Answer Key's

1	B	**2**	A	**3**	A	**4**	C	**5**	B	**6**	D	**7**	D
8	D	**9**	C	**10**	D	**11**	B	**12**	C	**13**	C	**14**	B
15	D	**16**	B	**17**	A	**18**	A	**19**	B	**20**	C	**21**	B
22	C	**23**	D	**24**	A	**25**	A	**26**	A	**27**	A	**28**	D
29	B	**30**	A	**31**	A	**32**	B	**33**	B	**34**	B	**35**	A
36	C	**37**	A	**38**	C	**39**	A	**40**	A	**41**	A	**42**	B
43	B	**44**	A	**45**	D	**46**	A	**47**	B	**48**	C	**49**	A
50	B	**51**	B	**52**	C	**53**	A	**54**	A	**55**	C	**56**	B
57	B	**58**	A	**59**	C	**60**	C	**61**	B	**62**	B	**63**	A
64	A	**65**	B	**66**	A	**67**	D	**68**	B	**69**	A	**70**	B
71	C	**72**	D	**73**	A	**74**	C	**75**	D	**76**	B	**77**	A
78	C	**79**	B	**80**	B	**81**	B	**82**	B	**83**	D	**84**	A
85	B												

Explanation:-

1. Let matrix A is of order $m \times n$

   Let matrix B is of order $n \times p$

   For matrix product AB,

   The number of multiplication operations = mpn

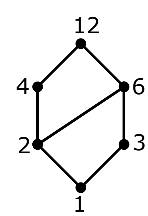
   The number of addition operations = mp(n − 1)

   Here m = 3, n = 4, p = 5

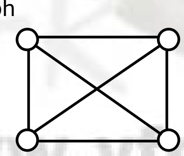
5. Let L be a bounded lattice with greatest element I and least element 0, and $a \in L$.

   An element $a' \in L$ is called a complement of a if $a \vee a' = I$ and $a \wedge a' = 0$

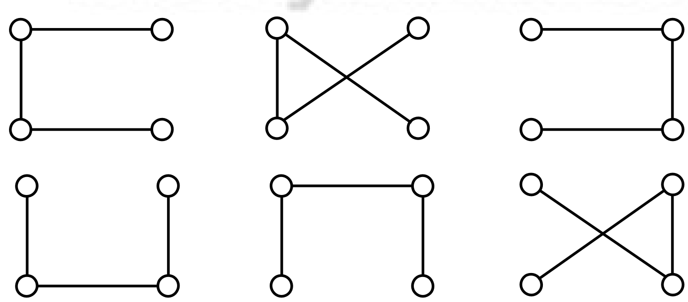
   Here I = 12, 0 = 1



6. (i) Consider the following graph



   Spanning trees are:



(ii) Adding an edge will always creates a cycle because there will be always a path from one vertex to another where u will add the new edge.

(iii) Prim's and Kruskal's algorithm both will add only those edes that will not create cycle otherwise it will no longer a tree.

7. Recursive enumrable are closed under Kleene star Concatenation, union and intersection.

8. (A) is the condition to avoid No preemption, (B) is the condition to avoid Circular wait (C) is the the condition to avoid hold and wait.

9.

x \ yz	00	01	11	10
0	1			1
1	1			1

10. Pipelined suffers from different hazard, (A) describe control hazard (B) is data hazard and (C) is structure hazard; All affects the performance of pipeline.

11. Each character takes 1 + 8 + 2 = 11 bits to transfer. The baud rate is 8800, i.e., 8800 bits per second. So 800 characters can be transmitted per second

12. The packet addressing should be taken care by network layer (IP), dividing the transmitted bit in to frames is done by data link layer and only UDP is connectionless protocol.

13. Both the regular expressions are equivalent to (a + b)*. So S = T

14. X = -11, Y = 8 and Z = -2. So $\frac{(XY)}{Z}$ = 44 = 00101100 (binary)

15. At least 2 edges should have same weight to get multiple minimum spanning trees. Here we are not adding any duplicate weighted edge to the graph. So it is not possible to get more than one minimal spanning tree.

16. According to the binary search tree the left child value should always be less than root node value and the right node value should always be greater than the root node. So, the tree should be visited in in-order to get the numbers in ascending order.

17. There are 6-bits for subnet, but we have to remove the first(all zeros) and last (all once)combinations for both subnet and host id's.

18. $130.50.\underbrace{\qquad\qquad}_{8\ \text{bits}}.\underbrace{\qquad\qquad}_{8\ \text{bits}}$

In this 16 the first 6 are for subnet address and the remaining 10 are for host ids. We shouldn't take the first combination of $130.50.\underbrace{00000000}_{\text{binary}}.00000000$

So the second choice is $130.50.\underbrace{00000100.00000001}_{\text{both are in binary}}$

So the first subnet address is 130.50.4.1 and the 4th subnet address is 130.50.00010000.00000001 which is nothing but 130.50.16.1.

19. $\frac{900}{10+3+1+1} = \frac{900}{15} = 60$ character per second.

20. 5 channels each of 100KHz.
Total of 5×10=500 KHz bw.
So there are 4 guard band of 5 KHz
Total of 5 02154=20KHz.
So min needed bw is 500+20=520KHz.

21. Counter example: R = {(a,b) (a,c)}. The transitive closure of the symmetric closure of R is {(a,a) (a,b) (a,c) (b,a) (b,b) (b,c) (c,a) (c,b)}. The symmetric closure of the transitive closure of this relation is {(a,b) (a,c) (b,a) (c,a)}

22. $S_1 : f(x) = x^3$ on 2

For any two integers a, b

$f(a) = f(b) \Rightarrow a^3 = b^3 \Rightarrow a = b$

Two diff elements have two diff images

$\therefore$ it is $1-1$

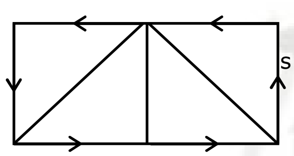
$y = f(x) = x^3,\ x = y^{1/3}$ need n't be integer

$\therefore$ it is not on to

23.

$+_6$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

24. Not Eulerian as edge repetition is there.



s Hamiltonian

26. $P(B \cap C) = P(B) - P(A \cap B \cap \overline{C}) - P(\overline{A} \cap B \cap \overline{C}) = \frac{1}{12}$

27. First sort the degree vertex which gives us (5, 4, 3, 3, 2). Let label these vertices as (a, b, c, d, e) respectively and the new vertex (to be added) as 'f'. The vertex 'a' (degree 5) should be adjacent to 5 vertices. This means 'f' vertex should have a degree of at least 1. The second vertex should be adjacent to another 3 vertices as vertex 'a' is already adjacent to this contributing 1 degree. We will take these 3 vertices as (c, d, e). The vertex 'c' should be adjacent to another vertex as 'a' and 'b' are already adjacent to this. This we can take as 'd'. With this all degrees are satisfied for all vertices. So the new vertex 'f' should have a degree of 1.

28. The required probability $= \frac{1}{6}.\frac{5}{6} + \frac{1}{6}.\left(\frac{5}{6}\right)^3 + \ldots \quad = \frac{1}{6}.\frac{5}{6}\left[1 + \left(\frac{5}{6}\right)^2 + \ldots\right]$

29.

x	0	1	2	3	4	5	6
f(x)	1	0.5	0.2	0.1	0.0588	0.0385	0.027
	y_0	y_1	y_2	y_3	y_4	y_5	y_6

By Simpson's $\frac{3}{8}$ rule

$$\int_0^6 \frac{dx}{1+x^2} = \frac{3h}{8}\left[(y_0+y_6)+3(y_1+y_2+y_4+y_5)+2y_3\right] = 1.3571$$

30. Let x_1, x2 , …. Be the variables representing the numbers on the first, second, .. 6th ticket. The probability of drawing a ticket out of 6 tickets is

1/6+ 2. 1/6+ 3.1/6 +4. 1/6 + 5. 1/6 + 6. 6/6=7/2

Expected value of the sum of the numbers on the tickets drawn = E(x_1+ x_2)= E(x_1) + E(x_2) = ½ 2. (6+1) = 7

31. If we have to prove that it is a contradiction then $\left[(p \to r) \wedge (q \to r)\right]$ should be true and $\left[(p \vee q) \to r\right]$ should be false. Then only the given preposition become false. In all other cases it is true according to the definition of "→" operator. Let us take $\left[(p \to r) \wedge (q \to r)\right]$ as preposition (1) and $\left[(p \vee q) \to r\right]$ as preposition (2).

Preposition (2) is false only if

$(p \vee q) - T \ldots(3)$

$r \quad - F \ldots(4)$

Preposition (1) is true only if $(p \to r)$ and $(q \to r)$ both are true. But r is false according to the equation (4). So p and q both should be false to make $(p \to r)$ and $(q \to r)$ True which doesn't satisfy equation (3). So the given preposition is a tautology.

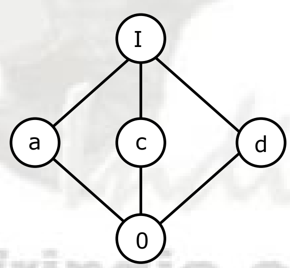
33. $a_n = a_{n-1} + 3^n$

$a_{n-1} = a_{n-2} + 3^{n-1}, a_{n-2} = a_{n-3} + 3^{n-2}, \underline{a_1 = a_0 + 3^1}$

$$a_n = 3^0 + 3^1 + 3^2 + \ldots + 3^{n-2} + 3^{n-1} + 3^n = \frac{1\left(3^{(n+1)} - 1\right)}{2} = \frac{3^{(n+1)} - 1}{2}$$

34. (A) is not true because dve doesn't exist.

In case of (C),



$a \wedge (c \vee d)$

$= a \wedge I = a$

$(a \wedge c) \vee (a \wedge d)$

$= 0 \vee 0 = 0$

Hence not distributive.

35. Suppose $x \in (A - B) - C$. Then x is in A-B but not in C. Since $x \in (A - B)$, $x \in A$ but $x \notin C$, hence $x \in (A - C)$

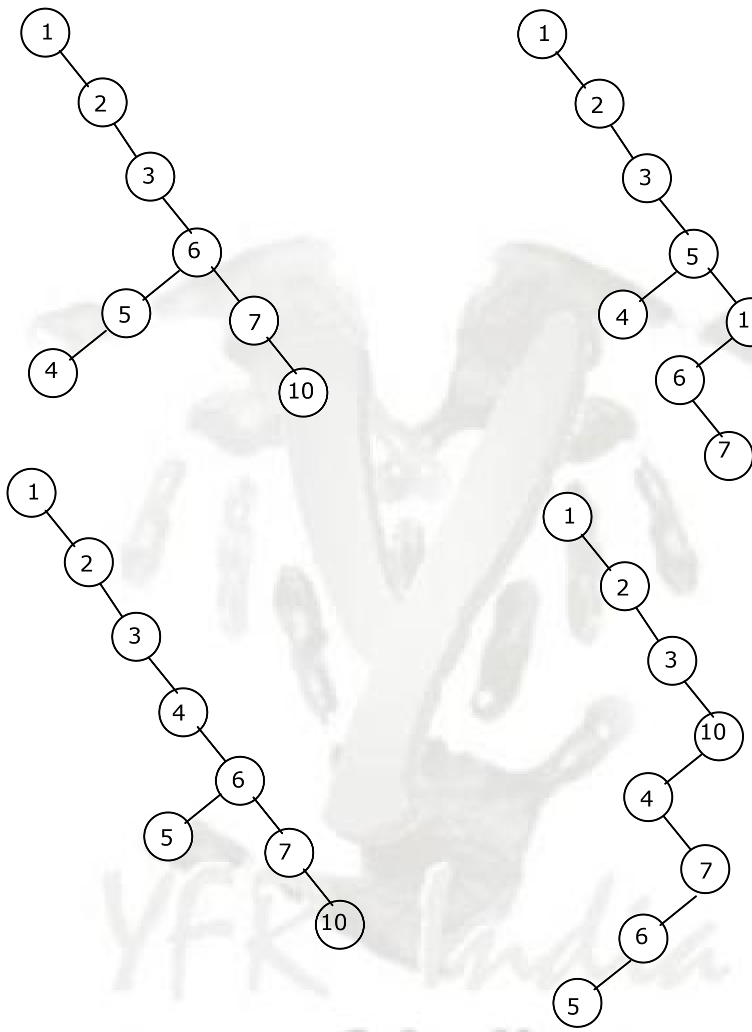
b) Let $x \in (A - C) \cap (C - B)$. Then $x \in (A - C)$ and $x \in (C - B)$

Hence $x \in C$ and also $x \notin C$, which is contradictory.

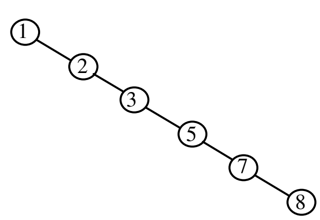
36. First choose the position for p and q; this can be done in C(6,2) ways, since once we pick two positions, we put p in the left-most and q in the other. The four remaining positions can be filled in P(24,4) ways.

Hence answer is C(6,2)P(24,4)

37.

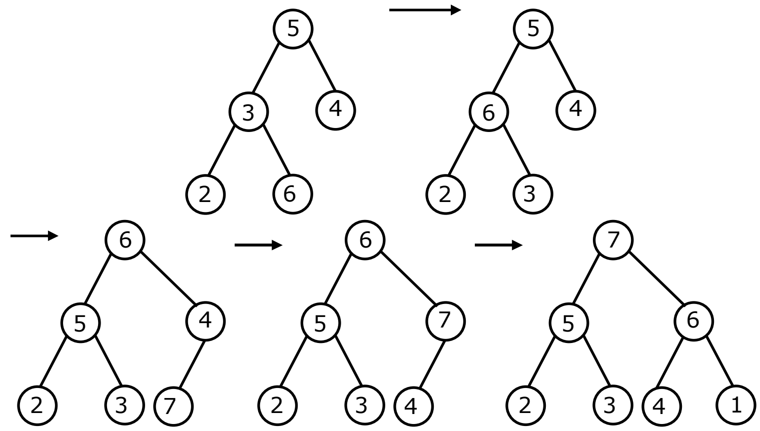


38. The choice (C) gives the following binary search tree in which both in order and preorder traversals are same. The sequence should always be in increasing or decreasing order to get such property.



39. Enqueue operation is O(1) and in Dequeue all the remaining element have to be shifted so O(n).

40. Follow the heap insertion algorithm and do an in-order traversal on the constructed heap.



41. FCFS 1st goes to 43 track

requests are handled in order of arrival

$\Rightarrow$ total no of tracks = 473

SSTF 1st goes to track 44

total no of tracks = 217 tracks

sequence is 44, 43, 158, 175, 203.

Look 1st 158

total no of racks = 263 tracks.

42.

Include vertex	Rechable vertex	A	B	C	D	E	F	G	H
$\{A\}$	$\{B, C\}$		2	15					
$\{A, B\}$	$\{D, E\}$		2	12	11	16			
$\{A, B, D\}$	$\{E\}$		2	12	11	16			
$\{A, B, D, C\}$	$\{F\}$		2	12	11	16	22		
$\{A, B, D, C, E\}$	$\{G\}$		2	12	11	16	22	33	
$\{A, B, D, C, E, F\}$	$\{G, H\}$		2	12	11	16	22	32	32
$\{A, B, D, C, E, F, H\}$	$\{G\}$		2	12	11	16	22	32	32
$\{A, B, D, C, E, F, H, G\}$			2	12	11	16	22	32	32

43. The program does a simple 'pre-order' traversal on the graph. The 'value' is incremented whenever both left and right child are NULL. So, the program finds out the number of leaves in the given graph.

44. There is no condition while swapping the links. So it swaps the left and right children of every node.

46. The write happened in function f gets destructed as soon as the function call is over. So, it prints the previous value of 'GATE'.

48. The function corresponds to the recursive equation $f_n = f_{n-1} + 1$. By solving this we get O(n),

49. To have a FA to check for divisibility by two we should have two states. To check for the divisibility by 3 we need 3 states. In the final FSM the states consist of the combination of FSM1, and FSM2. So the final FSM would have 6 states.

State	0	1
Q_s	Q_0	Q_1
Q_0	Q_{00}	Q_{01}
Q_1	Q_{01}	Q_{11}
Q_{00}	Q_0	Q_1
Q_{01}	Q_1	Q_{011}
Q_{11}	Q_{011}	Q_{00}
Q_{011}	Q_{11}	Q_0
Q_{111}	Q_0	Q_1

These three sate will combine

50. State S_4 denotes 'odd number of a's and even number of b's.

S_3 – odd number of a's and odd number of b's.

S_2 – even number of a's and odd number of b's.

S_1 – even number of a's and even number of b's.

So both S_2 and S_4 should be marked as final to accept the given language.

51. The given grammar generates the following regular expression $(0011)^+$. So L is regular but not $(0+1)^*$

52. The first choice has $\in$. In the second choice the rule $S \rightarrow AcB$ is not compliant to CNF

53. The given CFG defines a palindrome

Consider the following string 0001111111000

$S \Rightarrow 0S0 \Rightarrow 00S00 \Rightarrow 000S000 \Rightarrow 0001S1000 \Rightarrow 00011S11000 \Rightarrow 000111S111000 \Rightarrow 0001111111000$

55. Block size is 1KB and block address is 32 bits

$$\text{Number of entries} = \frac{2^{10} \times 2^3 \text{bits}}{2^5 \text{bits}} = 256$$

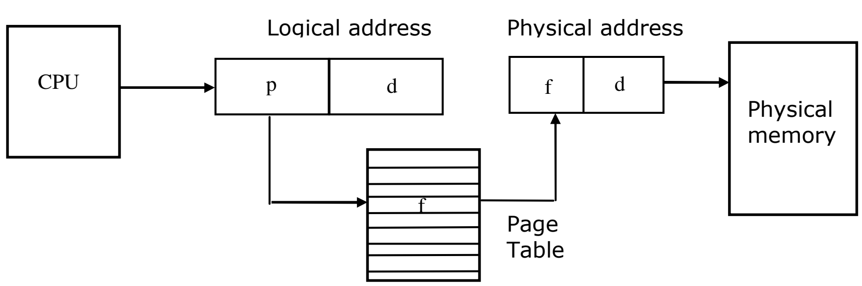
Triple indirection will Point $256 \times 256 \times 256 \times 1\text{KB} = 2^{34}$Byte

56. This is the order in which the jobs are processed.

J_1	J_1	J_1	J_3	J_4	J_4	J_2	J_2	J_2

So the average waiting time is = $\frac{0+1+1+5}{4} = 1.75$

57. Both p+d = 32 and f+d=27 as 128 MB of physical memory is there. In this'd' takes 11 bits as page size is 2 KB. So p becomes 21 and f becomes 16. With these parameters the page table size becomes $2^{21} \times 2^4$ bits $= 4MB$ (approximately ignoring other information bits that are present like dirty bit)



58. After first Four References

4	8	2	3

When '1' comes '4' becomes the LRU page. So we replace it with (1).

1	8	2	3

⇓

1	6	2	3

⇓

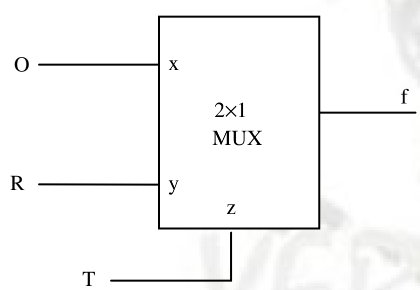
1	6	2	7

59. Truth table

w	x	y	z	A
0	0	0	0	0
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

For all other values it is '0'. Solve it using Karnaugh map to get a Boolean equation. Then the number of min-terms and the number of complements gives the required number of gates.

60.



61. The output of D-flip-flop should be '1' if P = 0 and X = 1. So the flip flop should be reset, i.e. P = 0 and TRST = 0 to get a '0' on X.

63. The output of OR gate comes out as $\sum(0,1,3,5,6,7)$. The negation of this gives us $\sum(2,4)$. The output of AND gate should be the common min-terms between $\sum(2,4)$ and $\sum(1,4,5)$ which is = $\sum(4)$.

64. If J and K are tied to '1' then the output is always complemented. When CP is triggered A_1 becomes 1. This triggers the second flipflop as CP is going from 0 to 1. The other flip flops are also triggered in the similar fashion. So the output is 1001.

66. If a relation R is decompose in to R_1 and R_2

$(R_1 \cap R_2) \to R_1 - R_2, (R_1 \cap R_2) \to R_2 - R_1$ *to make the decomposition lossless*

67. Consider the set

$A \to B...1, B \to C...2, C \to D...3$
from 1 and 2 $A \to C....4$
from 1 and 4 $A \to BC$
from 2 and 4 $AB \to C$
from 3 and 4 $A \to D.....5$
from 3 and 5 $AC \to D$

68. Since name is unique so name is primary key; Also zip is determining city and state so non key to non key dependency exists hear and given reation is not in 3NF so highest normal form satisfied by given relation is 2NF.

69. As the roll number is a primary key it is different for all entries in both student and enroll databases, so the join is always the minimum of both the tables entries.

70. Here course number is not a primary key. So there could be a chance that course number is same for all the entries in both the tables which gives us the maximum value of 120. If it is distinct as in (a) then it gives the minimum of 30.

76. Two stacks are needed.

77. Enqueue operation is done by the PUSH operation on stack 1. Dequeue is done by popping all elements on to STACK 2 and then returning the top element. All the elements have to be moved back to STACK1 to retain correct state. We can think of reducing the number of element movements. But it can never be less than n. so enqueue takes O(1) time and dequeue takes O(n) time

78. If we write P(T) then x or y should have p(s). in any case there is a chance that process p acquires s and waits for T. And, process Q acquires T and waits for S in which case the deadlock occurs. Similar is the case with p(U). If it is P(S) then only one of P or Q can acquire it and proceed further.

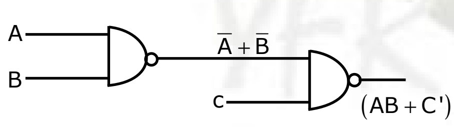
79. Can be P(T), P(U) or P(U), P(T)

81. All bits are shifted left through carry except the last bit. So last bit should also be shifted through carry to retain it's initial value.

83. $S(4) = S(3) + S(2) + 1$

$= [S(2) + S(1) + 1] + [S(1) + S(0) + 1] + 1$
$= [[S(1) + S(0) + 1] + S(1) + 1] + [S(1) + S(0) + 1] + 1$
$= [[2 + 1 + 1] + 2 + 1] + [2 + 1 + 1] + 1 = 12$

84.



85.

C \ AB	00	01	11	10
0	1	1	1	1
1	0	0	1	0
	C′	C′	1	C′