1. In CISC architecture most of the complex instructions are stored in \_\_\_\_\_
2. Register
3. Diodes
4. CMOS
5. Transistor
6. In super-scalar processors, \_\_\_\_\_\_\_\_ mode of execution is used.
7. In-order
8. Post-Order
9. Out of order
10. None of above
11. The stalling of the processor due to the unavailability of the instructions is called as \_\_\_\_
12. Control hazard
13. structural hazard
14. Input hazard
15. None of the mentioned
16. The parallel mode of communication is not suitable for long devices because of \_\_\_\_\_\_
17. Timing skew
18. Memory access delay
19. Latency
20. None of the mentioned
21. In LRU, the referenced blocks counter is set to’0′ and that of the previous blocks are incremented by one and others remain same, in case of \_\_\_\_\_\_
22. Hit
23. Miss
24. Hit
25. Delayed hit
26. What is the output of the following C++ Program:

static int a = 0;  
void func(void \*arg)  
{  
 a++;  
 if(!(bool)(&arg))  
 return;  
 else  
 func(!&arg);  
}  
int main()  
{  
 func(new int);  
 std::cout << a;  
}

1. Error
2. Overflow
3. 2
4. 0
5. What is the output of the following C++ Program:

std::vector<int> v = {1,2,3};  
for (auto i: v)  
 std::cout << i << ' ';

1. No output
2. 1 2 3
3. Error
4. None of the above
5. What kind of relationship is the following C++ Code:

class SteeringWheel  
{};  
  
class Vehicle  
{   
 virtual void doStuff() = 0;   
};  
  
class Car: public Vehicle  
{  
 SteeringWheel sWheel;  
 virtual void doStuff();  
};

1. Is a
2. Has a
3. Both a and b
4. None
5. A *k-sorted array* is a nearly sorted array in which no element is more than *k* locations away from its final position in the sorted array. Thus, a 0-sorted array is completely sorted and every array of size *n* is *n*-sorted.

Suppose that *A* is a *k*-sorted array of size *n*. If insertion sort is used to sort *A*, what is the order of growth of the number of comparisons performed by the sorting algorithm in the worst case?

1. Theta (k )
2. Theta (kn)
3. Theta k2n
4. Theta n (logk n)
5. Theta n2
6. Let *T n* be defined by *T(* 0) =*T (*1 ) =4 and T[n/2]+T[n/4]+cn for all intgers n>=2

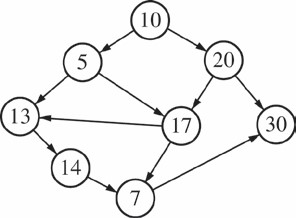
where *c* is a positive constant. What is the asymptotic growth of *T n* ?

1. log *n*
2. *n*
3. *N(* log *n)*
4. *n*2
5. *nlog3 4*
6. Which of the following refers to the associative memory?
7. The address of the data is generated by the CPU
8. The address of the data is supplied by the users
9. There is no need for an address i.e. The data is used as an address
10. The data are accessed sequentially
11. None of the above
12. A system program that combines the separately compiled modules of a program into a form
13. suitable for execution
14. Assembler
15. Linking loader
16. Cross compiler
17. Load and Go
18. None of above
19. Addressing structure
20. Defines the fundamental method of determining effective operand addresses
21. Are variations in the use of fundamental addressing structures, or some associated actions which are related to addressing.
22. Performs indicated operations on two fast registers of the machine and leave the result in one of the registers.
23. All of the above
24. None of the above
25. The Memory Buffer Register (MBR)
26. Is a hardware memory device which denotes the location of the current instruction being executed.
27. Is a group of electrical circuits (hardware), that performs the intent of instructions fetched from memory.
28. Contains the address of the memory location that is to be read from or stored into.
29. Contains a copy of the designated memory location specified by the mar after a "read" or the new contents of the memory prior to a "write".
30. The Storage-to-Storage instructions
31. Have both their operands in the main store.
32. Which perform an operation on a register operand and an operand which is located in the main store, generally leaving the result in the register, expect in the case of store operation when it is also written into the specified storage location.
33. Hich perform indicated operations on two fast registers of the machine and have the result in one of the registers
34. All of the above
35. None of the above
36. Which of the following statements is false?
37. The technique of storage compaction involves moving all occupied areas of storage to one end or other of main storage
38. Compaction does not involve relocation of programs.
39. Compaction is also know as garbage collection
40. The system must stop everything while it performs the compaction
41. None of the above
42. An algorithm’s real-time readiness (RTR) ratio is defined as the ratio of its average-case running time to its worst-case running time. Which of the following algorithms has an RTR ratio closest to 0 ?
43. Bubblesort
44. Heapsort
45. Insertion sort
46. Mergesort
47. Quicksort
48. Assumethat any n-bit positive integer x is stored as a linked list of bits so that the first element of the list is the least significant bit. For example, x = 14 ,binary 14 = 1110 is stored as the linked list 0, 1, 1, 1 of size n 4. For this data structure, the operation that replaces x by 8 (x/8) can be done in
49. theta(1) steps
50. log n steps
51. n steps
52. n (log n) steps
53. n2 steps
54. Consider the binary heap shown below that uses an array a[] = [9, 7, 4, 2, 6, 1, 3] to store its elements. What will be the values of the remaining elements in the array after one delete-max operation from the heap?
55. [7, 4, 2, 6, 1, 3]
56. [7, 4, 6, 2, 1, 3]
57. [7, 4, 6, 2, 3, 1]
58. [7, 6, 4, 2, 1, 3]
59. [7, 6, 4, 2, 3, 1]
60. Two classical algorithms for finding a minimum spanning tree in a graph are Kruskal’s algorithm and Prim’s algorithm. Which of the following are the design paradigms used by these algorithms?

|  |  |
| --- | --- |
| Kruskal’s algorithm | Prim’s algorithm |
| 1. The greedy method | The greedy method |
| 1. The greedy method | Dynamic programming |
| 1. Dynamic programming | The greedy method |
| 1. Dynamic programming | Divide and conquer |
| 1. Divide and conquer | Dynamic programming |

1. Consider the following directed graph.

Which of the following is a topological sort of the nodes of the graph?



a) 5, 7, 10, 13, 14, 17, 20, 30

b) 10, 5, 13, 14, 7, 30, 17, 20

c) 10, 5, 13, 17, 20, 14, 7, 30

d) 10, 5, 20, 13, 17, 30, 14, 7

e) 10, 20, 5, 17, 13, 14, 7, 30

22) Consider the following code

**int** Fun ( **int** n )

**if** ( n == 4 )

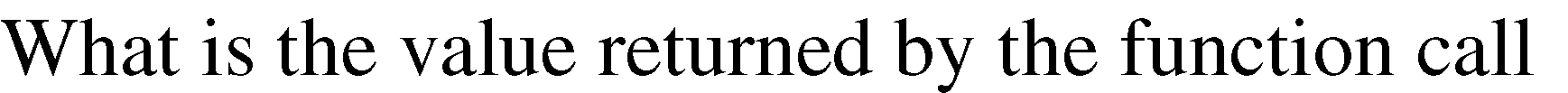
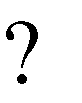
**return** 2

**else**

**return** 2 \* Fun ( n + 1 )

**end** **if**

**end** Fun

Fun ( 2 ) 

a) 16

b) 12

c) 4

d) 8

|  |
| --- |
| 23) Fragmentation of the file system |
| |  |  | | --- | --- | | a) | occurs only if the file system is used improperly | | b) | can always be prevented | | c) | can be temporarily removed by compaction | | d) | is a characteristic of all file systems | | e) | None of the above | |

24) Suppose problem *A* is **NP**-complete and problem *B* is in **NP** but is not necessarily **NP**-complete. Which of the following statements is (are) necessarily true?

1. A polynomial-time algorithm for *A* implies **P NP**.
2. II. A polynomial-time algorithm for *B* implies **P NP**.

III. A polynomial-time algorithm for *A* implies a polynomial-time algorithm for *B*.

a) I only

b) II only

c) I and II only

d)I and III only

e) I, II, and III

25) Consider the following two problems.

Nearest Neighbors: Given an unsorted array of *n* floating-point numbers as input, return two of the

numbers that are closest in value to each other.

Farthest Neighbors: Given an unsorted array of *n* floating-point numbers as input, return two of the

numbers that are farthest in value from each other.

Assume that the only operations allowed on the data are

• comparing the values of two entries in the array and identifying the larger value;

• comparing the distance between two array entries (the absolute value of the difference between the two

array entries) with the distance between two other array entries;

• swapping two entries in the array.

Further assume that each allowed operation has unit cost. What are the worst-case optimal asymptotic running

times for algorithms that solve the two problems?

Nearest Neighbors Farthest Neighbors

a) *n* log *n n*

b) *n* log *n n* log *n*

c) *n*2  *n*

d) *n*2  *n* log *n*

e) *n*2  *n*2

26) Which of the following typically occurs when a procedure call is executed on a processor?

I. Program counter is updated.

II. Stack pointer is updated.

III. Data cache is flushed to avoid aliasing problems.

a) I only

b) II only

c) I and II only

d) I and III only

e) I, II, and III

27) How non clustered index point to the data?

a) It never points to anything  
b) It points to a data row  
c) It is used for pointing data rows containing key values  
d) None of the mentioned

28) Consider a recursive algorithm for sorting an array of *n > Or =* 2 integers that works as follows.

(a) If there are only 2 elements to be sorted, compare them and swap them if they are out of order.

(b) Otherwise, do the following steps in order.

(1) Recursively sort the first *n* 1 elements of the array.

(2) In the resulting array, recursively sort the last *n* 1 elements.

(3) In the resulting array, recursively sort the first 2 elements of the array.

What is the asymptotic running time complexity of this algorithm measured in terms of the number

of comparisons made?

1. *n (*log *n)*
2. *n*2
3. *n*3
4. 2*n*
5. 3*n*

29) Which multiplexing technique transmits digital signals ?

* 1. FDM
  2. TDM
  3. WDM
  4. None of the mentioned

30) Insertion time of a Queue in worst case is

a)O(n)

b)O(1)

c)O(Log n)

d) O(n^2)

**Easy Question**

1. What is printed by the execution of TestClass.main () ?

class BaseClass{

int data = 1;

public void print () {

System.out.print ( data + " " );

}

public void fun () {

print ();

}

};

class SubClass: extends BaseClass{

int data = 2;

public void print () {

System.out.print ( data + " " );

}

};

class TestClass{

public static void main ( String[] args ) {

BaseClass obj = new SubClass ();

obj.print ();

obj.fun ();

System.out.print ( obj.data );

}

};

1. 1 1 1
2. 1 2 2
3. 2 1 1
4. 2 2 1
5. 2 2 2
6. In order to create a good solution for the mutual exclusion problem for concurrent processes, which of the

following conditions must hold?

I. No process should have to wait forever to enter its critical region.

II. No process running outside of its critical region may block other processes from entering their critical region.

III. There should be no assumptions about the speed or number of CPUs.

1. None
2. I and II only
3. I and III only
4. II and III
5. I, II, and III
6. What is the value of ***val*** in the following program:

int a = 2 << 1;  
cout << a;

1. 1
2. 2
3. 4
4. 8
5. Function overloading is to rewrite the function but with a changed return type and paramater list:
6. True
7. False
8. Is the given code an infinite loop:

for(int i = 0; i < 1; i+++i+++i---i--) {}

1. True
2. False
3. What is the value of the variable ***abc*** after the following Python Code has been executed:

abc = 1  
abc \*=3\*abc

1. 1
2. 3
3. 9
4. Error
5. Is the following C++ Code valid

class base

{};  
class derived: public base  
{};  
int main()  
{  
 base \*cls = new derived;  
}

1. True
2. False
3. What is the output of the following C++ Program:

cout << char('A' + 'Z' - 'A' + 'a' - 'Z');

1. Z
2. h
3. a
4. A
5. What is the output of the following C++ Program:

int a[] = {1, 2, 3};

cout << \*&(\*(\*(&a+NULL\*3)));

1. 1
2. 3
3. Memory Adress of a[0]
4. Garbage Value
5. What is the output of the following:

int arr[] = {1, 2, 3, 4};  
int\* p = arr;  
int\* k = p;  
cout << (\*(k + 2) + p[1] + \*(0 + arr)) << endl;

1. Garbage Value
2. Memory Address
3. 4
4. 6