

1. Which of the following is 66 in decimal represented in hexadecimal?

- A. 0x42
- B. 0x52
- C. 0x66
- D. 0x49

Correct: 0x42

2. Which of the following is 0x11 in binary (8-bit format)?

- A. 00010001
- B. 00100010
- C. 11110001
- D. 01001000

Correct: 00010001

3. 11110000_2 corresponds to which octal value?

- A. 0o360
- B. 0o170
- C. 0o340
- D. 0o716

Correct: 0o360

4. We have two 16-bit 2's complement numbers with patterns 01x1x0x1xx000xxx and 100xxxx00xxx0x10. Could their sum possibly overflow?

- A. Yes. Any addition of positive and negative numbers can overflow.
- B. Yes. If both have their most significant bit set to 1, an overflow is guaranteed.
- C. No. A positive plus a negative cannot exceed the 16-bit 2's complement range.
- D. Yes. It always overflows when high bits (x's) are filled with 1.

Correct: No, it cannot overflow because one is positive (leading bit 0) and the other is negative (leading bit 1).

5. Which is the correct binary normalized form (roughly) for $2^{(16/15)}$?

(Note: These are illustrative approximations.)

- A. $1.0000112 \times 2^{11.0000112} \times 2^1$
- B. $1.1000002 \times 2^{21.1000002} \times 2^2$
- C. $1.0000112 \times 2^{21.0000112} \times 2^2$
- D. $1.1000002 \times 2^{11.1000002} \times 2^1$

Correct: $1.0000112 \times 2^{11.0000112} \times 2^1$

(Because $2^{(16/15)} \approx 2.07$, which normalizes to something just over 2.0, i.e., $1.\text{something} \times 2^1$.)

6. Which of the following is the bias for the 8-bit exponent field in the standard 32-bit IEEE 754 format?

- A. 127
- B. 1023
- C. 255
- D. 128

Correct: 127

7. What is the minimum gap (difference) between two adjacent subnormal (denormalized) numbers in the 32-bit IEEE 754 format?

- A. 2^{-126}
- B. 2^{-127}
- C. 2^{-149}
- D. 2^{-23}

Correct: 2^{-149}

8. If we have a 10-bit floating-point format (1 sign bit + e exponent bits + f fraction bits = 10), which of the following exponent/fraction splits is consistent with $33/256 = 0.1289\dots$ being exactly representable as 0010000001_2 ?

- A. e = 4 bits, f = 5 bits
- B. e = 3 bits, f = 6 bits
- C. e = 2 bits, f = 7 bits
- D. e = 5 bits, f = 4 bits

Correct: e = 4 bits, f = 5 bits

9. Which is the bitwise AND of 0101_2 and 1010_2 ?

- A. 0000
- B. 0010
- C. 1000
- D. 1111

Correct: 0000

10. Which is the bitwise XOR of 0101_2 and 1010_2 ?

- A. 0000
- B. 0010
- C. 1111
- D. 1110

Correct: 1111

11. Which answer correctly lists an example of a zero-operand, one-operand, and two-operand instruction (in that order)?

- A. (zero) `halt`; (one) `negq %rdi`; (two) `movq %rax, %rbx`
- B. (zero) `ret`; (one) `call .abs`; (two) `pushq %rbp, %rax`

- C. (zero) `popq %rbx`; (one) `jmp .L1`; (two) `addq (%rbx)`
D. (zero) `call .abs`; (one) `halt`; (two) `movq %rax, %rbx`

Correct (example): A. Because “halt” takes no explicit operands, “negq %rdi” takes one, and “movq %rax, %rbx” takes two.

12. Which two pipeline stages are needed by every instruction in a standard 6-stage sequential CPU pipeline?

- A. IF (Instruction Fetch) and MEM (Memory Access)
B. IF (Instruction Fetch) and WB (Write-Back)
C. ID (Instruction Decode) and EX (Execute)
D. IF (Instruction Fetch) and ID (Instruction Decode)

Correct (commonly): Instruction Fetch (IF) and Instruction Decode (ID) are always used.

13. A simple one-bit branch predictor records only the outcome of the most recent branch (taken or not taken) and uses that to predict the next time. Suppose the last five branch outcomes were: *taken, taken, taken, taken, not taken*. How would this one-bit predictor predict the next branch?

- A. It predicts taken, because four out of five were taken.
B. It predicts not taken, because the most recent outcome was not taken.
C. It predicts taken, because it uses a saturation counter.
D. It predicts not taken, because once a branch is not taken, the predictor always stays that way.

Correct Answer: B. (A one-bit predictor simply records the *most recent* outcome, which was “not taken.”)

14. In a typical Linux or Unix shell, pressing *Ctrl+Z* while a process is running in the foreground places that process into which state?

- A. Running
B. Stopped (suspended)
C. Terminated (finished)
D. Zombie

Correct Answer: B. (Ctrl+Z suspends the foreground job, making it “stopped.”)

15. Cache blocking (also known as loop tiling) is a software optimization technique primarily intended to improve which characteristic of a program’s execution?

- A. Concurrency
B. Security
C. Locality
D. Parallelism

Correct Answer: C. (Cache blocking increases *data locality* by grouping accesses so that cache lines are reused more effectively.)

16. An application is 90% parallelizable and takes 1.5 hours on a single processor. Assuming an ideal scenario with unlimited processors, what is the lowest possible execution time?

- A. 0.15 hours (9 minutes)
- B. 0.75 hours (45 minutes)
- C. 1.00 hour (60 minutes)
- D. 1.35 hours (81 minutes)

Correct Answer: A. (By Amdahl's Law, the 90% parallel portion can shrink to nearly 0, leaving 10% serial $\rightarrow 0.10 \times 1.5 \text{ hours} = 0.15 \text{ hours}$.)

17. A processor runs at 1 GHz. Which of the following is its cycle time (time per clock cycle)?

- A. 1 nanosecond (1 ns)
- B. 1 microsecond (1 μ s)
- C. 0.5 nanosecond (0.5 ns)
- D. 10 nanoseconds (10 ns)

Correct Answer: A. (1 GHz means 1 billion cycles per second $\rightarrow 1/1\text{e}9 \text{ second} = 1 \text{ ns}$.)

18. In Linux, switching from one process to another has a higher overhead than switching between threads of the same process. Which of the following is the fundamental reason?

- A. Process switching requires changing the address space and possibly flushing/reloading memory mappings, while a thread switch does not.
- B. Thread switching always involves loading the entire kernel from disk.
- C. Process switching never uses hardware virtualization features.
- D. Thread switching must recompile the code on the fly, but process switching does not.

Correct Answer: A. (A process switch typically changes the virtual address space, requiring more overhead than switching between threads in the same process.)

19. Mark-sweep garbage collection identifies "root" references to begin the marking phase. Typically, which of the following are considered root references in a running program?

- A. All objects in the heap, plus any local variables stored in the stack
- B. Global variables, pointers in registers and on the current thread's call stack (e.g., local pointers)
- C. Only arrays in the heap
- D. Only global variables with static linkage

Correct Answer: B. (Root references usually include pointers in CPU registers, on the active stack frames, and in global/static storage.)

20. Two caches have the same total capacity and same line size. One is direct-mapped; the other is 2-way set-associative. Which statement about index bits is correct?

- A. The direct-mapped cache has fewer index bits because each set holds multiple lines.
- B. The 2-way set-associative cache has fewer index bits because each set holds more than one line.
- C. They have the same number of index bits, because capacity and line size are unchanged.
- D. The 2-way set-associative cache always has more index bits.

Correct Answer: B. (If a cache is 2-way associative, each set contains 2 lines, so the total number of sets is cut in half compared to a direct-mapped design of the same size and line size. Fewer sets \Rightarrow fewer index bits.)

21. Assume a simple one-level paging system and a TLB. If a load instruction causes a TLB miss but no page fault, how many physical memory accesses might be required to complete that load?

A. 1 access total
B. 2 accesses total (one for the page table, one for the actual data)
C. 3 accesses total (page directory, page table, data)
D. 0 accesses, because the TLB handles everything

Correct Answer: B. (On a TLB miss, the CPU must read the page-table entry from memory to update the TLB, then re-attempt the load from the data address.)

22. Two threads in a single process each have a local variable called **a**. Which of the following is true?

A. They share the same memory location for **a**.
B. Each local variable **a** resides in a separate stack frame (or register), so they are different.
C. They collide in the same register, causing race conditions.
D. They refer to the same address in the global data section.

Correct Answer: B. (Each thread's function call stack is separate, so each local variable **a** is distinct.)

23. Suppose one thread does $Y = 1$ (on a shared integer **Y** initially 0) while another thread, concurrently, does $Y = Y + 1$. If there are no synchronization operations, which final value(s) of **Y** are possible?

A. Only 1
B. Only 2
C. Either 1 or 2
D. 0, 1, or 2

Correct Answer: C. If the second thread reads $Y=0$ before the first sets $Y=1$, then $Y = 0 + 1 = 1$. If the second thread reads $Y=1$ after the first sets $Y=1$, then $Y = 1 + 1 = 2$. The initial value 0 disappears once one of the writes occurs, so 0 is not an outcome.

24. Which of the following is the hexadecimal representation of the decimal value 92?

A. 0x5C
B. 0x92
C. 0x3A
D. 0x60

(Correct: A)

25. Which of the following is the correct hexadecimal representation of the binary value 1110110111?

- A. 0x3D7
- B. 0x3B7
- C. 0xEDB
- D. 0x76E

(Correct: B; $111011011_2 = 0x3B7$)

26. The octal value 273 is to be represented in binary. Which choice is correct?

- A. 10111011_2
- B. 010111011_2
- C. 011101011_2
- D. 11101011_2

(Correct: B; $273_8 = 010111011_2$)

27. Given the registers before the instruction:

`%esi = 0xA2E058`

`%ebx = 0x800`

`%edx = 0x0`

After executing `lea (%esi, %ebx, 8), %edx`, what is the new value in `%edx`?

- A. 0xA32058
- B. 0xA2E458
- C. 0xA2E0D8
- D. 0xA2E058

(Correct: A; $\%edx = \%esi + 8 \times \%ebx = 0xA2E058 + 0x4000 = 0xA32058$)

28. Your ISA supports two instruction types, both requiring 8 possible arithmetic/logic operations. What is the minimum number of bits needed for the Opcode field?

- A. 1 bit
- B. 3 bits
- C. 4 bits
- D. 8 bits

(Correct: B, since we need at least 3 bits to represent 8 operations.)

29. Type A instructions have a 4-byte format and contain an immediate address Imm. The machine is byte-addressable with total memory 2^{20} bytes (1 MB). How many bits do we need for the Imm field so it can address any byte location in memory?

- A. 8 bits
- B. 16 bits
- C. 20 bits
- D. 32 bits

(Correct: C, if the immediate must be able to specify an address in 2^{20} bytes.)

30. Given that the first (most significant) bit indicates whether an instruction is Type A (0) or Type B (1), and we use the bits from (a) and (b), what is the maximum number of registers the ISA can support?

A. 8 registers
B. 16 registers
C. 32 registers
D. 64 registers

31. A pipelined processor has 5 stages with these stage delays (in nanoseconds):

Stage 1: 28 ns

Stage 2: 59 ns

Stage 3: 23 ns

Stage 4: 34 ns

Stage 5: 36 ns

Each pipeline register (between adjacent stages) adds 1 ns of delay. The clock cycle must accommodate the slowest stage plus one pipeline-register overhead. What is the processor's cycle time?

A. 36 ns
B. 59 ns
C. 60 ns
D. 181 ns

Correct Answer: C (60 ns).

(Explanation: The slowest stage is 59 ns, plus 1 ns for the pipeline register = 60 ns total.)

32. On that 5-stage pipelined processor (60 ns cycle time), assume no stalls and that you run 8 instructions. Compare this to a single-cycle (non-pipelined) design whose cycle time is the sum of all five stages ($28 + 59 + 23 + 34 + 36 = 180$ ns) for each instruction. Which of the following best describes the approximate speedup from pipelining under these conditions?

A. About $1.5\times$
B. About $2\times$
C. About $3\times$
D. About $5\times$

Correct Answer: B (about $2\times$).

33. Define a "Kryptonian" 32-bit binary number (two's complement format) to be one that has strictly more 0 bits than 1 bits when ignoring any leading zeros to the left. For instance, 1001002100100_21001002 (which is 36 in decimal) is Kryptonian because, once you strip

leading zeros, it has four bits total (1001), with two zeroes and two ones—actually that example is borderline. Let's consider a different example for clarity.

Which of the following decimal numbers is definitely “Kryptonian” in 32-bit two's complement?

- A. +18 (decimal)
- B. -18 (decimal)
- C. +8 (decimal)
- D. -1 (decimal)

Correct Answer: A (+18).

34. Define a “Xenonian” 32-bit binary number to be one that has exactly 4 bits set to 1 (i.e., 4 ones) in its representation. Which of the following decimal values is Xenonian?

- A. 30 (11110₂)
- B. 15 (0000...01111₂)
- C. 27 (11011₂)
- D. 240 (11110000₂)

Correct Answer: D (240).

35. A cache uses error-correcting codes (ECC) for each cache block. On a read or write, if the ECC bits indicate data corruption in that block, the cache treats it as a miss and fetches from memory again.

If memory itself is assumed incorruptible, which statement best describes why these corrupted-block accesses behave like cache misses?

- A. The block's “valid” bit is turned off, forcing a re-fetch.
- B. The ECC bits cause an immediate flush of the entire cache.
- C. Any ECC error means the data in the cache line is invalid, so the cache must fetch a correct copy from memory.
- D. The CPU reboots after any ECC error.

Correct Answer: C.

36. Suppose this same cache is used in a harsh radiation environment and is 2-way set-associative with ECC. The design choice is to ensure correctness even if some bits flip in the cache. Which write policy (for writes to an already-cached block) is more appropriate to avoid risking corrupted data eventually overwriting main memory?

- A. Write-through
- B. Write-back with write-allocate
- C. Write-back without allocation
- D. No writes at all

Correct Answer: A (write-through).

37. A cache with 4-byte blocks uses these 4 bytes as the fundamental unit of storage in each line. How many offset (a.k.a. block offset) bits does each address need to select a byte within that block?
- A. 1
 - B. 2
 - C. 3
 - D. 4

Correct Answer: B (2 bits).

38. Assume a system has 128 KB of virtual address space and 32 KB of physical memory. It uses single-level paging, with each page containing 1 KB. Each page table entry (PTE) takes 2 bytes. Approximately how many total pages are in the virtual address space, and how large is the page table?
- A. 128 pages total; 256-byte page table
 - B. 128 pages total; 128-byte page table
 - C. 128 pages total; 2-byte page table
 - D. 1024 pages total; 2 KB page table

Correct Answer: D.

39. In a single-level paging system, a TLB miss occurs but no page fault is needed (the page is present in memory). How many memory accesses must typically happen to service that single load?
- A. 1 memory access total (the load itself)
 - B. 2 memory accesses: one to read the page-table entry, then another for the actual data
 - C. 0 memory accesses, because the TLB always handles addresses internally
 - D. 3 memory accesses: one for the page directory, one for the page table, one for data

Correct Answer: B (2).

40. A single-level page table for a process can occupy multiple pages in memory. If the virtual address space is large and the page size is relatively small, the page table can become quite large. Which statement is correct about how the page table is typically stored?
- A. It is always stored entirely in the CPU registers.
 - B. It is stored as one big array in kernel-owned physical memory, and a page table base register (PTBR) points to its start.
 - C. It is stored across multiple TLB entries, so no extra memory is needed.
 - D. It is stored in a special I/O device controlled by the operating system.

Correct Answer: B.

41. Convert 683 (decimal) to hexadecimal.
- A. 0x2AB
 - B. 0x2BB

- C. 0x2ABF
- D. 0x56B

Correct Answer: A (0x2AB).

42. The 8-bit binary number 10000111_2 should be represented in base 6. Which of the following is correct?
- A. 203_6
 - B. 427_6
 - C. 507_6
 - D. 1003_6

Correct Answer: B (427_6).

43. Which of the following decimal values corresponds to the binary fraction 1101.101_2 ?
- A. 13.625
 - B. 13.3125
 - C. 27.625
 - D. 27.3125

Correct Answer: A (13.625).

44. Consider an x86-like CPU that sets the Carry Flag (CF), Zero Flag (ZF), Sign Flag (SF), and Overflow Flag (OF) after an integer addition. Is it possible (in a single addition of two registers) to set CF=1, ZF=0, SF=1, and OF=1 all at once?
- A. No, because setting SF=1 always implies CF=0.
 - B. No, because when OF=1 and CF=1 are set, ZF must be 1.
 - C. Yes, it is possible, for instance adding 0xC0000000 and 0xC0000000 in 32-bit leads to SF=1, CF=1, OF=1, ZF=0.
 - D. Yes, but only if the operands are both zero.

Correct Answer: C.

45. Which of the following 32-bit binary patterns represents a NaN (Not a Number) in IEEE 754 single-precision?
- A. 0111 1111 1100 1010 0100 1001 0001 0010
 - B. 1111 1111 0100 1010 0100 1001 0001 0010
 - C. 0000 0000 0000 0000 0000 0000 0000 0000
 - D. 0111 1111 1000 0000 0000 0000 0000 0000

Correct Answer: B.

46. Assume an IEEE-style 12-bit format with:

1 sign bit

E exponent bits

F fraction bits

A certain bias

The format can exactly represent some values but not others.

If this format uses 6 fraction bits and the exponent is $(12 - 1 - 6) = 5$ bits, what is the bias for the exponent if it follows the usual IEEE rule “bias = $2^{(E-1)} - 1$ ”?

- A. 15
- B. 31
- C. 63
- D. 7

Correct Answer: B (31).

47. A two-input NOR gate is called “functionally complete.” Which statement best explains why?

- A. You can build an XOR from two NOR gates, but you cannot build an AND from NOR.
- B. NOR gates can be used to build NOT, AND, and OR gates, hence any digital logic.
- C. A NOR gate is the same as an OR gate followed by an inverter, so it only builds OR operations.
- D. A NOR gate can only generate zero, so you cannot build all gates.

Correct Answer: B (NOR can generate all basic logic functions, so it is “functionally complete.”)

48. Consider this snippet of assembly on a typical 64-bit architecture:

```
mov(%rdi), %eax ; load a 32-bit value from address in %rdi
```

```
mov(%rsi), %edx ;
```

```
load a 32-bit value from address in %rsi mov %edx, (%rdi) ;
```

```
store that value back to memory at %rdi mov %eax, (%rsi) ;
```

```
store the old %eax value to memory at %rsi add(%rdi), %eax ;
```

```
add whatever is now in memory at %rdi to %eax ret
```

If initially:

- memory at 0x48C holds 9,
- memory at 0x484 holds 2,

and `%rdi = 0x48C`, `%rsi = 0x484` before the call, what value does `%eax` (the function’s return) hold after the snippet executes?

- A. 2
- B. 9
- C. 11
- D. 18

Correct Answer: C (11).

```
49. typedef struct student {  
    short year;  
  
    char major[4];  
  
    int *id;  
  
    struct {  
        char country[3];  
  
        int areacode;  
    } home;  
  
    struct student *nextstudent;  
}  
student;
```

Which of the following lines of code corresponds to reading the second character of `s->home.country`?

- A. `char x = (s->home).country[1];`
- B. `int x = *(s->nextstudent->id);`
- C. `int* x = &((s->home).areacode);`
- D. `int x = (s->year);`

Correct Answer: A.

50. Suppose we have a function in assembly that does the following in a loop:
1. Initialize a counter to 5.
 2. Repeatedly square that counter and add to a running sum.
 3. Decrement the counter.
 4. Continue while the counter > 0.
 5. Return the final sum.

If this loop squares each integer from 5 down to 1 and sums them, the final return value is:

- A. 55 ($1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$)
- B. 15
- C. 225
- D. 5

Correct Answer: A (55).

51. In that same ISA, a jump instruction uses a 4-bit mask for (C0, C1, C2, C3). If we want to jump whenever the result is less than or equal to zero, we set the mask bits for (C0, C1) so that if either “zero” (C0=1) or “negative” (C1=1) is true, we jump. That mask in binary is:

- A. 0011
- B. 1100
- C. 1010
- D. 0110

Correct Answer: A (0011).

52. Convert 683 (decimal) to hexadecimal.

- A. 0x2AB
- B. 0x5AB
- C. 0x2B8
- D. 0xAA3

Correct Answer: 683 \rightarrow 0x2AB.

53. The 8-bit binary number 10000111_2 is equal to 135 in decimal. Which of the following base-6 representations is 135 in base 6?

- A. 351_6
- B. 1003_6
- C. 343_6
- D. 427_6

Correct Answer: C (343_6)

54. The binary fraction 1101.101_2 equals which decimal value?

- A. 13.625
- B. 13.375
- C. 27.625
- D. 27.3125

Correct Answer: A (13.625)

55. Is it possible (in one register-to-register addition) for the CPU to set CF=1, ZF=0, SF=1, OF=1 all at once?

- A. Yes. Adding two large negative numbers can set all four.
- B. Yes. Adding $0xC0000000 + 0xC0000000$ does it.
- C. No. Having SF=1 and OF=1 implies adding two positives and getting a negative result, which cannot set CF=1.
- D. Yes. Provided the sum is -1 in two's complement.

Correct Answer: C (No, it is not possible)

56. In the exam, the number $7 \times 64^{\{19/\dots\}}$ was converted to a binary normalized form, resulting in 1.11010011×2^{21} . Which step is crucial for such a conversion?

- A. Shift the binary digits until exactly one "1" is left of the decimal point, and adjust the exponent.
- B. Multiply by 2 repeatedly until you get a fraction < 1 .

- C. Look up a table for powers of 64.
- D. You cannot convert 64 into binary directly.

Correct Answer: A (You shift until there is one leading 1, then adjust the exponent.)

57. Which one is recognized as NaN in IEEE-754 single-precision?

- A. 0111 1111 1100 1010 0100 1001 0001 0010
- B. 1111 1111 0100 1010 0100 1001 0001 0010
- C. 0000 0000 0000 0000 0000 0000 0000 0000
- D. 0111 1111 1000 0000 0000 0000 0000 0000

Correct Answer: A

58. In a hypothetical 12-bit floating-point format consistent with IEEE standards, the exam states:

- The fraction uses 6 bits.
- The bias is 15.
- The pattern 010000111001 is exactly $332253\frac{32}{25}32532$.

Which statement best describes the exponent in that format?

- A. 5 bits for the exponent, giving a bias of 31
- B. 5 bits for the exponent, giving a bias of 15
- C. 6 bits for the exponent, giving a bias of 63
- D. 4 bits for the exponent, giving a bias of 7

Correct Answer: B (5 exponent bits, bias = 15)

59. Consider a function that initializes a sum to 0, a counter to 5, then in a loop squares the counter and accumulates it, decrementing the counter each time, stopping when the counter is 0 or less. The final return is 30. Which sum is the code actually computing?

- A. $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$
- B. $2^2 + 3^2 + 4^2 + 5^2 = 54$
- C. $1^2 + 2^2 + 3^2 + 4^2 = 30$
- D. $5 + 4 + 3 + 2 + 1 = 15$

Correct Answer: C ($1^2 + 2^2 + 3^2 + 4^2 = 30$)

60. A snippet moves the 32-bit value from memory at `%rdi` into `%eax`, from `%rsi` into `%edx`, swaps them in memory, then adds `(%rdi)` to `%eax` and returns. If initially:

- `*(int*)0x48C = 9`
- `*(int*)0x484 = 2`
- `%rdi = 0x48C`

- `%rsi = 0x484`

the final memory at `0x484` = 9, `0x48C` = 2, and the return value is `0xB` (decimal 11). However, the function returned `0xB`.

Which final scenario is correct?

- A. Memory swapped to `[0x484]=9`, `[0x48C]=2`, sum in `%eax=11`
- B. Memory unchanged, sum in `%eax=0`
- C. Memory swapped to `[0x484]=2`, `[0x48C]=9`, sum in `%eax=11`
- D. Memory swapped to `[0x484]=9`, `[0x48C]=2`, sum in `%eax=0xB`, but that is 0

Correct Answer: A.

61. You have the binary number 11_2 (which is decimal 3) and add it to `0x40` (decimal 64). Express the result in octal (base 8). Which is correct?

- A. `0o77`
- B. `0o103`
- C. `0o100`
- D. `0o143`

Correct Answer: B (`0o103`)

62. Suppose you have a snippet:

```
add    rcx, rdx    ; combines rcx=5, rdx=2
mov     3, r11
.L1:   dec rcx
mov     r8, r9
cmp     0, 1
beq     0x584fc0
inc     r11
.L2:   cmp    rcx, 0
jne     .L1
.L3:   add     r8, r9
```

Question: When `.L3` is reached, what final value is stored in `r11`?

- A. 3
- B. 8
- C. 10
- D. 12

Correct Answer: C (10)

62. In the same code snippet (above), you see `.L2: cmp rcx, 0`. Which other instruction could replace `cmp rcx, 0` without changing the program's behavior?

- A. `test rax, rax`
- B. `test rcx, rcx`
- C. `sub rcx, 1`
- D. `or rdx, rdx`

Correct Answer: B (`test rcx, rcx`)

63. Which code correctly multiplies an integer `num` by 5 using no loops, no multiplication operator, and at most three operators?

- A. `return (num << 2) + num;`
- B. `return (num << 3) - num;`
- C. `return (num << 2) + (num << 1);`
- D. `return (num << 1) + 3;`

Correct Answer: A (Shift left by 2 = $\times 4$, plus original = $\times 5$.)

64. Which of the following is not a valid reason that a pipeline might stall?

- A. Data dependency
- B. Cache miss
- C. Control dependency (branch)
- D. The CPU doesn't "like" the instruction mnemonic

Correct Answer: D

65. Which binary normalized form is correct for $4\frac{3}{8}$?

- A. 1.10011×2^3
- B. 1.00011×2^2
- C. 0.100011×2^4
- D. 1.11100×2^1

Correct Answer: B (1.00011×2^2)

66. Suppose we have 8-bit floating-point numbers $A=11111011_2$, $B=00100100_2$, $C=00110001_2$, $D=10110001_2$. The:

- The sum of $B + C = 00110100_2$
- 4 exponent bits, 3 fraction bits

- Bias = 7

If $B + C = 00110100_2$, which pair is correct?

- A. $A + B = 00110100_2$
- B. $B + C = 00110100_2$
- C. $C + D = 00110100_2$
- D. $A + D = 00110100_2$

Correct Answer: B

67. We have a 4-way associative cache with 4-byte lines, 512 total lines, and a special replacement policy that never replaces the most recently used line. Overhead bits per set are 26. Indices 0..3 are assigned to lines in each set, and we store the “most recently used” index.

How many bits per set are needed to store the “index of most recently used line”?

- A. 1 bit
- B. 2 bits
- C. 3 bits
- D. 26 bits

Correct Answer: B (2 bits)

68. Continuing from above, we have a physical memory size of 16 KB, and the cache has 512 lines of 4 bytes each (so $512 \times 4 = 2048$ bytes in the cache). That means addresses are 14 bits total (because $16 \text{ KB} = 2^{14}$). We also have 9 bits for offset+index (2 bits for offset, 7 bits for index, since 512 lines is 128 sets?). We have 5 tag bits.

Which is correct for the number of tag bits?

- A. 9 bits
- B. 5 bits
- C. 7 bits
- D. 14 bits

Correct Answer: B (5 bits)

69. A system has:

- Virtual address space = 64 KB
- Physical memory = 16 KB
- Page size = 256 bytes

Hence there are 256 virtual pages and 64 physical pages. Each page table entry (PTE) stores:

- 1 valid bit
- 1 dirty bit
- Physical page number (6 bits to select among 64 pages)

How large is each PTE, in bits?

- A. 8 bits total
- B. 6 bits total
- C. 16 bits total
- D. 4 bits total

Correct Answer: A (8 bits)

70. Consider:

```
void square(int a[16][16]) {  
    for (int j = 0; j < 16; j++) {  
        for (int i = 0; i < 16; i++) {  
            a[i][j] = a[i][j] * a[i][j];  
        }  
    }  
}
```

If each array row is in a different page, and we have a 2-entry data TLB initially empty, the code iterates columns inside the inner loop, causing many TLB misses.

How many TLB misses does this code incur, and how can we reduce them?

- A. 256 misses total, reorder loops → 16 misses
- B. 64 misses total, reorder loops → 4 misses
- C. 64 misses total, reorder loops → 16 misses
- D. 32 misses total, reorder loops → 4 misses

Correct Answer: B (64 misses as-is, and reordering i/j leads to only 4 misses)

71. Which of the following is the correct 7-bit binary representation of 92 (decimal)?

- A. 0101110
- B. 1011100
- C. 1101010
- D. 0111101

Correct Answer: B (1011100)

72. The base-5 number “243₅” is equal to which decimal value?

- A. 53
- B. 73
- C. 81
- D. 103

Correct Answer: B (73)

73. In an 8-bit two's-complement representation, which hexadecimal value corresponds to -92 ?

- A. 0xA4
- B. 0xBC
- C. 0x5C
- D. 0xD4

Correct Answer: A (0xA4)

74. We have 4-bit registers $R1=1100_2$ (decimal 12) and $R2=0111_2$ (decimal 7). We do **add R1, R2** $\rightarrow 12 + 7 = 19$ decimal $= 10011_2$ in 5 bits, but only the low 4 bits are stored, etc.

Which combination of Carry (CF), Overflow (OF), Sign (SF) is set?

- A. CF=0, OF=1, SF=1
- B. CF=1, OF=0, SF=0
- C. CF=1, OF=1, SF=1
- D. CF=0, OF=0, SF=0

Correct Answer: B (Carry=1, Overflow=0, Sign=0)

75. $19\frac{3}{8}$ is 19.375 in decimal. Which normalized binary form is correct?

- A. 1.0011011×2^3
- B. 1.0011011×2^4
- C. 1.0011011×2^2
- D. 1.0011011×2^1

Correct Answer: B (1.0011011×2^4)

76. According to the exam, the format uses:

- 14 total bits
- 5 exponent bits
- 8 fraction bits

How many bits are left for the exponent and fraction?

- A. Exponent=4, Fraction=9
- B. Exponent=5, Fraction=8
- C. Exponent=6, Fraction=7
- D. Exponent=7, Fraction=6

Correct Answer: B (Exponent=5 bits, Fraction=8 bits)

77. What is the result of XOR between 0101_2 (decimal 5) and 1001_2 (decimal 9)?

- A. 1100_2
- B. 0000_2

- C. 1101_2
- D. 0100_2

Correct Answer: A (1100_2)

78. Which is the bitwise NOR result for those same two 4-bit values?

- A. 1111_2
- B. 0010_2
- C. 1100_2
- D. 0000_2

Correct Answer: B (0010_2)

79. We have a 2-to-1 multiplexer with inputs In0, In1, and select S.

To build NOT(A) using one MUX, we set:

- In0 = 1
- In1 = 0
- S = A

Which is the output?

- A. A
- B. 1
- C. $\sim A$
- D. 0

Correct Answer: C ($\sim A$)

80. We have a 3-input circuit ($A, B, C \rightarrow \text{Out}$) with two identical gates "X" and the truth table demands the gates be OR. If each gate is 1ps and we have three gates in series, the total delay is 3ps.

Which gate must "X" be so that the final output matches the given table?

- A. AND
- B. OR
- C. XOR
- D. NOR

Correct Answer: B (OR)

81. In a struct with these fields:

```
char  netid[12];  
  
char* name;  
  
int   sid;
```

```
double gpa;
```

We want to reorder for minimal padding. .

Question: Why does placing `sid` before `name` and `gpa` reduce padding?

- A. Because `sid` is 4 bytes and `gpa` is 8 bytes, we avoid gaps by aligning them in ascending order of alignment.
- B. Because the compiler discards `sid` if it's at the end.
- C. Because a `char*` is smaller than an `int`.
- D. Because a `char[]` is always 8 bytes aligned.

81. We have to pipelines. The 5-stage pipeline runs at 1 GHz (1 ns per cycle). The 3-stage pipeline runs at 600 MHz (1.666... ns per cycle). The 5-stage pipeline finishes in 24 cycles = 24 ns, while the 3-stage pipeline finishes in 20 cycles = ~33 ns.

Question: Which pipeline completes the code faster?

- A. 5-stage CPU
- B. 3-stage CPU
- C. They tie because $20 \text{ cycles} \times 1.666... \text{ ns} \approx 24 \text{ cycles} \times 1 \text{ ns}$
- D. Not enough information

Correct Answer: A (The 5-stage is faster in wall-clock time: 24 ns vs 33 ns.)

82. In a 32-bit IEEE 754 float, the fraction has 23 bits. A “quiet NaN” sets the top fraction bit=1, plus an additional bit for metadata. We can store up to 21 bits of “pointer” inside that fraction field.

Question: If 2 fraction bits are used as “metadata,” how many fraction bits are left for storing a pointer?

- A. 21
- B. 23
- C. 25
- D. 16

Correct Answer: A (21 bits)

83. The decimal number 334 is equal to which of the following hexadecimal values?

- A. 0x14E
- B. 0x134
- C. 0x15E
- D. 0x1AE

Correct Answer: A (0x14E)

84. On a single-CPU system, a user program enters an infinite loop, but the whole computer remains responsive.

Question: Which statement best explains *why* the machine still doesn't freeze?

- A. Modern CPUs detect infinite loops automatically and skip them.
- B. The operating system preempts the process to perform a context switch periodically.
- C. The BIOS forcibly terminates any looping process after one second.
- D. The CPU uses additional hardware threads to handle other tasks.

Correct Answer: B.

85. We have a shell with a foreground job (myprogA) and two background jobs: myprogB (running) and myprogC (stopped). When the user presses Ctrl-C:

Question: Which signal(s) are typically sent, and to which recipient(s)?

- A. **SIGINT** goes to the shell alone.
- B. **SIGINT** goes to **myprogA**, then **SIGCHLD** goes back to the shell when **myprogA** terminates.
- C. **SIGINT** goes to all background jobs, then **SIGCHLD** is ignored.
- D. **SIGSTOP** is sent to **myprogA**; no **SIGCHLD** is generated.

Correct Answer: B.

86. Suppose we have the decimal number 6.3125 (which is $6 \frac{5}{16}$). We want its binary floating representation in normalized form.

Question: Which is correct?

- A. 1.100101×2^2
- B. 1.100101×2^1
- C. 1.100101×2^3
- D. 1.100101×2^4

Correct Answer: A (1.100101×2^2)

87. You have a custom floating-point standard with:

- A certain sign bit
- 4 exponent bits
- 6 fraction bits
- The bias = 7.

Question: Which statement best describes the smallest positive *normalized* value in this format?

- A. $2^{(-12)}$
- B. $2^{(-7)}$
- C. $2^{(-6)}$
- D. $2^{(-23)}$

Correct Answer: A ($2^{(-12)}$)

88. We have a function **find_next(x)** which:

- If $x \leq 0$, returns $x+1$.
- If $x=2$, returns 3.
- For other positive x , returns the smallest prime $> x$.

Question: If we call `find_next(2)`, what is the return value?

- A. 2
- B. 3
- C. 4
- D. 5

Correct Answer: B (3)

89. In standard x86-64 AT&T syntax, `call label` does two key things:

1. It pushes the return address onto the stack,
2. It jumps to `label`.

Question: If `call` is “faulty,” which two instructions can replicate its effect (in that order)?

- A. `jmp` then `push`
- B. `push` then `jmp`
- C. `pushf` then `jmp`
- D. `cmp` then `jmp`

Correct Answer: B (Push the return address, then jump.)

90. We have an 8-bit machine (so addresses are 8 bits). The line size is 2 bytes. The cache is 4-way set associative with 2 sets.:

- Offset bits = 1 (because each line holds 2 bytes)
- Index bits = 1 (2 sets)
- Tag bits = 6 (the remaining bits)

Question: How many bits does each address use for the cache’s tag field?

- A. 5 bits
- B. 6 bits
- C. 7 bits
- D. 4 bits

Correct Answer: B (6 bits)

91. We have 8 total lines (4-way \times 2 sets). Each line has:

- 1 valid bit
- 1 dirty bit
- 6 tag bits

Question: How many total overhead bits are in the entire cache?

- A. 56 bits
- B. 64 bits
- C. 48 bits
- D. 72 bits

Correct Answer: B (64 bits)

92. The program references addresses in a fixed pattern, and we have a write-back 4-way/2-set cache. Suppose the first run has 8 misses out of 9 accesses. The second run repeats the same address sequence.

Question: How many cache misses occur on this second run?

- A. 0
- B. 1
- C. 4
- D. 8

Correct Answer: A (0)

93. We have:

- Virtual address space = 256 KB ($= 2^{18}$ bytes)
- Physical memory = 32 KB
- Single-level page table
- Page size = 1 KB

If the page size is 1 KB (2^{10} bytes), how many bits are used for the virtual page number (VPN) and how many for the physical page number (PPN)?

- A. VPN=8, PPN=5
- B. VPN=10, PPN=8
- C. VPN=12, PPN=6
- D. VPN=8, PPN=3

Correct Answer: A (VPN=8, PPN=5)

94. Under the same setup, the each PTE includes:

- A physical page number (5 bits)
- A “disk address” field 2 bits larger than the PPN
- A valid bit

Question: How many bits is each PTE?

- A. 5 bits
- B. 6 bits
- C. 8 bits
- D. 10 bits

Correct Answer: C (8 bits)

95. We have a function that accesses `arr[i]` or `arr[(i << 8)+1]`, etc. The first iteration can cause 2 page faults. The maximum number of page faults is 9, and the minimum is 2.

Question: Which scenario maximizes the page faults?

- A. The code always accesses the same page each time
- B. The function chooses array elements in a pattern that triggers a new page for every iteration
- C. The function never touches the array at all
- D. The code randomly flushes the TLB each iteration

Correct Answer: B

96. If we switch from page size 1 KB to 4 KB, we reduce the number of pages. In many typical workloads, the TLB coverage improves.

Usually, when page size is increased from 1 KB to 4 KB, the TLB hit rate tends to:

- A. Decrease, because the TLB can hold fewer entries
- B. Stay the same
- C. Increase, because each TLB entry covers more addresses
- D. Make no difference at all

Correct Answer: C

97. We only have 2 bits of permission data per page, and we want to prevent user-writable memory from also being executable.

Question: Which design effectively uses those 2 bits?

- A. One bit for read permission, one bit for “write or execute” but never both.
- B. One bit for kernel mode, one bit for user mode.
- C. Two bits that track the program’s next instruction fetch address.
- D. One bit for “owner ID,” one bit for “public access.”

Correct Answer: A.

98. Which of the following hexadecimal values represents decimal 42?

- A. 0x2E
- B. 0x2A
- C. 0x32
- D. 0x3A

Correct Answer: B

99. The octal (base-8) number 35_8 corresponds to which pair of decimal/binary values?

- A. 29 decimal, 11101_2
- B. 27 decimal, 11011_2
- C. 25 decimal, 11001_2
- D. 35 decimal, 100011_2

Correct Answer: A

100. In a 6-bit two's-complement representation, which pair of 6-bit patterns correctly represents -25 and -18?

- A. -25: 111001, -18: 101110
- B. -25: 100111, -18: 101110
- C. -25: 110011, -18: 101010
- D. -25: 100100, -18: 100111

Correct Answer: B

101. If $R1 = 100111_2$ (-25) and $R2 = 101110_2$ (-18) in two's complement 6-bit registers, we do **add R1, R2** ($R2 \leftarrow R2 + R1$).

Which flags are set afterward?

- A. Zero=1, Sign=1, Overflow=0
- B. Zero=0, Sign=1, Overflow=0
- C. Zero=0, Sign=0, Overflow=1
- D. Zero=1, Sign=0, Overflow=1

Correct Answer: C (Zero=0, Sign=0, Overflow=1)

102. The decimal -35.75 equals which normalized binary representation?

- A. -1.0001111×2^5
- B. -1.1110001×2^3
- C. -1.0001111×2^3
- D. -1.1110001×2^5

Correct Answer: A (-1.0001111×2^5)

103. If 4 bits are for exponent and 8 bits are for fraction, which is the correct exponent/fraction partition?

- A. Exponent=4 bits, Fraction=8 bits
- B. Exponent=5 bits, Fraction=7 bits
- C. Exponent=3 bits, Fraction=9 bits
- D. Exponent=7 bits, Fraction=4 bits

Correct Answer: A (4 exponent, 8 fraction)

104. Suppose $f1 < f2$ as real (floating) numbers. We store $f1$ and $f2$ in IEEE single-precision format and interpret those bit patterns as unsigned integers $i1$ and $i2$.

Question: Can we guarantee $i1 < i2$ always?

- A. Yes, because a smaller float always has a smaller bit pattern in IEEE format.
- B. No, for example a negative $f1 < 0 < f2$ can produce a "larger" unsigned pattern for $f1$.
- C. Yes, only if both are positive. Otherwise, we can't interpret them as unsigned.
- D. No, because the exponent field scrambles the order.

Correct Answer: B

105. Which is the bitwise NAND of 101101_2 and 010110_2 , and the bitwise XOR of 101010_2 and 111011_2 , respectively?

- A. (NAND) 111011 and (XOR) 010001
- B. (NAND) 000100 and (XOR) 101001
- C. (NAND) 111101 and (XOR) 000100
- D. (NAND) 110010 and (XOR) 001101

Correct Answer: A

106. We have a chain of N XOR gates, each taking the output of the previous one with input A (for each stage).:

- For N=50, output=1
- For N=63, output=NOT(A)

Question: Which pattern holds for the final XOR output?

- A. If N is even, output=NOT(A); if N is odd, output=1
- B. If N is odd, output=NOT(A); if N is even, output=1
- C. Output always equals A, for any N
- D. Output always equals 1, for any N

Correct Answer: B

107. We have $A, B, C \rightarrow \text{Output}$, with a truth table that sets output=1 only if exactly two inputs are 0 and 1 is 1 in a certain pattern. The official key says each unknown gate X is XOR, giving final logic that matches the table.

Question: Which gate is used for X?

- A. OR
- B. AND
- C. XOR
- D. NOR

Correct Answer: C (XOR)

108. The C struct is:

```
struct Data {  
    int matrix[3][3];  
    int *value;  
};
```

What is the total size of `struct Data`?

- A. 36 bytes
- B. 40 bytes
- C. 48 bytes
- D. 56 bytes

Correct Answer: C (48 bytes)

109. A function `is_equal(&d1, &d2)` compares all bytes (including padding). If every byte is the same, returns 1, else 0.

If two `struct Data` objects differ only in uninitialized padding bytes, can `is_equal` always return 1?

- A. Yes, C standard guarantees padding is zero.
- B. No, the padding could be anything, so it might differ.
- C. Yes, as long as the pointers are both `NULL`.
- D. No, if `value` fields differ, it's always 0.

Correct Answer: B

110. We have an array `[9,8,7,6]`, initial `%esi=4`, and the function sums certain elements until it hits a condition.

Question: Which final value does the function return?

- A. 17
- B. 24
- C. 30
- D. The function never terminates

Correct Answer: C (30)

111. We compare two processors:

1. 5-stage pipeline at 2.5 GHz (400 ps cycle).
2. 3-stage pipeline at 1.6 GHz (625 ps cycle).

Assume no stalls. Which throughput in instructions per second is correct?

- A. 2.5 billion vs. 1.6 billion
- B. 2.5 million vs. 1.6 million
- C. 4.0 billion vs. 3.2 billion
- D. 1.6 billion vs. 2.5 billion

Correct Answer: A

112. We run a snippet with `incq %rax; cmpq %rax, %rbx; jge .L8; ... jmp .L9; nop` Only one jump is actually taken.

Question: Which jump label is taken?

- A. .L3
- B. .L7
- C. .L8
- D. .L13

Correct Answer: A

113. In a 5-stage pipeline (Fetch, Decode, Execute, Memory, Writeback), at which pipeline stage is a branch or jump's direction known, i.e., when do we resolve the control dependency?

- A. At the end of Fetch stage
- B. At the end of Decode stage
- C. At the end of Execute stage
- D. At the end of Memory stage

Correct Answer: C (Execute stage)

114. What is the binary representation of the sum of 328 (decimal) and 0x32 (hex)?

- A. 01001100₂
- B. 10011100₂
- C. 00101110₂
- D. 01100010₂

Correct Answer: A (01001100₂)

115. For two caches of the same total size, one is direct-mapped and the other is fully associative. Which inequality best describes the typical access time relationship?

- A. Direct-mapped > Fully associative
- B. Direct-mapped = Fully associative
- C. Direct-mapped <= Fully associative
- D. Direct-mapped >= Fully associative

Correct Answer: C

116. True or False: Virtual memory has no size limitation.

- A. True
- B. False

Correct Answer: B

117. Suppose we have:

```
struct Person {  
    long id;
```

```

float age;

float weight;

float height;

char name[20];

char sex[7];

struct Person *nextPerson;

};

```

On a 64-bit system, which is the total size of `struct Person`?

- A. 48 bytes
- B. 52 bytes
- C. 56 bytes
- D. 64 bytes

Correct Answer: C (56 bytes)

118. We have an 8-bit instruction format with an OpCode in bits 7–5 and either a 5-bit immediate (signed) or 5 “don’t care” bits. The instruction “store -8” has OpCode=100 and immediate=-8 in 5-bit two’s complement.

Question: Which 8-bit binary correctly encodes “store -8”?

- A. 10011000₂
- B. 10001000₂
- C. 10010111₂
- D. 10101000₂

Correct Answer: A (10011000₂)

119. We have an n-bit IEEE-like float that:

- Precisely represents $6\frac{1}{3}$
- Fails to represent $11\frac{7}{16}$ exactly
- Has a smallest positive normalized value = 2^{-30}
- Fraction=6 bits, exponent=6 bits, bias=31, total n=13.

Question: Which is the total bit-length “n” of this floating-point format?

- A. 12
- B. 13
- C. 14
- D. 16

Correct Answer: B (13)

120. We do either:

1. $(256 + 2^{1/4}) - 256$
2. $(256 - 256) + 2^{1/4}$

Which outcome best describes these two results in that n-bit float?

- A. Both evaluate to $2^{1/4}$ precisely.
- B. Both evaluate to 0 due to rounding.
- C. The first yields 0 (rounded away), the second yields $2^{1/4}$.
- D. The first yields $2^{1/4}$, the second yields 0.

Correct Answer: C

121. We have:

- Tag bits = 5
- Offset bits = 3
- So each cache line is $2^3 = 8$ bytes

Question: If the machine address is 12 bits total, offset=3, index=4, how many bits remain for the tag?

- A. 3 bits
- B. 5 bits
- C. 4 bits
- D. 7 bits

Correct Answer: B (5 bits)

122. We have a direct-mapped cache of 16 lines (each 8 bytes). We have:

- #1: $110111110000_2 \rightarrow \text{Miss}$
- #2: $000011011111_2 \rightarrow \text{Miss}$
- #3: $110101110101_2 \rightarrow \text{Miss}$
- #4: $000011011100_2 \rightarrow \text{Hit}$
- #5: $110111110011_2 \rightarrow \text{Miss}$
- #6: $111101110010_2 \rightarrow \text{Miss}$
- #7: $110111110000_2 \rightarrow \text{Miss}$
- #8: $000011011101_2 \rightarrow \text{Hit}$
- #9: $111101110100_2 \rightarrow \text{Miss}$

How many Misses total?

- A. 5
- B. 6
- C. 7
- D. 8

Correct Answer: D (8 misses)

123. Given:

- Virtual address space=32 KB
- Physical memory=8 KB
- Page size=128 B
- 1-level page table with each PTE has { valid bit, dirty bit, PPN }

Question: The total number of virtual pages is:

- A. 64
- B. 128
- C. 256
- D. 512

Correct Answer: C ($256 \rightarrow 32 \text{ KB} / 128 \text{ B} = 256$)

124. We have 256 virtual pages, each PTE includes 1 valid bit, 1 dirty bit, and 6 bits for PPN, total 8 bits per entry. The $256 \text{ entries} \times 8 \text{ bits} = 256 \text{ bytes total}$.

Question: How many bytes is the entire page table?

- A. 128 bytes
- B. 256 bytes
- C. 512 bytes
- D. 64 bytes

Correct Answer: B (256 bytes)

125. We have a TLB that holds only the last used page table entry. Reading **a[1]** might cause TLB misses. There are possibly 2 or 4 TLB misses depending on assumptions about register usage.

Question: Under the “1 register only” assumption, how many TLB misses occur?

- A. 2
- B. 3
- C. 4
- D. 6

Correct Answer: C (4)

126. Which hexadecimal number corresponds to decimal 24?

- A. 0x16
- B. 0x18
- C. 0x1A
- D. 0x24

Correct Answer: B (0x18)

127. The octal (base 8) number 53_8 is which decimal/binary pair?

- A. 43 decimal, 101011_2
- B. 43 decimal, 110101_2
- C. 35 decimal, 111001_2
- D. 35 decimal, 100011_2

Correct Answer: A

128. In binary, what is $A + B$ ($1011_2 + 0101_2$) with enough bits to be precise?

- A. 1111_2
- B. 10000_2
- C. 01110_2
- D. 1000_2

Correct Answer: B (10000_2)

129. In binary, what is $A - B$ ($1011_2 - 0101_2$) with enough bits to be precise?

- A. 0110_2
- B. 1100_2
- C. 1000_2
- D. 1010_2

Correct Answer: A (0110_2)

130. Given $F = -100.50$ decimal.

Which is F 's binary fraction and normalized form?

- A. -1.1001001×2^6 with fraction 1001001
- B. -1.001001×2^5 with fraction 001001
- C. -1.100101×2^6 with fraction 100101
- D. -1.111001×2^7 with fraction 111001

Correct Answer: A

131. The 32-bit float pattern 0x7F800000 corresponds to which value?

- A. NaN
- B. Positive infinity
- C. Negative infinity
- D. 0

Correct Answer: B

132. We design a custom float representation representing all possible scores from 64 multiple-choice questions.:

- 7 bits for the fraction (mantissa)
- 4 bits for the exponent
- Bias = 7
- This suffices to represent all valid fractional increments of 0.25 up to 64

If we used a fixed-point representation instead, how many bits minimum would be required?

- A. 8 bits
- B. 10 bits
- C. 12 bits
- D. 16 bits

Correct Answer: B (10 bits)

133. We have $A_3A_2A_1A_0$, $B_3B_2B_1B_0$, and an input X controlling XOR gates. The full adders produce $\Sigma_3\Sigma_2\Sigma_1\Sigma_0$ with various carry logic.

If $A=1001_2$, $B=0001_2$, and $X=0$

The XOR outputs $P_3P_2P_1P_0 = ?$

- A. 1110
- B. 0001
- C. 1000
- D. 1111

Correct Answer: B

134. A single full adder with certain gates takes 3ps. If we chain 4 of these adders plus additional XOR gates (1ps each) in a path, the total circuit delay is 9ps.

Typically, how many picoseconds does each full adder stage add to the critical path?

- A. 1ps
- B. 2ps
- C. 3ps
- D. 4ps

Correct Answer: C (3ps)

135. If we have:

```
struct Car {  
    char *model;  
    int  year;  
    long mpg;  
};
```

What is the total size (in bytes) of `struct Car` (with alignment on a 64-bit machine)?

- A. 16
- B. 20
- C. 24
- D. 32

Correct Answer: C (24 bytes)

136. We see `call get_cars`, store in `%rax`, then `addq $24, %rax`, then `movq 16(%rax), %rax`. The final `%rax` is `myCars[1].mpg`.

In the assembly snippet, after `addq $24, %rax`, the instruction `movq 16(%rax), %rax` is reading which field in C syntax?

- A. `myCars[0].model`
- B. `myCars[1].mpg`
- C. `myCars[0].mpg`
- D. `myCars[1].year`

Correct Answer: B (`myCars[1].mpg`)

137. We see partial assembly:

```
3:  movq -32(%rbp), %rax
4:  leaq (%rax,%rax), %rdx    # A
5:  movq -24(%rbp), %rax
6:  addq %rdx, %rax          # B, C
...
9:  shrq $2, %rax
...
12: ret
```

Given `x=4`, `y=4`, the final return is 3.

Which final result is returned by `p2(4,4)`?

- A. 2
- B. 3
- C. 4
- D. 12

Correct Answer: B (3)

138. For a typical R-type instruction with 1 opcode & 3 register fields, how many bits are required?

- A. 16
- B. 20
- C. 24
- D. 32

Correct Answer: B (20 bits)

139. We also consider pipelined versions with 5 stages or 10 stages, each adding 1ns overhead for pipeline register.

- 5-Stage: each stage = $30/5 = 6\text{ns}$, plus 1ns = 7ns cycle time.
- 10-Stage: each stage = $30/10 = 3\text{ns}$, plus 1ns = 4ns cycle time.

If the 10-stage pipeline has each stage = 3ns plus 1ns pipeline overhead, how many total cycles does a single instruction take?

- A. 10
- B. 7
- C. 4
- D. 3

Correct Answer: A (10 stages → 10 cycles). Each cycle is 4ns, so time per instruction is $10 \times 4 = 40\text{ns}$.

140. For a pipeline with k stages, the total cycles = $k - 1 + 50$.

Given a 5-stage pipeline at 7ns per cycle = 54 cycles → 378ns.

Given a 10-stage pipeline at 4ns per cycle = 59 cycles → 236ns.

Which is the correct formula for total time?

- A. $(k - 1 + 50) \times (30 / k)$
- B. $(k + 50) \times (30 / k + 1)$
- C. $(k - 1 + 50) \times ((30/k) + 1)$
- D. $(50 / k) \times 30$

Correct Answer: C

141. The `fork()` system call creates a new process, not a new thread, within the parent process. Which statement is correct?

- A. `fork()` creates a new thread in the same process.
- B. `fork()` spawns a lightweight thread that shares memory with the parent.
- C. `fork()` duplicates the entire process, creating a new process.
- D. `fork()` only clones the heap and stack, but not the code segment.

Correct Answer: C (The `fork()` spawns a new process, not a thread.)

142. Does `malloc()` on modern systems directly allocate physical memory in DRAM?

- A. Yes, it always fetches a physical page.
- B. Yes, after zero-filling the page.
- C. No, it only arranges virtual address space; physical pages are committed lazily or upon demand.
- D. No, it only modifies environment variables.

Correct Answer: C (It's false to say it allocates physical memory directly—`malloc()` typically allocates or reserves virtual space.)

143. Which of the following is not an advantage of virtual memory?

- A. It allows multiple processes to share the same physical memory safely.
- B. It provides memory protection and isolation among processes.
- C. It ensures the CPU never stalls.
- D. It can swap pages to disk, enabling larger virtual address spaces than physical memory.

Correct Answer: C (Virtual memory does not guarantee the CPU never stalls. The other listed items are standard VM benefits.)

144. If multiple threads each hold a different lock and try to acquire the other threads' locks, which problem can arise?

- A. Race conditions
- B. Deadlock
- C. Segmentation fault
- D. Paging

Correct Answer: B (Deadlock)

145. Which normalized binary form is correct for $6 + \frac{16}{64} = 6.25$?

- A. 1.101×2^2
- B. 1.001×2^3
- C. 1.1001×2^2
- D. 1.0001×2^3

Correct Answer: C (1.1001×2^2)

146. We have:

```
int x = 0x8f7;

int* pi = &x;

float* pf = (float*) pi;

printf("%f\n", *pf);
```

Question: Which outcome is correct?

- A. It prints a NaN (not a number).
- B. It prints a floating-point number equal to $0x8f7 \approx 2295.0$.

- C. It prints a subnormal float near 0.
- D. Compilation error: cannot cast `int*` to `float*`.

Correct Answer: C (The bits likely form a subnormal or very small float value.)

146. Why do typical floating-point calculations give 0.8999999999 instead of exactly 0.9 for 0.3×3.0 ?

- A. Hardware is broken.
- B. 0.3 and 0.9 cannot be exactly represented in binary, causing rounding errors.
- C. The compiler inserts a random offset.
- D. The CPU is in single-precision mode incorrectly.

Correct Answer: B (It's due to binary approximation and rounding.)

147. We have a snippet with lines that effectively do:

```
cmpq -32(%rbp), %rax
jge .L3
movq -32(%rbp), %rax
movq %rax, -16(%rbp)
```

Which best describes the condition for updating each array element?

- A. If element is negative, set it to 0.
- B. If element < b, replace it with b.
- C. If element > b, replace it with b.
- D. If element == b, add 1.

Correct Answer: B (If the element is less than b, store b into that element.)

148. After the function runs, which final array emerges?

- A. [17, 10, 10, 10, 15]
- B. [17, 7, 10, 8, 15]
- C. [17, 10, 10, 8, 15]
- D. [10, 10, 10, 10, 10]

Correct Answer: A (The only elements <10 become 10.)

149. We have the standard 5 stages: F, D, E, M, W.

In which stage is the branch target typically resolved (i.e., known if a jump is taken)?

- A. Fetch
- B. Decode
- C. Execute
- D. Memory

Correct Answer: C (Execute stage)

150. If we want to maximize throughput (instructions/second) with no stalls, which pipeline modifications are best, according to the official key?

- A. Split the M (100ns) and W (135ns) stages for shorter stage times, and possibly combine the first 3 (F,D,E).
- B. Combine M and W for simpler logic.
- C. Combine everything into 1 stage.
- D. No changes — the original design is best.

Correct Answer: A (Splitting the longest stages yields higher frequency, combining short stages helps reduce overhead.)