GATE

Subject : CS 2005 - SOLUTIONS

(Q. NO. 1 - 30) 1 MARK

1. The given declaration
int (*f) (int *) ;
is a pointer to a function which takes one argument of type integer pointer and returns an integer.
$\therefore$ (C) is the answer.

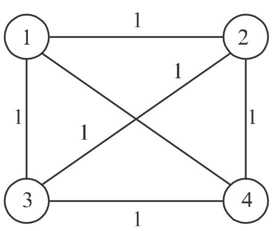
2. An abstract data type (ADT) is a mathematical model for data types where a data type is defined by its behaviour (semantics) from the point of view of a user of the data, specifically in terms of possible values, possible operations on data of this type, and behaviour of these operations.
An ADT can also be defined as a "class of objects whose logical behaviour is defined by a set of values and a set of operations."
$\therefore$ (C) is the answer.

3. Logic programming languages and functional languages both are declarative.
$\therefore$ (C) is the answer.

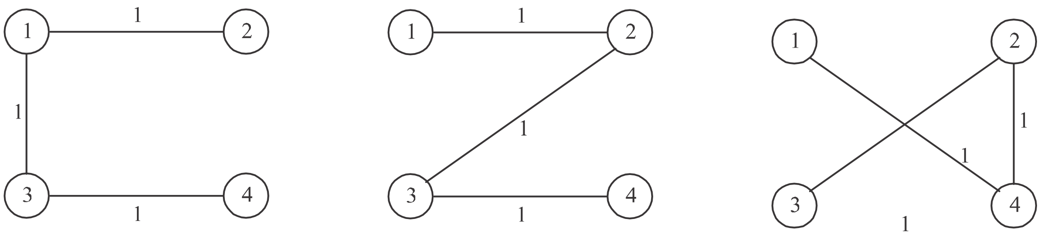
4. Essential features of an object-oriented programming (OPP) language are abstraction, encapsulation, polymorphism and inheritance.
So, only (i) and (iv) are correct.
$\therefore$ (B) is the answer.

5. The program prints the frequency of each score above 50 i.e in the range 51 to 100. So, we can easily ignore the scores below 50. Hence the best way for the program to store the frequencies would be an array of 50 numbers and to index the scores from 51 to 100, we can subtract 50 from the score value.
(A) is the answer.

6. The graph G is a complete graph on n nodes with each of its edges having weight as 1. So to minimally span n vertices we need at least (n-1) edges
$\therefore$ cost of MST = n -1
However, G will have multiple distinct MSTs with cost n-1.
consider the graph G for n=4 vertices



Few MSTs of G will be :



each having cost as 4-1 = 3

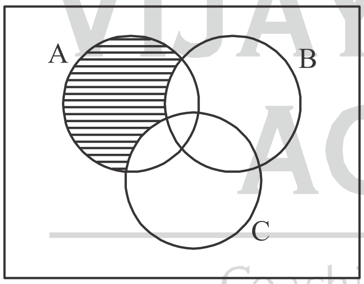
(C) is the answer

7. The transitive closure of a binary relation R on a set X is the smallest relation on X that contains R and is transitive.

Warshall's algorithm is used to constuct the transitive closure of a relation R. It is a variation of the Floyd - Warshall shortest path algorithm which is used for finding shortest paths in a weighted graph with positive or negative edge weights (but no negative cycles). It makes use of dynamic programining and has $O(n^3)$ worst case complexity where 'n' is the number of vertices in the graph.

(D) is the answer.

8. Using venn diagram, both X and Y give the following shaded area.



Thus we get X=Y

(A) is the answer.

9. The poset is a lattice because it has a unique LUB (Least Upper Bound) and GLB (Greatest Lower Bound) for each element.

However it is not a distributive lattice beacause it does not obey the distributive law for the elements b, c and d

As per the distributive law, we should have $(b \wedge c) \vee (b \wedge d)$

But,

$b \wedge (c \vee d) = b \wedge a = b$ and

$(b \wedge c) \vee (b \wedge d) = e \vee e = e$

Hence it is not a distributive lattice.

(B) is the answer.

10. Using Euler's formula for connected planar graph, we have

$v - e + f = 2$

where e - number of edges in a graph

v - number of vertices in a graph

f - number of faces / regions in a graph

Given that e = 19 and v = 13,

$\therefore\ 13 - 19 + f = 2$

$\therefore\ f = 19 - 3 + 2$

$\therefore\ \boxed{f = 8}$

So, the number of faces in the planar embedding of the graph is 8.

(B) is the answer.

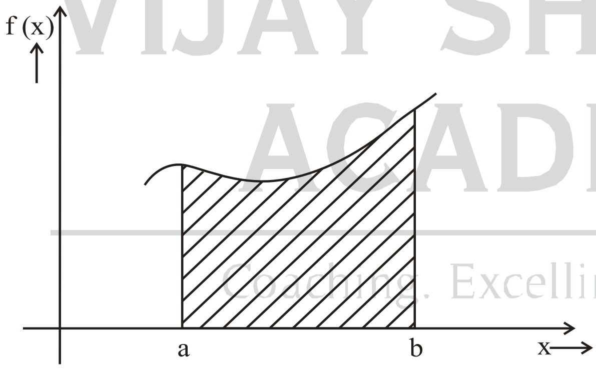
11. The number of vertices in a graph is equal to its minimum vertex cover plus the size of a maximum independent set.

Given that v=20 and the size of minimum vertex cover = 8

$\therefore$ Size of maximum independent set = 20 − 8 = 12

(A) is the answer.

12. For a random variable X, f(x) is the pdf of X. Then the probability $a < X \leq b$ is give as :



The shaded portion i.e. the area under the curve between the points x=a and x=b gives the required probability.

$\therefore P(a < X \leq b) = \int_a^b f(x)dx$

(C) is the answer

13.

$*_{15}$	1	2	4	7	8	11	13	14
1								
2								
4	4	8	1	13	2	14	7	11
7	7	14	13	4	11	2	1	8
8								
11								
13								
14								

It is already given that the set is a group under multiplication modulo 15. And we know that, for multiplication e = 1

Consider the entries of only 4 and 7

$4 *_{15} a = e$ for a to be an inverse of 4

$\therefore\ 4 *_{15} a = 1$

$\therefore\ a = 4$ (from the above table)

Similarly, the inverse of 7 is 13

(C) is the answer.

14. $A \rightarrow AA \mid (A) \mid \varepsilon$

The above grammar is left - recursive and so it is not suitable for predictive passing as the parser may run into an infinite loop while parsing.

(B) is the answer

15. $f = \overline{\overline{X\overline{Y}}\;\overline{YZ}}$

$= \overline{(\overline{X}+Y)(\overline{Y}+\overline{Z})}$

$= \overline{\overline{X}}\,\overline{\overline{Y}} + \overline{\overline{X}}\,\overline{\overline{Z}} + \overline{Y\overline{Z}}$

$= (X+Y)(X+Z)(\overline{Y}+Z)$

$= (X + \times Z + XY + YZ)(\overline{Y}+Z)$

$= (X(1+Z+Y)+YZ)(\overline{Y}+Z)$

$= (X+YZ)(\overline{Y}+Z)$

$= X\overline{Y} + XZ + YZ$

So, neither of X, Y, Z is redundant in f

(D) is the answer

16. The range of integers that can be represented by a n-bit 2's complement number system is

-2^{n-1} to $(2^{n-1}-1)$

(A) is the answer.

17. $(657)_8 = (110101111)_2$

Now, to convert it into hexadeaimal number, make groups of 4 digits each from the right end

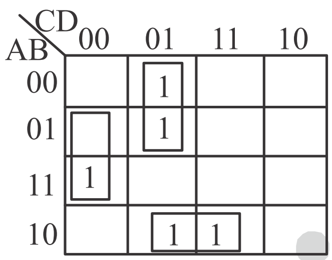
$\underbrace{0001}\underbrace{1010}\underbrace{1111}$

$(1 \quad A \quad F)_{16}$

(A) is the answer

18. $f(A,B,C,D) = \sum(1,4,5,9,11,12)$

Using K-map



$\therefore f = A\bar{B}D + B\bar{C}\bar{D} + B\bar{C}\bar{D} + \bar{A}\bar{C}D$

(A) is the answer

19. For a CPU having a single interrupt request line and a single interrupt grant line, both vectored interrupts and multiple interrupting devices are possible.

The single interrupt request input line can have multiple I/O devices connected to it. The interrupts from many devices are ORed together and then applied to the microprocessor. In response to an interrupt from any of the ORed devices, it executes the corresponding ISR and after its execution, INTA is sent through one line.

Vectored interrupts are also possible using daisy chain implementation

(C) is the answer.

20. All I/O requests are handled by the operating system through system calls which are performed in the kernel mode.

User programs do not have the previlege to perform do not have the previlege to perform I/O in user mode.

(A) is the answer.

21. Swap space in the disk is used when the amount of physical memory i.e. RAM is full. If the system, needs more memory resources and the RAM is full , inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives which have a slower access time than physical memory.

(B) is the answer.

22. Increasing the RAM size of a computer would result in a larger page table (i.e. more entries in the page table). Hence, there will be less number of page faults which would indeed improve the overall performance.

(C) is the answer.

23. Routing takes place in Network Layer and hence is independent of the Transport Layer protocols TCP and UDP. Thus the Transport Layer is transparent to the router and the routing path is determined based on the network configuration and hence paths used to route two packets for the same session may be different.

(B) is the answer.

24. The ARP (Address Resolution Protocol) is a communications protocol used for resolution of Internet Layer addresses into link layer addresses i.e. MAC addresses.

(D) is the answer.

25. In selective repeat or selective reject ARQ, the maximum window size is given as 2^{n-1} for n-bit frame sequence numbers.

(B) is the answer.

26. The Spanning Tree Protocol (STP) is a network protocol that builds a logical loop-free topology for Ethernet networks. The basic function of STP is to prevent bridge loops and broadcast radiation that results from them.

(B) is the answer.

27. Default mask of class B is 255.255.0. departments. The organization wishes to form subnets for (64) department. So, we need to borrow $\log_2(64)$ i.e. 6 subnet bits from the host bits.

So, the subnet mask becomes:

255.255.11111100.00000000

i.e. 255.255.252.0

(D) is the answer.

28. Data in B+trees is stored in the form of records in disk blocks.

However if we compare in terms of memory, B+ trees take more memory than BSTs as fanout of B+ trees is more.

(D) is the answer.

29. (A) is true. A relation in BCNF is always in 3 NF but vice versa is not true.

(B) is true. In 3NF, lossless and dependency-preserving decomposition is always possible.

(C) is false.

(D) is true. Any binary relation is always in BCNF.

30. Consider the following lossy decomposition of r into r1 and r2

Let r :

A	B	C	D
10	1	100	50
10	2	200	50
10	2	200	100

r1:

A	B	C
10	1	100
10	2	200

r2:

A	D
10	50
10	100

Now, s=r1* r2. Here * performs natural join of r1 and r2 of r1, and r2

$\therefore$ s =

A	B	C	D
10	1	100	50
10	2	200	50
10	1	100	100
10	2	200	100

Clearly, we can see that $r \subset s$

(C) is the answer.

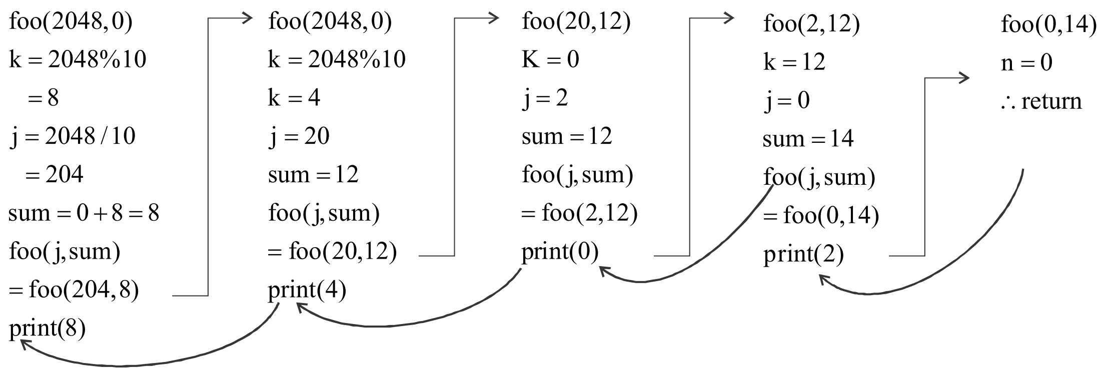
(Q. NO. 31 - 85) 2 MARKS

31. a = 2048 and sum = 0

foo (a, sum)

Here the parameters are passed by value.

So we have foo (2048,0)



∴ Output will be 2, 0, 4, 8, 0. The sum will be printed as 0 and not 14 because the variable sum is passed by value in the function foo and hence any changes made to sum will persist locally to that function without affecting its value in the main function.

(C) is the answer.

32. In C, whenever a function is called before its declaration, the compiler assumes return type of the function as int.

But the compiler assumes nothing about parameters. So, it will not be able to perform compile - time checking of argument types and arity when the function is applied to some arguments.

Here, foo has the argument as double. So the compiler throws a type mis-match error.

(D) is the answer.

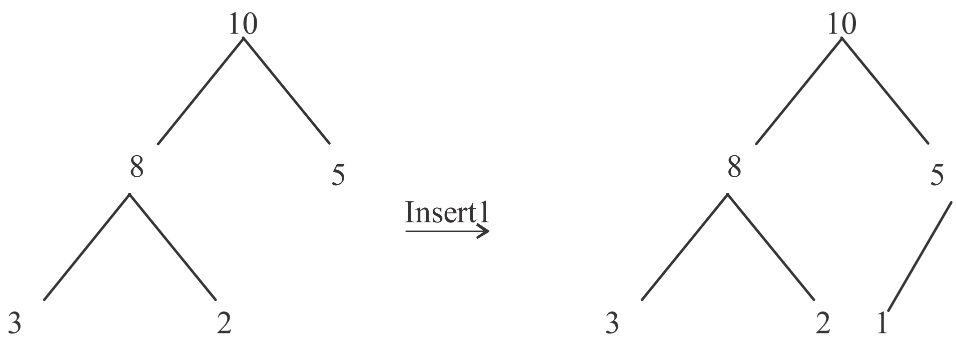
33. In-order traversal of a binary search tree always gives the sequence of elements in an increasing order.

Of the given options, only (A) has the sequence in increasing order.

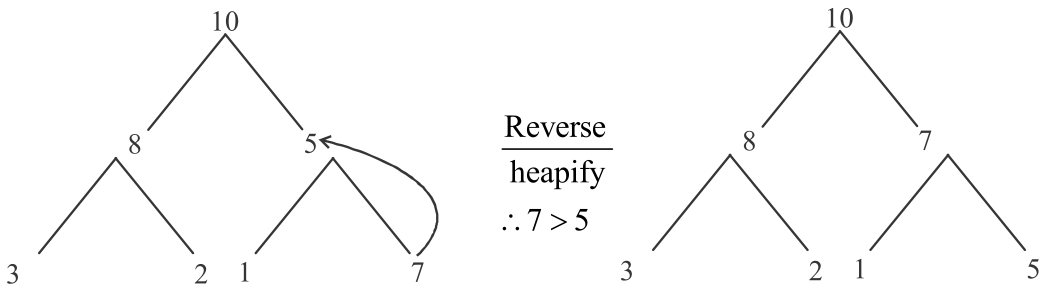
(A) is the answer.

34. Level-order traversal of the given max-heap is 10,8,5,3,2

So the heap can be constructed as :



Now insert 7



Resultant heap

$\therefore$ The level-order traversal of the heap after insertion of elements is :

10,8,7,3,2,1,5

(D) is the answer.

35. Total number of distinct binary search trees with n different elements is given by Catalan number C_n which is

$$\frac{1}{(n+1)}\ {}^{2n}C_n$$

so, for $n = 4, C_n = \frac{1}{5}\,{}^{8}C_4 = \frac{70}{5} = 14$

(B) is the answer.

36. For a complete k-ary tree,

Number of external nodes (E)=(k-1)*Number of internal nodes (I) + 1

It is given that I = n

$\therefore$ E = (k-1)*n+1

(C) is the answer

37. T(n) = 2T (n/2)+n

On solving the above recurrence relation, we get

T (n) = O(nlogn)

(A) is correct as T (n) $\leq$ c.n^2 for c>0

$\therefore$ T(n) = O(n^2)

(B) is correct as c_1.nlogn $\leq$ T(n) $\leq$ c_2.nlogn

for c_1,c_2>0

$\therefore$ T(n)= θ (nlogn)

(C) is false as there exists no c which satisfies

T(n) $\geq$ c.n^2

$\therefore T(n) \neq \Omega(n^2)$

(D) is correct

(C) is the answer.

38. For Dijkstra's single-source shortest path algorithm using a binary heap, we have |E| decrease key operations and each operations takes O(log |V|) time. Also we ahve |V| extract-min operations each taking O(log|V|) time so time complexity = O((|E|+|V|)log |V|)

    (D) is the answer.

39. Total elements to be sorted = $\log n \times \dfrac{n}{\log n} = n$

    Here we can make a min-heap of size logn by taking first elements from each of the logn sorted lists. Now we can delete each element from the heap and insert one new element from the corresponding sorted list from which the element was added into the heap So, each insert and delete will take O(log logn) time (as heap size is of logn. For a heap size of n, insert and delete operations take O(logn) time)

    $\therefore$ To perform insert and delete on n elements, time complexity = O (nloglog n)

    Alternate solution :

    We can merge k sorted arrays each of size m in O (m.k log k) time

    Here k=logn and $m = \dfrac{n}{\log n}$

    $$\therefore m.k \log k = \frac{n}{\log n} \times \log n \times \log\log n$$

    $$= n \ \log \log n$$

    (A) is the answer

40. $X \quad = (P \vee Q) \rightarrow R$

    $= \neg(P \vee Q) \vee R$

    $= \neg P \wedge \neg Q \vee R$

    $Y \quad = (P \rightarrow R) \vee (Q \rightarrow R)$

    $= \neg P \vee R \vee \neg Q \vee R$

    $= \neg P \vee \neg Q \vee R$

    (A) is false as $X \neq Y$

    Consider option (B)

    $X \rightarrow Y$

    $\therefore \big((P \vee Q) \rightarrow R\big) \rightarrow \big((P \rightarrow R) \vee (Q \rightarrow R)\big)$

    LHS $= (P \vee Q) \rightarrow R$

    $= \neg(P \vee Q) \vee R$

    $= (\neg P \wedge \neg Q) \vee R$

    $= (\neg P \vee R) \wedge (\neg Q \vee R)$

    $= (P \rightarrow R) \wedge (Q \rightarrow R)$

Now, we have

$$\big((P \rightarrow R) \wedge (Q \rightarrow R)\big) \rightarrow \big((P \rightarrow R) \vee (Q \rightarrow R)\big)$$

$$= \neg\big((P \rightarrow R) \wedge (Q \rightarrow R)\big) \vee \big(P \rightarrow R\big) \vee (Q \rightarrow R)$$

$$= \neg(P \rightarrow R) \vee \neg(Q \vee R) \vee (P \rightarrow R) \vee (Q \rightarrow R)$$

= True always $\quad [\because A \vee \neg A = \text{True}]$

So, it is a tautology

(B) is the answer.

41. (B) correctly expresses the given statement in first order predicate calculus.

(A) is wrong because it says for all teachers if X is a teacher then there exists Y such that if Y is a student, then Y likes X.

(C) is wrong as it means there exists a student who likes all teachers.

(D) is wrong because it says for every teacher X and there exists a Y such that if Y is a student then Y likes X.

(B) is the answer.

42. Equivalence relations are closed under intersection but not closed under union . So, if R and S are two equivalence relations on a non-empty set then $R \cap S$ is also an equivalence relation. But $R \cup S$ is not guaranteed to be an equivalence relation. It may or may not be an equivlence relation.

(C) is the answer.

43. A function f: $F : X \rightarrow Y$ is called on-to function if for every value in set Y, there is a value in set X

i.e. range of f = co-domain of f

Given that, $f : B \rightarrow C$ and $g : A \rightarrow B$

h = fog

= f (g(x))

$\therefore$ h is a function from A to C

Let A = {1,2,3}

B = {a,b}

C = {e}

The following values hold:

f (a) = e

g (1) = a, g (2) = a, g (3) = a

So, h would be :

h(1) = e, h(2) = e, h(3) = e

So we get f as onto function as h is onto. But g need not be an onto function.

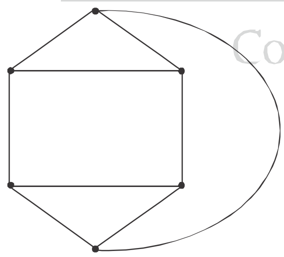
(B) is the answer.

44. Given that, a = c mod 3 i.e. a can have values 0, 1, 2
b = d mod 5 i.e b can have values 0,1,2,3,4
So, the ordered pair (a,b) can take $3 \times 5 = 15$ different values which are :
(0,0), (0,1), (0,2), (0,3),(0,4),(1,0),(1,1),(1,2),(1,3),(1,4),(2,0),(2,1),(2,2),(2,3),(2,4),
And we have one more ordered pair (c,d) so, total ordered pairs = 15+1=16
(C) is the answer.

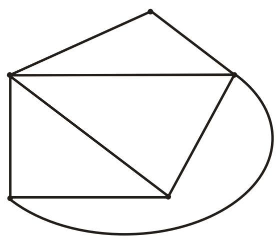
45. Consider the following concepts about decidability:
(1) If $A \leq_m B$ (i.e. A is mapping reducible to B) and if B is decidable then A is also decidable.
(2) If $A \leq_m B$ and if A is undecidable then B is also undecidable.
So, from the given options, (C) is correct using (2)
(C) is the answer.

46. The given matrix is an upper triangular matrix and determinant of such a matrix will be abc.
Given that abc $\neq 0$ i.e. the matrix is non-singular. So inverse of the matrix will definitely exist.
Also there exists an identity matrix I for all such matrices A such that
AI = IA = A
So, under matrix multiplication operation H is a group.
However, H is not an Abelian group as matrix multiplication is not commoutative.
So, (A) is the answer.

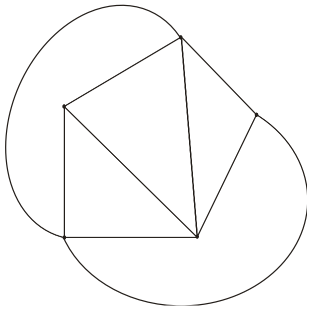
47. A graph is said to be planar if it can be redrawn in a plane without any two edges crossing each other.
G_2 can be redrawn as :



G_3 can be redrawn as :



G_4 can be redrawn as :



But there is no way in which G1 can be redrawn in a single plane without its edges crossing each other.

(A) is the answer.

48. Consider the determinant of the coefficient matrix

$$= A = \begin{bmatrix} 2 & -1 & 3 \\ 3 & 2 & 5 \\ -1 & 4 & 1 \end{bmatrix}$$

$|A| \quad = \quad 2(2 - 20) + 1(3 + 5) + 3(12 + 2)$

$\quad\quad = \quad 2(-18) + 8 + 42$

$\quad\quad = \quad -36 + 50$

$\therefore |A| \quad = \quad 14$

As the determinant of A is non-zero, rank(A)=3 which is same as number of unknowns So, we have a unique solution.

(B) is the answer.

49. Let $A = \begin{bmatrix} 2 & -1 \\ -4 & 5 \end{bmatrix}$

To calculate eigen values of A, we have

$| A - \lambda | 0$

$\therefore \begin{vmatrix} 2-\lambda & -1 \\ -4 & 5-\lambda \end{vmatrix} = 0$

$\therefore (2-\lambda)(5-\lambda) + 4 = 0$

$\therefore \lambda^2 - 7\lambda + 6 = 0$

$\therefore (\lambda - 6)(\lambda - 1) = 0$

$\therefore \lambda = 6 \text{ or } \lambda = 1$

So the eigen values of the given matrix are 1and 6

(B) is the answer

50. Consider,

$$
\begin{array}{rl}
S &= 1 + 2x + 3x^2 + 4x^3 + \ldots\ldots \\
-Sx &= \quad\;\; x + 2x^2 + 3x^3 + \ldots\ldots \\
\hline
S - Sx &= 1 + x + x^2 + x^3 + \ldots\ldots
\end{array}
$$

$$\therefore \therefore S - Sx = \frac{1}{1-x}\left(\because \text{sum of inf inite GP} = \frac{a}{1-r}.\text{Here } \; r = x\right)$$

$$\therefore S(1-x) = \frac{1}{1-x}$$

$$= \frac{1}{(1-x)^2}$$

So, for g(i) = (i+1), we get G(x) as $\frac{1}{(1-x)^2}$

(B) is the answer.

51. Probability of selecting a red ball

$$= \frac{1}{3} \times \frac{2}{5} + \frac{2}{3} \times \frac{3}{4}$$

$$= \frac{19}{30}$$

Probability of selecting a red ball from box P

$$= \frac{1}{3} \times \frac{2}{5}$$

$$= \frac{2}{15}$$

$$\therefore \text{Required probability} = \frac{2/15}{19/30}$$

$$= \frac{4}{19}$$

(A) is the answer.

52. Let the two random bit strings be S_1 and S_2 of length .

Number of possible strings generated by S_1 or S_2 of length n = 2^n (as each bit can take value either 0 or 1)

$$\therefore \text{Probability of selecting two randomly generated strings} = \frac{1}{2^n} \times \frac{1}{2^n}$$

Probability that both the selected strings are identical = $\frac{2^n}{2^n \times 2^n}$

So, the probability that the two randomly generated strings are not identical = = $1 - \frac{1}{2^n}$

(D) is the answer

53. L(M) = {abb,abbb,babb,babbabb,...}
(A) is false as the string abbb is accepted by M.
(B) is true. M accepts all strings where every a is followed by at least two bs
(C) is false as M does not accept aabb
(D) is false as M does not accept b

(B) is the answer.

54. Both NFA and DFA have the same expressive power i.e. both accept the same set of languages.
$\therefore D_f = N_f$
Deterministic PDA accepts all deterministic CFLs whereas non-deterministic PDA accepts all CFLs.
$\therefore D_p \subset N_p$
(D) is the answer.

55. $L_1 = \{a^n b^n c^n \mid n, m > 0\}$
It is a CFL PDA can be constructed as follows :
For each a, push one X into the stack
For each b, pop one X from the stack
For each c, we perform no-operation.

$L_2 = \{a^n b^m c^m \mid n, m > 0\}$
It is also a CFL. PDA can be constructed as follows:
For each a, perform no operation
For each b, push one X into the stack
For each c, pop one X from the stack

(A) is false as CFLs are not closed under intersection Rest all the statements are true.
(A) is the answer.

56. • If L_1 is a recursive language, then $\overline{L}_1$ is also recursive because reaursive languages are closed under complementation.

- If L_2 is a recursively enumerable language then $\overline{L}_2$ is not a recursively enumerable language as RELs are not closed under complementation.

(B) is the answer.

57. $L_{1=}\{ ww^R | w \in \{ 0, 1 \}^* \}$

It is a CFL but not a DCFL as there is no way we can determine the string w beforhand.

$L2 = \{w\#w^R | w \in \{ 0, 1 \}^* \}$

It is a DCFL because of the special symbol # here. We can distinctly determine the string w here until # is encountered.

$L_3=\{ww \mid w \in \{0,1\}^*\}$

It is a CSL.

(B) is the answer.

58. Both α and β are based on graph independent set decision problem. And graph independent set decision problem is NP-Complete.

(C) is the answer.

59. Let w $= n + n \times n$

$= E + n \times n \qquad (\because E \to n)$

$= E \times n \qquad (\because E \to E + n)$

$= E \qquad (\because E \to E \times n)$

So we get 3 handles which are n, E+n and E n

(D) is the answer.

60. The number of states in SLR (1) grammar and LALR (1) grammar are always equal.

$\therefore$ n1 = n3

The parsing table of LALR(1) grammar is formed by merging the states of LR(1) {or CLR (1)} grammar.

$\therefore$ n3 < n2

So, we get n1 = n3 < n2

(B) is the answer.

61. The given C program will give only a syntactic error for the ‘for’ loop as a syntactically correct for loop must have two semicolons inside ( ).

Also note that here it is ‘fro’ instead of for. This is not a lexical error as lexical analysis only involves generation of tokens. However, we may get compiler warnings but no compiler errors.

(C) is the answer.

62. From the given circuit, we get

$D = AX + X'Q'$

$Y = D$

It is given that A_i represents the logic level on the line A in the i-th clock period. As the flip flop is positive edge triggered, so during the first clock cycle (which is the rising edge of the clock), the D flip flop will change its state.

So, for 1st clock period , $X = 1$ $\therefore D = A = A_0$

2nd clock period , $X = 1$ $\therefore D = A = A_1$

3rd clock period , $X = 0$ $\therefore D = Q_2' = A_1$

4th clock period , $X = 1$ $\therefore D = A = A_3$

5th clock period , $X = 1$ $\therefore D = A = A_4$

So, the sequence of output on Y is $A_0 A_1 A'_1 A_3 A_4$

(A) is the answer.

63. The given finite state machin takes as input a binary number from the LSB.

Assume the input string w as 10110101

Let's calculate the output string λ. In order to take input from LSB, we need to reverse to the given string. So, the output λ will be as follows

λ = $Q_0 10101101$

= $1Q_1 0101101$

= $11Q_1 101101$

= $110Q_1 01101$

= $1101Q_1 1101$

= $11010Q_1 101$

= $110100Q_1 01$

= $1101001Q_1 1$

= 11010010

Now reverse the output string as this output is calculated from LSB to MSB

So, for the string w=10110101, the output is 01001011 which is nothing but the 2's complement of the input number.

(A) is the answer.

64. On analyzing the given circutit, we get

$D_0 = Q'_0$

$\therefore Q_{0new} = Q'_{0old}$

$D1 = Q_0 \oplus Q'_1$

$Q_{1new} = Q_{0old} \oplus Q'_{1old}$

Q_0	Q_1	Q_{0new}	Q_{1new}
0	0	1	1
1	1	0	1
0	1	1	0
1	0	0	0

So, the state transition sequence is

00 → 11 → 01 → 10 → (back to 00)

(D) is the answer.

65. The given instruction ADD is executed as

$A[R0] \leftarrow A[R0] + @B$

We need

- 1 memory read to get the first operand from memory address A+R0
- 1 memory read to get address of second operand using indirect addressing.
- 1 memory read to get the second operand from the address of previous memory read.
- 1 memory write to store the result in the first operand i.e. the destination.

So, in all we need 4 memory cycles to execute the given instruction.

(B) is the answer.

66. (1) A[I] = B[J];

Here indexing is used. ∴ (1) - (b)

(2) While [*A++];

Here, the memory locations are automatically incremented.

(3) int temp = *x; ∴ (2) - (c)

Here, we are storing the value at address of x in temp which has type as int. We are making use of pointere which support indirect addressing ∴ (3) - (a)

(C) is the answer.

67. For direct mapped cache, the format is :

tag	line offset	block offset

←——— 32 ———→

Block size = 32 B

$\therefore$ block offset = $\log_2 \lceil \text{Block size} \rceil$

$= \log_2 \lceil 32 \rceil$

$= 5$ bits

#cache lines $= \dfrac{\text{cache size}}{\text{block size}}$

$= \dfrac{32\,\text{KB}}{32\,\text{B}}$

$= 1$ K

$= 2^{10}$

$\therefore$ Line offset $= \log_2 \lceil \#\text{cachelines} \rceil$

$= \log_2 \lceil 2^{10} \rceil$

$= 10$ bits

$\therefore$ tag bits $= 32\text{-}15$

$= 17$ bits

So, we need 10 bits for cache indexing as we have 2^{10} different lines in cache and 17 bits for tag (A) is the answer.

68. Using operand forwarding technique, the pipeline is as follows:

	C1	C2	C3	C4	C5	C6	C7	C8
I_1 :	1F	RD	EX	MA	WB			
I_2 :		1F	RD		EX	MA	WB	
I_3 :			1F	RD		EX	MA	WB

so we need 8 clock cycles to complete the execution of given instructions.
(A) is the answer.

69. In the programmed I/O, CPU remains in waiting state until I/O operations are complete
Data transfer rate = 10 KB / sec.
i.e. 10 KB data is transferred in 1 second

$\therefore$ 1 B data transfer time $= \dfrac{1}{10\text{K}}$ sec

$= 100 \times 10^{-6}\ \mu$sec

Interrupt overhead is given as 4 µsec.
In interrupt mode, data is transfered byte-wise.
So, 1B overhead is 4 µsec

$\therefore$ Performance gain $= \dfrac{100\times10^{-6}\text{ sec}}{4\times10^{-6}\text{ sec}}$

$= 25$

(B) is the answer.

70. $RPS = \dfrac{3000}{60} = 50$

Rotational delay $= \dfrac{1}{RPS} = \dfrac{1}{50}$ sec

= 20 msec

The disk reads 512*1024 B in one rotation i.e. in 20 msec.

$\therefore$ To read 4B data, we need time= $\dfrac{20\times10^{-3}\times4\not{B}}{512*1024\not{B}}$

= 152.58 nsec.

Memory cycle time is given as 40 nsec.

$\therefore$ % of time CPU gets blocked during DMA operation

$= \dfrac{40\text{ nsec}}{152.58\text{ nsec}}$

$=$ 26.21%

Closest option is 25

So, (B) is the answer.

71. To ensure that deadlock never occurs, we allocate each process its peak demands i.e. its (maximum requirement - 1) and 1 extra resource so that at least one of them can complete their requirment.

$\therefore \Sigma(S_i - 1) + 1 <= m$

$\therefore \Sigma_{i=1}^{n} S_i - \Sigma_{i=1}^{n} 1 + 1 <= m$

$\therefore \Sigma S_i - n + 1 <= m$

$\therefore \Sigma S_i + 1 <= m + n$

$\therefore \Sigma S_i < m + n$

(C) is the answer.

72. fork returns 0 when it encounters a child process

Let a = 10 initially

For child process,

a = a + 5 = 15

For parent process

$a = a - 5 = 5$

$\therefore u = 5$ and $x = 15$

So, we get $u + 10 = x$

However, $v = y$ as both parent and child share the same logical address.

(C) is the answer.

73. Let 24 B data be divided into m packets each of size 'x' B

$\therefore$ Number of packets (m) $= \dfrac{24}{x}$

$\therefore$ Packet size = (x + 3) B

Total transmission time $= 3(x+3) + (m-1)(x+3)$

$= 3x + 9 + mx + 3m - x - 3$

$= 2x + mx + 3m + 6$

We know that $\left[m = \dfrac{24}{x} \right]$

$\therefore$ Time as a function of x i.e. $T(x) = 2x + \dfrac{24}{x} \times x + 3 \times \dfrac{24}{x} + 6.$

$T(x) = 2x + \dfrac{72}{x} + 30$

To find the optimal packet size,

$T'(x) = 0$

$\therefore 2 - \dfrac{72}{x^2} = 0$

$\therefore x^2 = 36$

$\therefore \boxed{x = 6}$

Including 3B of header for each packet,

optimal packet size = 6+3=9B

(D) is the answer

74. In the worst case the collision is detected when the start of the frame is about to reach the receiver and the receiver starts sending data. So the collision happens and a jam signal is produced to send it to the sender.

$\therefore$ RTT = Time for first bit of the frame to reach the receiver + transmission time for jam signal

$= 46.4\mu s + \dfrac{48b}{10\,\text{Mbps}}$

$= 46.4\mu\text{sec} + 4.8\mu\text{sec}$

$= 51.2\ \mu\text{sec}$

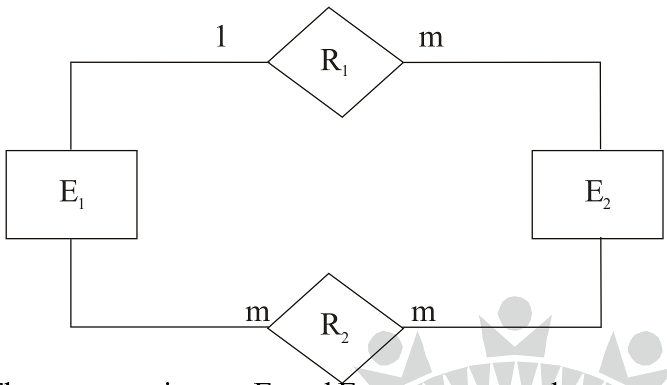
$\therefore$ Minimum frame length must be such that its transmission time must be more than 51.2 usec

So, $L_{min} = 51.2 \times 10^{-6}s \times 10 \times 10^{6}$ bps

$= 512$ bits

(D) is the answer

75. `



The strong entity sets E_1 and E_2 are represented as separate tables.

$E_1 = \{\underline{a_1}, a_2\}$ and $E_2 = \{\underline{b_1}, b_2\}$

For one to many relationship R_1, the primary key on the 'one side' of the relationship is added to the 'many side' as a foreign key. So, it does not need a separate table and we have,

$R_1 = \{b_1, b_2, a_1\}$

For many to many relationships, we have a composite key formed by primary keys of each of the entities taking part in the many to many relationship.

So, we have

$R_2 = \{\underline{a_1, b_1}\}$

Here, in all we need 3 tables

$E_1 = \{\underline{a_1}, a_2\}$

$E_2 = \{\underline{b_1}, b_2\}$

$R_2 = \{\underline{a_1, b_1}\}$

(B) is the answer.

76. When the tuple (2,4) is deleted, we have to delete its foreign key occurrences as well i.e. (5,2) and (7,2).

Since we are deleting 5 and 7, we have to delete there foreign key occurrences as well i.e. the tuple (9,5)

However, there is no foreign key occurence for 9.

So, we need to delete 3 tuples which are (5,2), (7,2) and (9,5)

(C) is the answer.

77. The outer query selects all titles from the book table. For every selected book, the inner subquery returns the count of those books which are more expensive than the selected book So, count (*) will be 0 for the most expensive book, count(*) will be 0 for the second most expensive book.

Similarly count (*) will be 4 for the 5th most expensive book.

So the where clause of the outer query will return true for 5 most expensive books.

(D) is the answer.

78. R=(A,B,C,D,E,H)

Functional dependencies are $\{A \rightarrow B\}, BC \rightarrow D, E \rightarrow C, D \rightarrow A\}$

Here H is a multivalued attribute as it does not appear in any of the functional dependencies

Consider

$(A\,E)^+ = \{A\,B\,C\,D\,E\}$

$\therefore (AEH)^+ = \{ABCDEH\}$

So, AEH is a super key.

However no proper subset of AEH determines all the attributes of R distincly.

Hence, AEH is a candidate key.

Also $D \rightarrow A$ (non key $\rightarrow$ prime attribute)

$\therefore DEH \rightarrow AEH$

So, DEH is also a candidate key

Again $BC \rightarrow D$

$\therefore BCEH \rightarrow DCEH$

So, BCEH is a super key.

Let us consider one of its proper subset as BEH.

$\{BEH\}^+ = \{BEHCDA\}$

$\therefore$ BEH is a candidate key

So in all, we get 3 candidate keys as AEH, BEH and DEH

(D) is the answer.

79. Instruction fetch will require 2 cycles Execution will be as follows :

(1) $R0_{out}$, Sin

(2) $R1_{out}$, Tin

(3) S_{out}, T_{out} Add $R0_{in}$

$\therefore$ We need 3 cycles for execution,

(B) is the answer.

80. Following is the requence of execution

(1) $\left.\begin{array}{c} MAR \leftarrow PC \\ S \leftarrow PC \end{array}\right\}$ 1cycle

Since the above two actions are independent, they can be done in the same cycle.

(2) MDR $\leftarrow$ M[MAR]- takes place in system bus

$R_N \leftarrow$ S+1- takes place in internal bus.

Even the above two actions are independent

(3) PC $\leftarrow$ MDR

So, we need 3 cycles

(B) is the answer

81. (a) The maximum space complexity will be for the last iteration of the for loop when the recursive calls foo (n-1), foo (n-2), ...foo(1), foo(0) will be active at that instant.

So we need o(n) space to store these n function calls.

(B) is the answer.

(b) Modifying the function foo won't change the space complexity of the function as we still required O(n) space to store the values of n function calls.

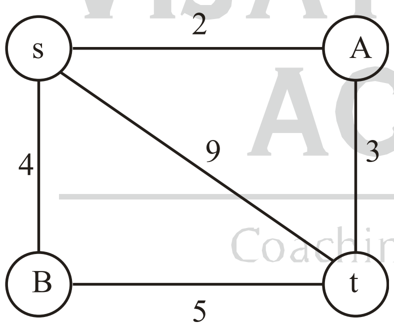
(B) is the answer.

82. (a) The cut property of Minimum Spanning Tree states that :

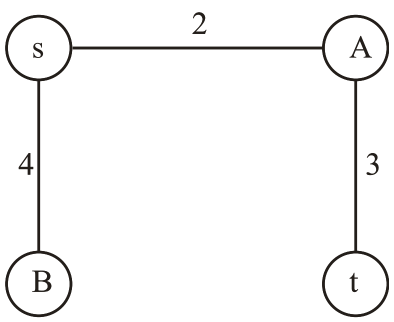
For any cut C of the graph if the weight of an edge e in the cut-set of C is strictly smaller than the weights of all other edges of the cut-set of C, then this edge belongs to all MSTs of the graph.

So, (A) is the answer.

(b) Consider the following graph



Spanning tree of the above graph will be :



Consider the paths from A to B in the graph we have 2 paths

1) A-t-B having weight = 3+5+8

2) A-s-B having weight = 2+4=6

But the MST includes the path A-s-B which is the path of minimum congestion.

(A) is the answer.

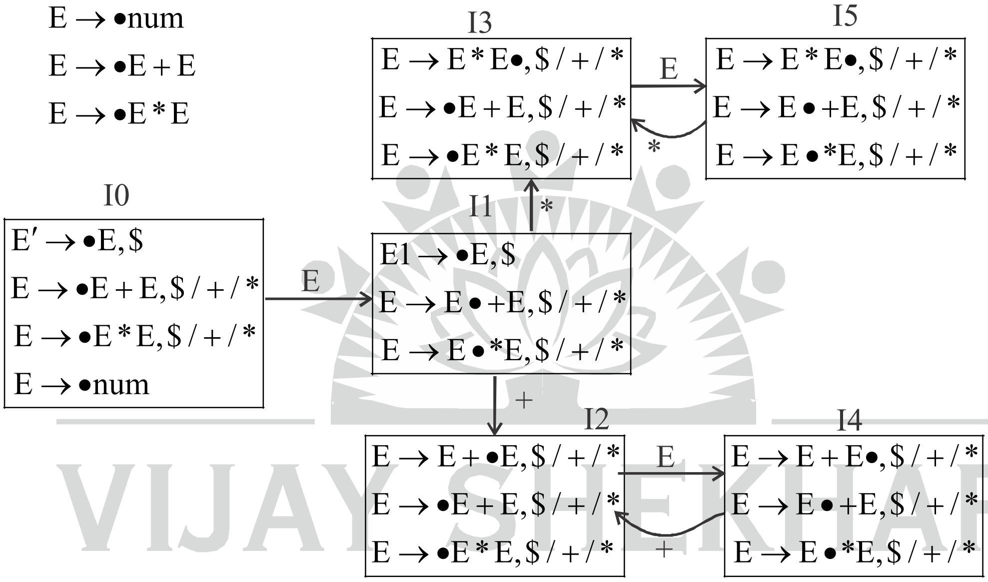
83. (a) Consider the LR(1) sets of items for the given grammar Augmented grammar is

$E' \to \bullet E$

$E \to \bullet num$

$E \to \bullet E + E$

$E \to \bullet E * E$



So, we have 2 SR conflicts in states I4 and I5 over + and *

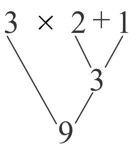
On SR conflict, YACC by default resolves the conflict in favour of a shift over a reduce action

(c) is the answer.

(b) The given grammar is ambiguous. Hence, we have equal precedence for + and $\times$. Also there will be equal associativity.

As YACC by default perfers shift over reduce, we will shift until the last operator and then reduce operations will start.

$\therefore$ The given expression E =3 × 2+1 will be evaluated as :



So, we can conclude it has right associativity.

(B) is the answer.

84. (a) To maximize the profit, the tasks are completed in the following order:
T7, T2, T9, T5, T3, T8, T1
we can see that the tasks T4 and T6 are left out.
(D) is the answer.

(b)Maximum profit is easned when the following tasks are completed

Task	T7	T2	T9	T5	T3	T8	T1
Profit	23	20	25	18	30	16	15

So total profit = 23+20+25+18+30+16+15 = 147
(A) is the answer.

85. a) For the decimal numbar 0.239×2^{13}
Sign bit = 0
Exponent = 13
Excess - 64 exponent = 13 + 64 = 77
$\therefore (77)_{10} = (1001101)_2$
Binary of 0.239

0.239 x 2 0
0.478 x 2 0
0.956 x 2 1
0.912 x 2 1
0.842 x 2 1
0.648 x 2 1
0.296 x 2 0
0.592 x 2 1 ↓

$\therefore$ Mantissa will have 8 bits as 00111101
So in floating point format, we have.

0	1001101	00111101	
0100	1101	0011	1101
4	D	3	D

(D) is the answer.

(b) $N = 0.239 \times 2^{13}$
Binary form of 0.239 = $(0.00111101)_2$
Normalized form = $(1.11101)_2 \times 2^{10}$
$\therefore$ Excess - 64 exponent = 10 + 64 = 74
Binary of 74 = $(1001010)_2$

Mantissa (8 bit) = 11101000

3 bits padding

Sign - bit = 0

So, in the floating point format, we have

0	1001010	11101000	
0100	1010	1110	1000
4	A	E	8

(D) is the answer.