

JNTUK Online Examinations[Mid1-ACA]

1. Pick the odd man out.

- a. Scrap
- b. Waranty
- c. Labour costs
- d. Taxes**

2. Pick the odd man out

- a. Purchasing
- b. Sales**
- c. Scrap
- d. Waranty Components

3. Pick the odd man out

- a. Labour Costs**
- b. Marketing
- c. R & D
- d. Sales

4. Pick the odd man out

- a. R & D
- b. Cost of financing
- c. Waranty**
- d. Pretax profits

5. _____ refer to the costs directly related to making a product.

- a. Average Discount
- b. Gross margin
- c. Direct costs**
- d. Component costs

6. Which of the following is a direct costs.

- a. Labour-cost**
- b. R & D
- c. Sales
- d. taxes

7. Which of the following is an indirect cost?

- a. Labour-cost
- b. R & R**
- c. Scrap
- d. Waranty

8. Average selling price = Component cost + direct cost + _____

- a. Average Discount
- b. Gross margin
- c. Indirect cost
- d. Gross margin**

9. Indirect cost is also known as _____

- a. Component
- b. Prince
- c. Gross margin**
- d. Discount margin

10. Pick the odd man out.

- a. Dithering
- b. IP routing**
- c. Text processing
- d. Image rotation

11. Pick the odd man out

- a. FFT
- b. decoder
- c. text processing**
- d. encoder

12. Pick the odd man out.

- a. automobile control benchmarks
- b. multimedia benchmarks**
- c. Filter benchmarks
- d. FFT bench marks

13. Pick the odd man out.

- a. JPEG compress
- b. filtering
- c. IP routing**
- d. packet flow operations

14. Pick the odd man out.

- a. IP routing
- b. packet flow operation
- c. shortest path calculation
- d. Bezier curve calculation**

15. The time between the 'start' and the 'Completion' of an event is _____ time

- a. response**
- b. delay
- c. waiting
- d. total

16. The time between the 'start' and the 'Completion' of an event is _____ time.

- a. execution**
- b. delay
- c. waiting
- d. total

17. _____, is the total amount of work done in a given time.

- a. Kernel
- b. Kernel (or) throughput
- c. Shell
- d. throughput**

18. Pick the odd man out.

- a. response
- b. Wall-clock time
- c. elapsed time
- d. throughput**

19. Pick the odd man out.

- a. fittering**
- b. arithmetic operations
- c. pointer chasing
- d. matrix arithmetic

20. CPU time = (IC X Clock cycle time) / _____.

- a. CPI
- b. IPC

c. Clock rate

- d. Clock tick

21. CPU time = (_____ X Clock cycle time) / clock rate.

- a. CPI
- b. IPC

c. IC

- d. Clock tick

22. Clock Cycles = IC / _____.

- a. CPI
- b. IPC**
- c. Clock rate
- d. Clock tick

23. Clock cycles = _____ X CPI.

- a. IC**
- b. Clock rate
- c. IPC
- d. Clock tick

24. Clock cycles = _____ / IPC.

- a. CPI
- b. IC**
- c. Clock rate

d. Clock tick

25. CPU time = CPU Clock cycles for a program X _____.

a. Clock cycle time

b. Clock rate

c. IC

d. CPI

26. CPU time = CPU clock cycles for a program / _____.

a. Clock cycle time

b. Clock rate

c. IC

d. CPI

27. Instruction count is a synonym for _____.

a. CPI

b. Clock rate

c. instruction path length

d. Clock tick

28. CPI = CPU clock cycles for a program / _____.

a. Clock tick

b. IPC

c. Clock rate

d. Instruction count

29. Clock cycles = IC X _____.

a. CPI

b. IPC

c. Clock rate

d. Clock tick

30. Which of the following is an example for 'Register-memory' Architecture?

a. Alpha

b. Intel 80 X 86

c. MIPS

d. Power PC

31. Maximum number of operands allowed per instruction are _____ (in registerregister architecture)

a. 3

b. 2

c. 1

d. 4

32. Maximum number of operands allowed per instruction are _____ (in registermemory architecture)

a. 3

b. 2

c. 1

d. 4

33. Maximum number of operands allowed per instruction are _____ (in memory-memory architecture, 2 memory addresses)

a. 3

b. 2

c. 5

d. 4

34. Maximum number of operands allowed per instruction are _____ (in memory-memory architecture, 3 memory addresses).

a. 3

b. 2

c. 5

d. 4

35. The instruction 'Add' is related to _____ Instruction set architecture class.

a. Stack

b. Accumulator

c. register-memory

d. Load-store

36. The instruction 'Add B' is related to _____ Instruction set architecture class.

a. Stack

b. Accumulator

c. register-memory

d. Load-store

37. The instruction 'Add R₃, R₁, B' is related to _____ Instruction set architecture

class

a. Stack

b. Accumulator

c. register-memory

d. Load-store

38. The instruction 'Add R₃, R₁, R₂' is related to _____ Instruction set architecture class.

a. Stack

b. Accumulator

c. register- memory

d. Load-store

39. Which of the following is an example for 'Register-Register' Architecture?

a. Alpha

b. Intel 80 X 86

c. IBM 360

d. Motorola 68000

40. In R-M architecture number of memory addresses per instruction are _____.

a. 0

b. 1

c. 2

d. 3

41. In M-M architecture number of memory address per instruction are _____.

a. 1

b. 2

c. 3

d. 2 (or) 3

42. In R-R architecture number of memory addresses per instruction are _____.

a. 0

b. 1

c. 2

d. 3

43. 1 byte = _____ bits

a. 8

b. 16

c. 32

d. 64

44. Which of the following is an example for 'Register' addressing mode?

a. Add R₄, R₃

b. Add R₄, # 3

c. Add R₄, 100(R₁)

d. Add R₁, (R₂)

45. Half word = _____ bits

a. 8

b. 16

c. 32

d. 64

46. 1 word = _____ bits

a. 8

b. 16

c. 32

- d. 64
47. 1 Double word = _____ bits
a. 8
b. 16
c. 32
d. 64
48. 1 nible = _____ bits
a. 32
b. 16
c. 8
d. 4
49. 'Add R₄, R₃' is an example for _____ addressing mode.
a. Immediate
b. Register
c. Autoincrement
d. Scaled
50. 'Add R₁, - (R₂)' is an example for _____ addressing mode
a. Scaled
b. Auto decrement
c. Auto increment
d. Direct
51. 'Add R₁, 100 (R₂)[R₃]' is an example for _____ addressing mode.
a. Scaled
b. Auto decrement
c. Auto increment
d. Indexed
52. 'Add R₁(1001)' is an example for _____ addressing mode
a. absolute
b. Indexed
c. Scaled
d. Memory Indirect
53. 'Add R₁, (R₃)' is an example for _____ addressing mode
a. Auto Increment
b. Memory Indirect
c. Scaled
d. Auto decrement
54. 'Add R₁, (R₂)+' is an example for _____ addressing mode
a. Indexed
b. Direct
c. Auto increment
d. Auto decrement
55. 'Add R₄, # 3' is an example for _____ addressing mode.
a. Direct
b. Indexed
c. Immediate
d. Scaled
56. 'Add R₄, 100(R₁)' is an example for _____ addressing mode.
a. Register Indirect
b. Indexed
c. Scaled
d. Displacement
57. 'Add R₄, (R₁)' is an example for _____ addressing mode.
a. Direct
b. Indexed
c. Register Indirect
d. Scaled
- 58. 'Add R₃, (R₁+R₂)' is an example for _____ addressing mode**
a. Scaled
b. Indexed
c. Direct
d. absolute
59. 'Add R₁(1001)' is an example for _____ addressing mode
a. Memory Indirect
b. Auto Increment
c. Direct
d. Auto decrement
60. The size of _____ operand is 8 bits
a. int
b. char
c. floating point
d. double precision floating point
61. The size of _____ operand is 16 bits.
a. single precision floating point
b. int
c. double precision floating point
d. Char
62. In 'packed decimal' _____ decimal digits are packed into each byte.
a. 3
b. 2
c. 4
d. 5
63. In 'binary-coded decimal' _____ decimal digits are packed into each byte
a. 3
b. 4
c. 5
d. 2
64. In 'binary-coded decimal' _____ bits are used to encode the values 0-9
a. 2
b. 5
c. 4
d. 6
65. The size of 'Char' operand is _____ bits
a. 8
b. 16
c. 32
d. 64
66. The size of 'int' operand is _____ bits.
a. 8
b. 16
c. 32
d. 64
67. The size of 'single precision floating point' is _____ bits.
a. 8
b. 16
c. 32
d. 64
68. The size of 'doubled-precision floating point' operand is _____ bits
a. 8
b. 16
c. 32
d. 64
69. In 'packed decimal' _____ decimal digits are packed into each byte
a. 3
b. 2

c. 4
d. 5

70. The average % of 'Call' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 1 %
- b. 12 %
- c. 13 %
- d. 14 %

71. The average % of 'return' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 12 %
- b. 1 %
- c. 20 %
- d. 30 %

72. The average % of 'and' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 36 %
- b. 6 %
- c. 46 %
- d. 56 %

73. The average % of 'Sub' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 50 %
- b. 40 %
- c. 5 %
- d. 30 %

74. The average % of 'move R-R' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 25 %
- b. 35 %
- c. 33 %
- d. 4 %

75. The average % of 'load' instruction in the integer programs running on the popular Intel 80 X 86 is _____

- a. 22 %
- b. 4 %
- c. 1 %
- d. 2 %

76. The average % of 'Conditional branch' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 10 %
- b. 20 %
- c. 60 %
- d. 50 %

77. The average % of 'Compare' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 10 %
- b. 60 %
- c. 16 %
- d. 40 %

78. The average % of 'store' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 30 %
- b. 40 %
- c. 50 %
- d. 12 %

79. The average % of 'add' instruction in the integer programs running on the popular Intel 80X86 is _____

- a. 8 %
- b. 80 %
- c. 70 %
- d. 50 %

80. _____ uses 'Compare and branch' method for evaluating branch conditions.

- a. Alpha
- b. MIPS
- c. Power PC
- d. VAX

81. Power PC uses _____ method for evaluating branch conditions.

- a. Condition Code
- b. Condition register
- c. Compare and branch
- d. CC and Condition register

82. _____ uses condition code method for evaluating branch conditions.

- a. 80X86
- b. Alpha
- c. MIPS
- d. VAX

83. _____ uses 'Condition code' method for evaluating branch conditions

- a. Alpha
- b. SPARC
- c. MIPS
- d. VAX

84. _____ uses 'Condition register' method for evaluating branch Conditions

- a. SPARC
- b. PA-RISC
- c. Alpha
- d. VAX

85. Pick the odd man out (branch condition method)

- a. Alpha
- b. 80X86
- c. ARM
- d. Power PC

86. Pick the odd man out. (branch condition methods)

- a. ARM
- b. MIPS
- c. Power PC
- d. SPARC

87. Pick the odd man out. (branch condition methods)

- a. Power PC
- b. SPARC
- c. VAX
- d. Super H

88. Pick the odd man out. (branch condition methods).

- a. SPARC
- b. Super H
- c. 80X86
- d. PA-RISC

89. Which of the following is simple method for evaluating branch conditions.

- a. CC
- b. Condition register
- c. Compare and branch
- d. Can't say

90. Pick the odd man out (encoding Instruction set)

- a. Alpha
- b. Intel 80X86**
- c. ARM
- d. SPARC

91. Pick the odd man out (encoding Instruction set)

- a. Alpha
- b. Power PC
- c. IBM 360/70**
- d. SPARC

92. In _____ 'fixed format' is used for encoding the instruction set.

- a. VAX
- b. MIPS**
- c. Intel 8086
- d. Thumb

93. In _____ 'hybrid approach' is used for encoding the instruction set.

- a. Super H
- b. SPARC
- c. IBM 360/70**
- d. Power PC

94. Pick the odd man out. (encoding Instruction set)

- a. VAX
- b. MIPS
- c. ARM
- d. Alpha**

95. In _____ 'Variable format' is used for encoding the instruction set.

- a. VAX
- b. Alpha
- c. ARM
- d. MIPS**

96. In _____ 'Variable format' is used for encoding the instruction set.

- a. ARM
- b. Intel 80X86**
- c. MIPS
- d. Power PC

97. In _____ 'fixed format' is used for encoding the instruction set.

- a. VAX
- b. Intel 8086
- c. Alpha**
- d. IBM 360/70

98. In _____ 'fixed format' is used for encoding the instruction set

- a. VAX
- b. Intel 8086
- c. MIPS 16
- d. Power PC**

99. In _____ 'fixed format' is used for encoding the instruction set

- a. SPARC**
- b. VAX
- c. Intel 8086
- d. IBM 360/70

100. In Compilers 'High-level optimizations' phase is _____

- a. Machine Independent
- b. Slightly machine depend
- c. Largely machine Independent**
- d. Machine dependent

101. In Compilers 'High-level optimizations' phase is _____

- a. Language Independent

b. Largely Language Independent

c. Largely Language dependent

d. Some what Language dependent

102. Pick the odd man out.

- a. Compiler
- b. Interpreter
- c. OS**
- d. Assembler

103. Front end of the compiler is _____

- a. Language dependent
- b. Language Independent
- c. Somewhat Language dependent
- d. Can't say

104. Front end of the Compiler is _____

- a. Machine dependent
- b. Machine Independent**
- c. Slightly machine dependent
- d. Can't say

105. _____ returns the address of a variable.

- a. &
- b. *
- c. +
- d.

106. _____ dereference a pointer

- a. &
- b. ***
- c. +
- d.

107. _____ is used for activation records.

- a. Tree
- b. Queue
- c. Stack**
- d. Heap

108. _____ is used to allocate dynamic object

- a. Tree
- b. Queue
- c. Stack
- d. Heap**

109. The global data area is used to allocate statically declared objects, such as _____

- a. global variables
- b. Constants
- c. local variables

d. global variables and constants

110. Choose the correct statement int *p, a, b, *q

- a. p= &a;
- b. b=q;
- c. a=p+q;
- d. a=p;**

111. _____ is used to allocate local variables.

- a. Heap**
- b. Stack**
- c. Heap & Stack
- d. Queue

112. In Compilers _____ is 'somewhat language dependent' & 'largely machine independent'

- a. Front end
- b. Global optimization**
- c. High-level optimization
- d. Code generator

113. In Compilers _____ is language independent

- a. Front end
- b. High-level optimization**

c. Code generator

d. Global optimizer

114. Pick the wrong statement int *p, a,b, *q;

a. p= &a;

b. q= &a;

c. *p= a;

d. a=p;

115. In Compilers 'Global' optimizer' phase is _____

a. Language dependent

b. Some what language dependent

c. Small language dependencies

d. Language independent

116. In Compliers 'Global Optimization' phase is _____

a. Machine independent

b. Machine dependencies slight

c. Largely machine independent

d. Highly machine dependent

117. In Compilers 'Code generator' phase is _____

a. machine independent

b. machine dependencies slight

c. Highly machine dependent

d. Largely machine independent

118. In Compilers 'Code generator' phase is _____

a. Language dependent

b. Somewhat language dependent

c. Small language dependencies

d. Language independent

119. In Compilers _____ is Language dependent & machine independent.

a. Front end

b. High level optimizations

c. Global optimizer

d. Code generator

120. 'Instruction j tries to write an operand before it is written by i' is _____ hazard

a. RAW

b. WAW

c. WAR

d. RAR

121. 'Instruction j tries to write a destination before it is read by i. So i incorrectly gets the new value' is _____ hazard

a. RAW

b. WAW

c. WAR

d. RAR

122. WAR stands for _____

a. Read and write

b. Read after write

c. Write after read

d. Write and read

123. RAR stands for _____

a. Random access memory

b. Read and Read

c. Random and Random

d. Read after read

124. 'Instruction j tried to read a source before i writes it, so j incorrectly gets the old value' is _____ hazard.

a. RAW

b. WAW

c. WAR

d. RAR

125. An _____ between instruction i and instruction j occurs when instruction j writes a register or memory location that instruction i reads

a. antidependence

b. output dependence

c. input dependence

d. Output/input dependence

126. An antidependence between instruction i and instruction j occurs when instruction j _____ a register or memory location that instruction i _____

a. -- reads, reads

b. writes, reads

c. writes, writes

d. reads, writes

127. Which of the following is not a hazard?

a. RAW

b. WAW

c. WAR

d. RAR

128. RAW stands for _____

a. reads after write

b. read and write

c. write and read

d. write after read

129. WAW stands for _____

a. Write and write

b. Write after write

c. Write and write/Read

d. Write/Read and write

130. Consider the following code sequence. DIV.D , , , F0, F2, F4 ADD.D , , , , , F6, F0, F8 S.D , , , , , , , , , , , , , , , , , , , F6, 0(R1) SUB.D , , , , , F8, F10, F14 MUL.D , , , , , F6, F10, F8 The above code sequence includes _____ hazards.

a. WAR

b. WAW

c. WAR and WAW

d. Can't say

131. Consider the following code sequence. DIV.D , , , F0, F2, F4 ADD.D , , , , , F6, F0, F8 S.D , , , , , , , , , , , , , , , , , , , F6, 0(R1) SUB.D , , , , , F8, F10, F14 MUL.D , , , , , F6, F10, F8 There is an antidependence between the ADD.D and the _____

a. S.D

b. SUB.D

c. MUL.D

d. DIV.D

132. If a load and a store access the same address then the store is before the load in program order and interchanging them results in a _____ hazard.

a. WAR

b. RAR

c. RAW

d. WAW

133. _____ hazard are avoided by executing an instruction only when its

operands are available.

- a. WAW
- b. WAR
- c. RAR
- d. RAW

134. _____ hazards which arise from name dependences are eliminated by register naming

- a. RAR
- b. WAR
- c. WAW
- d. WAR and WAW

135. Out-of-order execution introduces the possibility of _____ hazard.

- a. WAR
- b. RAR
- c. RAW
- d. RAR and RAW

136. Out-of-order execution introduces the possibility of _____ hazard.

- a. WAW
- b. RAR
- c. RAW
- d. RAR and RAW

137. n-bit Counter can take on values between _____ and $2^n - 1$

- a. 1
- b. 0
- c. 2
- d. n

138. n-bit Counter, can take on values between 0 and _____

- a. n
- b. 2^{n-2}
- c. 2^n
- d. 2^{n-1}

139. If a load and a store access the same address then the load is before the store in program order and interchanging them results in a hazard.

- a. WAR
- b. RAR
- c. RAW
- d. WAW

140. Consider the following code sequence.

DIV.D , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,

F14 MUL.D , , , , , F6, F10, F8 There is a true data dependence between S. D

and _____
a. SUB.D
b. ADD.D
c. MUL.D
d. DIV.D

141. Consider the following code sequence.

DIV.D , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,

F14 MUL.D , , , , , F6, F10, F8 There is a true data dependence between the SUB.D and _____

- a. S.D
- b. ADD.D
- c. MUL.D
- d. DIV.D

142. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 Leading to a WAR hazard on the use of _____

- by ADD.D
- a. F0
 - b. F8
 - c. F6
 - d. F0 (or) F6

143. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 Leading to a WAR hazard on the use of F8 by _____

- a. SUB.D
- b. MUL.D
- c. ADD.D
- d. Can't say

144. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 There is a true data dependence between DIV.D

and _____
a. ADD.D
b. S.D
c. SUB.D
d. MUL.D

145. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 There is an output dependence between the

ADD.D and the _____
a. MUL.D
b. SUB.D
c. S.D
d. DIV.D

146. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 There is an antidependence between the SUB.D

and the _____
a. S.D
b. ADD.D
c. MUL.D
d. DIV.D

147. Consider the following code sequence.

DIV.D , , , , , F0, F2, F4 ADD.D , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 There is an output dependence between the

MUL.D and the _____
a. S.D
b. SUB.D
c. ADD.D
d. DIV.D

148. Consider the following code sequence. DIV.D , , ,
,,, , F0, F2, F4 ADD.D , ,
,,, , F6, F0, F8 S.D , , , , , , , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 Leading to a _____
hazard on the use of F8 by ADD.D

- a. RAR
- b. RAW
- c. WAW
- d. WAR

149. Consider the following code sequence. DIV.D , , ,
,,, , F0, F2, F4 ADD.D , ,
,,, , F6, F0, F8 S.D , , , , , , , , , , F6, 0(R1)
SUB.D , , , , , F8, F10,
F14 MUL.D , , , , , F6, F10, F8 Leading to a _____
hazard, since the

ADD.D may finish later than the MUL.D

- a. WAW
- b. RAW
- c. RAR
- d. WAR

150. If Memory access time = 0.9X100 ns, Clock Cycle
time = 1.25 ns, then

miss penalty =?

- a. 72 Cycles
- b. 120 Cycles
- c. 100 cycles
- d. 127 Cycles

151. If misses per instruction=0.01, miss penalty=100
then Cache CPI=?

- a. 2.0
- b. 1.0
- c. 3.0
- d. 4.0

152. Miss penalty=(Memory access time)/ _____.

- a. Misses per instruction
- b. Clock cycles
- c. Cache CPI
- d. CPI

153. If Memory access time = 100 ns, Clock cycle
time = 1 ns. then miss

penalty=?

- a. 75 Cycles
- b. 72 Cycles
- c. 100 Cycles
- d. 120 Cycles

154. If memory access time = 100 ns, Clock Cycle
time = 0.83 ns, then miss

penalty =?

- a. 72 Cycles
- b. 100 Cycles
- c. 125 Cycles
- d. 120 Cycles

155. Misses per instruction X miss penalty = _____

- a. Cache CPI
- b. Clock cycle
- c. Clock tick
- d. Misses per instruction

156. Cache CPI = _____ X Miss penalty.

- a. Clock tick
- b. Misses per instruction
- c. Memory access tick
- d. Clock cycle

157. Cache CPI = Misses per Instruction X _____

- a. Clock tick
- b. Clock cycle
- c. Miss penalty
- d. Memory access time

158. = _____

- a. Clock tick
- b. Cache CPI
- c. CPI

d. Miss Penalty

159. Miss penalty = ()/Clock cycles

- a. Memory access time
- b. Misses per instruction
- c. CPI

d. Cache CPI

160. The Complexity of dependence analysis arises
because of _____ in
FORTRAN.

- a. Pass by reference
- b. Pass by name
- c. Pass by value
- d. Can't says

161. The Complexity of dependence analysis arises
because of pass by
reference in _____

- a. LISP
- b. PROLOG
- c. FORTRAN
- d. LISP (or) PROLOG

162. The Complexity of dependence analysis arises
because of the presence of
_____ and pointers in languages like C or C++

- a. int
- b. arrays
- c. float
- d. Char

163. the Complexity of dependence analysis arises
because of the presence of
arrays and _____ in languages like C or C++.

- a. int
- b. float
- c. Pointers
- d. Char

164. The Complexity of dependence analysis arises
because of the presence of
arrays and pointers in languages like _____

- a. FORTRAN
- b. C
- c. C++
- d. C or C++

165. VLIW stand for _____

- a. Very long instruction word
- b. Very long immediate word
- c. Very limited instruction word
- d. Very limited & immediate word

166. VLIW instructions are very wide id., _____ to
128 bits or more.

- a. 32 bits
- b. 64 bits
- c. 125 bits
- d. 62 bits

167. VLIW instructions are very wide ie., 64 bits to
_____ (or) more

- a. 1024
- b. 512
- c. 128

d. 256

168. Local scheduling is related to _____ blocks

- a. n
- b. 5
- c. 2
- d. 1

169. Global scheduling is related to _____ blocks

- a. n
- b. 5
- c. 2
- d. 1

170. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an output dependence from S₁ to S₄ based on _____

- a. X[i]
- b. Y[i]
- c. Z[i]
- d. Can't say

171. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an output dependence from _____ based on Y[i]

- a. S₁ to S₂
- b. S₁ to S₃
- c. S₁ to S₄
- d. S₂ to S₃

172. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an antidependence from _____ based on X[i]

- a. S₁ to S₄
- b. S₁ to S₃
- c. S₁ to S₂
- d. Can't say

173. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an antidependence from _____ based on Y[i]

- a. S₂ to S₃
- b. S₂ to S₄
- c. S₁ to S₄
- d. S₃ to S₄

174. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an antidependence from S₃ to S₄ based on _____

- a. Y[i]
- b. X[i]
- c. Z[i]
- d. Can't say

175. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is a true dependence from S₁ to S₃ because of _____

- a. Y[i]
- b. X[i]
- c. Z[i]
- d. Can't says

176. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is a true dependence from S₁ to S₄ because of _____

- a. X[i]
- b. Y[i]
- c. Z[i]
- d. Can't says

177. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } The dependences will force S₃ and S₄ to wait for _____ to complete.

- a. S₁ (or) S₂
- b. S₂
- c. S₁
- d. Can't says

178. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an _____ from S to S₂, based on X[i]

- a. antidependence
- b. output dependence
- c. Input/output dependence
- d. Can't says

179. Consider the following for loop. for (i=1; i =100;
i++) { Y[i] = X[i] /
C; /* S1 */ X[i] = X[i] + C; /* S2 */ Z[i] = Y[i] + C; /* S3 */
Y[i] = C - Y[i] ; /

* S4 */ } There is an antidependence from S₁ to S₂ based on _____

- a. Y[i]
- b. X[i]
- c. Z[i]
- d. Can't say

180. The Component of IA-64 register state is 128 64-bit _____ registers is 128 64-bit _____ registers.

- a. Predicate
- b. branch registers
- c. general-purpose
- d. floating-point

181. In IA-64 floating-point registers, it provides _____ extra exponent bits over the standard IEEE format.

- a. 1
- b. 2
- c. 3
- d. 4

182. In IA-64 Register model total no. of general purpose registers are _____

- a. 128
- b. 127
- c. 126
- d. 125

183. In IA-64 Register model the size of general purpose registers is _____ bits

- a. 60
- b. 64
- c. 32
- d. 128

184. In IA-64 Register model the actual size of general-purpose registers is _____ bits

- a. 32
- b. 64
- c. 65
- d. 128

185. IA-64 is developed by _____

- a. Intel
- b. IBM
- c. ORACLE
- d. Can't say

186. IA-64 is a _____ style Architecture

- a. CISC
- b. RISC
- c. CISC(or)RISC
- d. Can't say

187. The IA-64 is a RISC style & _____ instruction set Architecture.

- a. R-M
- b. M-M
- c. R-R
- d. Can't say

188. In ILP 'The major disadvantage of supporting speculation in hardware is the Complexity' is _____

- a. True
- b. False
- c. T or F
- d. Can't say

189. In ILP the major disadvantage of supporting speculation in hardware is 'it requires additional hardware resources' is _____

- a. False
- b. True
- c. T or F
- d. Can't say

190. In IA-64 Register model the size of Predicate registers is _____ bits

- a. 1
- b. 2
- c. 3
- d. 4

191. In IA-64 Register model the size of branch registers is _____ bits

- a. 74 bit
- b. 64 bit
- c. 84
- d. 94

192. In IA-64 the total no. of predicate register are _____

- a. 60
- b. 64
- c. 62
- d. 63

193. In IA-64 the total no. of branch registers are _____

- a. 5
- b. 6
- c. 8
- d. 7

194. In IA-64 Register model the size of floating-point registers is _____ bits

- a. 60
- b. 80
- c. 81
- d. 82

195. In IA-64 floating-Point registers, it provides 2 extra exponent bits over the standard _____ IEEE format.

- a. 60 bit
- b. 70 bit
- c. 80 bit
- d. 90 bit

196. The Component of IA-64 register state is 128 82 bit _____ register

- a. Predicate
- b. branch
- c. general-purpose
- d. floating point

197. The Component of IA-64 register state is 64 1 bit _____ registers.

- a. Predicate
- b. branch
- c. general-purpose
- d. floating-point

198. The Component of IA-64 register state is 8 1 bits _____ registers.

- a. Predicate
- b. branch
- c. floating-point
- d. general-purpose

199. In IA-64 the total no. of floating point registers are _____

- a. 128
- b. 127
- c. 126
- d. 125