (symbolic and recursive programming); 778. A (immutability); 779. B (JVM interoperability); 780. D (Lisp-style syntax)

781. A (concurrency on the BEAM VM); 782. A (fault-tolerant actor model); 783. D (strong type inference); 784. C (symbolic and recursive programming); 785. C (immutability); 786. B (JVM interoperability)

787. D (Lisp-style syntax); 788. B (concurrency on the BEAM VM); 789. A (fault-tolerant actor model); 790. B (strong type inference); 791. B (symbolic and recursive programming); 792. C (immutability)

793. B (JVM interoperability); 794. B (Lisp-style syntax); 795. D (concurrency on the BEAM VM); 796. B (fault-tolerant actor model); 797. A (strong type inference); 798. A (symbolic and recursive programming)

799. C (immutability); 800. C (JVM interoperability); 801. B (expressive scripting and web apps); 802. C (server-side web scripting); 803. C (lightweight embeddable scripting); 804. A (command-oriented scripting)

805. C (shell automation on Unix-like systems); 806. B (compiled language with Python-like feel); 807. C (Ruby-like syntax with native compilation); 808. D (expressive scripting and web apps); 809. B (server-side web scripting); 810. C (lightweight embeddable scripting)

811. A (command-oriented scripting); 812. B (shell automation on Unix-like systems); 813. B (compiled language with Python-like feel); 814. A (Ruby-like syntax with native compilation); 815. B (expressive scripting and web apps); 816. D (server-side web scripting)

817. D (lightweight embeddable scripting); 818. D (command-oriented scripting); 819. D (shell automation on Unix-like systems); 820. A (compiled language with Python-like feel); 821. B (Ruby-like syntax with native compilation); 822. B (expressive scripting and web apps)

823. A (server-side web scripting); 824. D (lightweight embeddable scripting); 825. B (command-oriented scripting); 826. A (shell automation on Unix-like systems); 827. C (compiled language with Python-like feel); 828. B (Ruby-like syntax with native compilation)

829. C (expressive scripting and web apps); 830. B (server-side web scripting); 831. D (lightweight embeddable scripting); 832. A (command-oriented scripting); 833. D (shell automation on Unix-like systems); 834. A (compiled language with Python-like feel)

835. D (Ruby-like syntax with native compilation); 836. C (expressive scripting and web apps); 837. B (server-side web scripting); 838. A (lightweight embeddable scripting); 839. B (command-oriented scripting); 840. D (shell automation on Unix-like systems)

841. D (compiled language with Python-like feel); 842. D (Ruby-like syntax with native compilation); 843. B (expressive scripting and web apps); 844. C (server-side web scripting); 845. A (lightweight embeddable scripting); 846. C (command-oriented scripting)

847. A (shell automation on Unix-like systems); 848. C (compiled language with Python-like feel); 849. C (Ruby-like syntax with native compilation); 850. D (expressive scripting and web apps); 851. B (server-side web scripting); 852. D (lightweight embeddable scripting)

853. C (command-oriented scripting); 854. A (shell automation on Unix-like systems); 855. D (compiled language with Python-like feel); 856. A (Ruby-like syntax with native compilation); 857. D (expressive scripting and web apps); 858. B (server-side web scripting)

859. D (lightweight embeddable scripting); 860. A (command-oriented scripting); 861. B (shell automation on Unix-like systems); 862. D (compiled language with Python-like feel); 863. A (Ruby-like syntax with native compilation); 864. A (expressive scripting and web apps)

865. A (server-side web scripting); 866. A (lightweight embeddable scripting); 867. B (command-oriented scripting); 868. A (shell automation on Unix-like systems); 869. C (compiled language with Python-like feel); 870. D (Ruby-like syntax with native compilation)

871. A (expressive scripting and web apps); 872. C (server-side web scripting); 873. B (lightweight embeddable scripting); 874. D (command-oriented scripting); 875. B (shell automation on Unix-like systems); 876. A (compiled language with Python-like feel)

877. D (Ruby-like syntax with native compilation); 878. B (expressive scripting and web apps); 879. C (server-side web scripting); 880. A (lightweight embeddable scripting); 881. C (command-oriented scripting); 882. C (shell automation on Unix-like systems)

883. A (compiled language with Python-like feel); 884. D (Ruby-like syntax with native compilation); 885. B (expressive scripting and web apps); 886. A (server-side web scripting); 887. C (lightweight embeddable scripting); 888. C (command-oriented scripting)

889. B (shell automation on Unix-like systems); 890. C (compiled language with Python-like feel); 891. A (Ruby-like syntax with native compilation); 892. D (expressive scripting and web apps); 893. D (server-side web scripting); 894. A (lightweight embeddable scripting)

895. A (command-oriented scripting); 896. C (shell automation on Unix-like systems); 897. D (compiled language with Python-like feel); 898. B (Ruby-like syntax with native compilation); 899. B (expressive scripting and web apps); 900. A (server-side web scripting)

901. B (Vite); 902. D (npm); 903. B (npx); 904. B (Deno); 905. D (Bun); 906. D (Vite)

907. D (npm); 908. D (npx); 909. B (Deno); 910. A (Bun); 911. B (Vite); 912. C (npm)

913. C (npx); 914. D (Deno); 915. B (Bun); 916. B (Vite); 917. C (npm); 918. A (npx)

919. D (Deno); 920. B (Bun); 921. D (Vite); 922. D (npm); 923. A (npx); 924. A (Deno)

925. B (Bun); 926. B (Vite); 927. D (npm); 928. D (npx); 929. B (Deno); 930. C (Bun)

931. B (Vite); 932. A (npm); 933. C (npx); 934. C (Deno); 935. B (Bun); 936. B (Vite)

937. D (npm); 938. B (npx); 939. A (Deno); 940. A (Bun); 941. B (Vite); 942. A (npm)

943. D (npx); 944. A (Deno); 945. C (Bun); 946. C (Vite); 947. B (npm); 948. C (npx)

949. C (Deno); 950. A (Bun); 951. D (Vite); 952. B (npm); 953. B (npx); 954. A (Deno)

955. A (Bun); 956. C (Vite); 957. A (npm); 958. A (npx); 959. D (Deno); 960. C (Bun)

961. D (Vite); 962. D (npm); 963. D (npx); 964. A (Deno); 965. A (Bun); 966. A (Vite)

967. B (npm); 968. C (npx); 969. A (Deno); 970. B (Bun); 971. C (Vite); 972. D (npm)

973. C (npx); 974. A (Deno); 975. A (Bun); 976. A (Vite); 977. A (npm); 978. C (npx)

979. D (Deno); 980. A (Bun); 981. A (Vite); 982. D (npm); 983. C (npx); 984. D (Deno)

985. B (Bun); 986. D (Vite); 987. D (npm); 988. D (npx); 989. A (Deno); 990. A (Bun)

991. B (Vite); 992. D (npm); 993. C (npx); 994. B (Deno); 995. B (Bun); 996. B (Vite)

997. B (npm); 998. B (npx); 999. A (Deno); 1000. D (Bun)