1. Consider the following C program segment.

```
#include <stdio.h>

int main()
{
      char s1[7] = "1234", *p;
      p = s1 + 2;
      *p = '0';
      printf("%s", s1);
}
```

What will be printed by the program?
A.12
B.120400
C.1204
D.1034
Answer ||| C
Solution |||
Char s1 [7] = "1234", *p;
p = s1 + 2; // p holds address of character 3
*p = '0'; // memory at s1 + 3 now becomes 0
Print ("%s", s1); // All characters are printed
So the output is option C 1204
C is an imperative (procedural) language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. C was therefore useful for many applications that had formerly been coded in assembly language, as in system programming.
Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers.

2. Suppose *U* is the power set of the set *S* = {1, 2, 3, 4, 5, and 6}. For any *T*∈*U*, let |*T*| denote the number of elements in *T* and *T'* denote the complement of *T*. For any *T, R* ∈ *U let T* \ *R* be the set of all elements in T which are not in *R*. Which one of the following is true?

A.$\forall X \in U\ (|X|=|X'|)$
B.$\exists X \in U \exists Y \in U\ (|X|=5, |Y|=5 and X \cap Y=\emptyset)$
C.$\forall X \in U \forall Y \in U\ (|X|=2, |Y|=3 and X \setminus Y=\emptyset)$
D.$\forall X \in U \forall Y \in U\ (X \setminus Y=Y' \setminus X')$
Answer ||| D
Solution |||
D is true; it can be seen by drawing Venn diagram.
A is false, Take an example like X = {1, 2, 3, 4},
X' = {5, 6}, |X| is not same as |X'|.
B is false, as any two subsets of size 5 of U would definitely have some common elements.
C is false, Take an example like X = {1, 2}
Y = {3, 4, 5}, X\Y = {1, 2}.
The true value is $\forall X \in U \forall Y \in U\ (X \setminus Y=Y' \setminus X')$

3. Consider the relation X (P, Q, R, S, T, and U) with the following set of functional dependencies
F= {
{P, R}→{S, T},
{P, S, U}→{Q, R}
}
Which of the following is the trivial functional dependency in F^+ where F^+ is closure of F?
A. {P, R}→{S, T}
B. {P, R}→{R, T}
C. {P, S}→{S}
D. {P, S, U} → {Q}
Answer ||| C
Solution |||
A functional dependency X -> Y is trivial if Y is a subset of X.
In relational database theory, a functional dependency is a constraint between two sets of attributes in a relation from a database. In other words, functional dependency is a constraint that describes the relationship between attributes in a relation. The result is {P, S} →{S}

4.The maximum number of processes that can be in *Ready* state for a computer system with *n* CPUs is
A. n
B. n^2
C. 2^n
D. Independent of *n*
Answer ||| D
Solution |||
The size of ready queue doesn't depend on number of processes. A single processor system may have many processes waiting in ready queue.
One of the common misconceptions with CPU Ready is that with a large amount of available pCPU GHz on ESXi hosts the infrastructure should operate with relatively low levels of CPU Ready. Active usage does not cover how many cores are being used by virtual guests at any point in time, preventing other virtual guests from being scheduled by the VM kernel scheduler. An important fact to note is that a virtual machines CPU Usage and CPU Ready values are not directly related to each other. A virtual guest can very easily have extremely high CPU utilization but low CPU Ready values in an environment with low consolidation ratios or vice-versa. The CPU is Independent of *n*

5. Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?
A. SLR, LALR
B. Canonical LR, LALR
C. SLR, canonical LR
D. LALR, canonical LR
Answer ||| C
Solution |||
SLR parser is a type of LR parser with small parse tables and a relatively simple parser generator algorithm. Canonical LR parser or LR (1) parser is an LR (k) parser for k=1, i.e. with a single look ahead terminal. It can handle all deterministic context-free languages. LALR parser or Look-Ahead LR parser is a simplified version of a canonical LR parser.

6. Let # be a binary operator defined as
$X \# Y=X'+Y'$ where X and Y are Boolean variables.
Consider the following two statements.

(S1)(P#Q)#R=P# (Q#R)
(S2)Q#R=R#Q
Which of the following is / are true for the Boolean variables P, Q and R?
A. Only S1 is true
B. Only S2 is true
C. Both S1 and S2 are true
D. Neither S1 nor S2 are true
Answer ||| B
Solution |||
S2 is true, as X' + Y' = Y' + X'
S1 is false.
Let P = 1, Q = 1, R = 0, we get different results
(P # Q) # R = (P' + Q')' + R' = (0 + 0)' + 1 = 1 + 1 = 1
P # (Q # R) = P' + (Q' + R')' = 0 + (0 + 1)' = 0 + 0 = 0

7. Consider a software project with the following information domain characteristics for calculation of function point metric.
Number of external inputs (I) = 30
Number of external outputs (O) = 60
Number of external inquiries (E) =23
Number of files (F) =08
Number of external interfaces (N) =02
It is given that the complexity weighting factors for I, O, E, F and N are 4, 5, 4, 10 and 7, respectively. It is also given that, out of fourteen value adjustment factors that influence the development effort, four factors are not applicable, each of the other four factors has value 3, and each of the remaining factors has value 4. The computed value of function point metric is _____.
A. 612.06
B. 199
C. 345
D. 098
Answer ||| A
Solution |||
Function point metrics provide a standardized method for measuring the various functions of a software application
The value of function point metric = UPF * VAF
Here,
UPF: Unadjusted Function Point (UFP) count
VAF: Value Adjustment Factor
UPF = 4*30 + 60*5 + 23*4 + 8*10 + 7*2 = 606
VAF = (TDI * 0.01) + 0.65
Here TDI is Total Degree of Influence
TDI = 3*4 + 0*4 + 4*6 = 36
VAF = (TDI * 0.01) + 0.65
= 36*0.01 + 0.65
= 0.36 + 0.65
= 1.01
FP = UPF * VAF
= 1.01 * 606
= 612.06

8. In a web server, ten Web Pages are stored with the URLs of the form ://.yourname.com/*var*.html; where, vary is a different number from 1 to 10 for each Webpage. Suppose, the client stores the Web page with vary = 1 (say W1) in local machine, edits and then tests. Rest of the Web Pages remains on the web server. W1 contains several relative URLs of the form "var.html" referring to the other Web Pages. Which one of the following statements needs to be added in W1, so that all the relative URLs in W1 refer to the appropriate Web Pages on the web server?
A. <a href:":///.yourname.com/",
href:"...var.html">
B. <base href:":///.yourname.com/">
C. <a href:":///.yourname.com/">
D. <base href:":///.yourname.com/",
range:"...var.html">
Answer ||| D
Solution |||
<head>
<base href="://..com/images/"
target="_blank">
</head>
<body>
<img src="stickman.gif" width="24" height="39"
alt="Stickman">
</body>
So option D <base href:":///.yourname.com/", range:"...var.html"> is the correct one.

9. Consider the following statements.

I. TCP connections are full duplex
II. TCP has no option for selective acknowledgement
III. TCP connections are message streams
A. Only I is correct
B. Only I and III are correct
C. Only II and III are correct
D. All of I, II and III are correct
Answer ||| A
Solution |||
TCP connections are byte streams. In TCP, selective acknowledgements are possible.
So the term "TCP connection" refers to the application of the TCP protocol. The protocol is state full, naturally, and typically proceeds in a SYN-ACK-data-FIN sequence or SYN/RST in case of a rejected transmission; both peers maintain a status of the connection (handshake, established, closing, closed.) So Only I is correct. TCP connections are full duplex

10. Consider the equality $\sum_{i=0}^{n} i^3 = X$ and the following choices for X
I. $\theta(n^4)$
II. $\theta(n^5)$
III. $0(n^5)$
IV. $\Omega(n^3)$
The equality above remains correct if Xis replaced by
A. Only I
B. Only II
C. I or III or IV but not II
D. II or III or IV but not I
Answer ||| C
Solution |||
X = Sum of the cubes of {1, 2, 3, .. n| $X = n^2 (n+1)^2 / 4$.
By examining the first five sums a remarkable discovery is suggested:
$1^3 = 1$
$1^3+2^3=9$
$1^3+2^3+3^3=36$
$1^3+2^3+3^3+4^3=100$
$1^3+2^3+3^3+4^3+5^3=225$

It seems that the sum is always square, but what is even more remarkable is that the sum of the first n cubes, $1^3+2^3+...+n^3 = (n(n+1)/2)^2$, which is the square of the n^{th} triangle number.
For example, $1^3+2^3+...+10^3=(10\times11/2)^2=55^2 = 3025$
Using a similar method used to prove that formula for the Sum of Squares, we shall prove this result deductively; it is hoped that it will offer some insight into how further the series of powers may be found. So option C is correct answer that is I or III or IV but not II is correct.

11. Consider a binary tree T that has 200 leaf nodes. Then, the numbers of nodes in T that have exactly two children are _____.
A. 199
B. 200
C. 450
D. 560
Answer ||| A
Solution |||
This can be proved using Handshaking Lemma.
A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. A recursive definition using just set theory notions is that a (non-empty) binary tree is a triple (L, S, R), where L and R are binary trees or the empty set and S is a singleton set. Some authors allow the binary tree to be the empty set as well.
From a graph theory perspective, binary (and K-aryl) trees as defined here are actually arborescence. A binary tree may thus be also called a bifurcating arborescence, a term which actually appears in some very old programming books, before the modern computer science terminology prevailed. It is also possible to interpret a binary tree as an undirected, rather than a directed graph, in which case a binary tree is an ordered, rooted tree. Some authors use rooted binary tree instead of binary tree to emphasize the fact that the tree is rooted, but as defined above, a binary tree is always rooted. A binary tree is a special case of an ordered K-array tree, where k is 2. Thus the numbers of nodes in T that have exactly two children are 199

12. Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is_____.
A. 80
B. 50
C. 40
D. 30
Answer ||| A
Solution |||
Load factor = (no. of elements) / (no. of table slots) = 2000/25 = 80
A hash table (hash map) is a data structure used to implement an associative array, a structure that can map keys to values. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.
Ideally, the hash function will assign each key to a unique bucket, but it is possible that two keys will generate an identical hash causing both keys to point to the same bucket. Instead, most hash table designs assume that hash collisions—different keys that are assigned by the hash function to the same bucket—will occur and must be accommodated in some way.
In a well-dimensioned hash table, the average cost (number of instructions) for each lookup is independent of the number of elements stored in the table. Many hash table designs also allow arbitrary insertions and deletions of key-value pairs, at (amortized) constant average cost per operation.

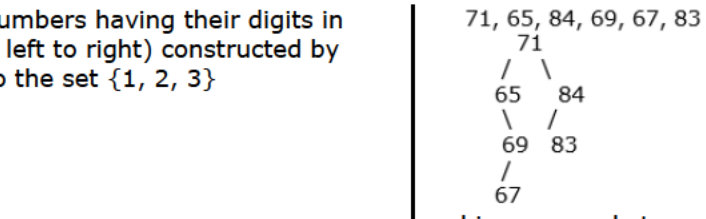
13. In the given matrix $\begin{bmatrix} 1 & -1 & 2 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}$, one of the Eigen values is 1. The Eigen vectors corresponding to the Eigen value1are
A. $\{\alpha(4,2,1)|\alpha \neq 0, \alpha \in \mathbb{R}\}$
B. $\{\alpha(-4,2,1)|\alpha \neq 0, \alpha \in \mathbb{R}\}$
C. $\{\alpha(\sqrt{2},0,1)|\alpha \neq 0, \alpha \in \mathbb{R}\}$
D. $\{\alpha(-\sqrt{2},0,1)|\alpha \neq 0, \alpha \in \mathbb{R}\}$
Answer ||| B
Solution |||
Let z represents the Eigen values.
And let the given matrix be a (square matrix of order 3 x3)
The characteristic equation for this is:
AX = zX (X is the required eigenvector)
AX - zX = 0
[A - Z I] [X] = 0 (I is an identity matrix of order 3)
Put z = 1 (because one of the Eigen value is 1)
[A - 1 I] [X] = 0
The resultant matrix is:
[0 -1 2] [X1] [0]
| 0] |x2] =|0|
[1 2 0] |x3] [0]
Multiplying the above matrices and getting the equations as:
-X2 + 2x3 = 0 ---------- (1)
X1 + 2x2 = 0---------- (2)
Now let x1 = k, then x2 and x3 will be -k/2 and -k/4 Respectively.
Hence eigenvector X = {(k, -k/2, -k/4)} where k! = 0
Put k = -4c (c is also a constant, not equal to zero),
We get X = {(-4c, 2c, 1c)}, i.e. {c (-4, 2, 1)}
Hence option B is correct answer.

14. The value of $\lim_{x\to\infty}(1+x^2)^{e^{-x}}$ is
A.0
B.1/2
C.1
D.∞
Answer ||| A
Solution |||
This can be solved using L'Hôpital's rule that uses derivatives to help evaluate limits involving indeterminate forms. Since [Tex] \lim_{x \to c}f(x)=\lim_{x \to c}g(x)=∞ , and \lim_{x\to c}\frac{f'(x)}{g'(x)} exists[/Tex] We get [Tex] \lim_{x\to c}\frac{f(x)}{g(x)} = \lim_{x\to c}\frac{f'(x)}{g'(x)}. [/Tex] [Tex] \lim_{x\to \infty}\frac{1 + x^2}{e^x} = \lim_{x\to \infty}\frac{2x}{e^x} = \lim_{x\to \infty}\frac{2}{e^x} = 0 [/Tex]. The result is 0

15. The number of 4 digit numbers having their digits in non-decreasing order (from left to right) constructed by using the digits belonging to the set {1, 2, 3} is_____.
A. 5
B. 10
C. 12
D. 15
Answer ||| D
Solution |||
The number of 4 digit numbers having their digits in non-decreasing order (from left to right) constructed by using the digits belonging to the set {1,2,3} is
{1, 1, 1, 1} {1, 1, 1, 2} {1, 1, 1, 3} {1, 1, 2, 2} {1, 1, 2, 3}
{1, 1, 3, 3} {1, 2, 2, 2} {1, 2, 2, 3} {1, 2, 3, 3} {1, 3, 3, 3}
{2, 2, 2, 2} {2, 2, 2, 3} {2, 2, 3, 3} {2, 3, 3, 3} {3, 3, 3, 3}
The number of 4 digit numbers having their digits in non-decreasing order is 15.

16. In a room there are only two types of people, namely Type 1 and Type 2. Type 1 people always tell the truth and Type 2 people always lie. You give affair coin to a person in that room, without knowing which type he is from and tell him to toss it and hide their sult from you till you ask for it. Upon asking, the person replies the following
"There sult of the toss is head if and only if I am telling the truth."
Which of the following options is correct?
A. There sult is head
B. The result is tail
C. If the person is of Type 2, then there sult is tail
D. If the person is of Type 1, then there sult is tail
Answer ||| A
Solution |||
"The result of the toss is head if and only
if I am telling the truth."
If the person is of Type 1 who always tell truth, then result must be head.
If the person is of Type 2 who always tell lie, then result must be head.
Negation of a sentence of the form "X is true if and only if Y is true" is
"Either X is true and Y is false, or X is false and Y is true."
Which means "Either toss is head and I am not telling truth, or toss is tail?
and I am telling truth".
Since the person always lies, it is "Either toss is head or I am not telling truth"
Thus, option A is correct.

17. While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is
A.65
B.67
C.69
D.83
Answer ||| B
Solution |||



binary search trees (BST), sometimes called ordered or sorted binary trees, are a particular type of containers: data structures that store "items" (such as numbers, names etc.) in memory. They allow fast lookup, addition and removal of items, and can be used to implement either dynamic sets of items, or lookup tables that allow finding an item by its key (e.g., finding the phone number of a person by name).
Binary search trees keep their keys in sorted order, so that lookup and other operations can use the principle of binary search: when looking for a key in a tree (or a place to insert a new key), they traverse the tree from root to leaf, making comparisons to keys stored in the nodes of the tree and deciding, based on the comparison, to continue searching in the left or right sub trees. On average, this means that each comparison allows the operations to skip about half of the tree, so that each lookup, insertion or deletion takes time proportional to the logarithm of the number of items stored in the tree. This is much better than the linear time required finding items by key in an (unsorted) array, but slower than the corresponding operations on hash tables.
So, Option B is correct.

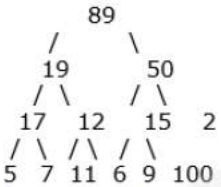
18. The result evaluating the postfix expression 105 + 606 / * 8 – is
A. 284
B. 213
C. 142
D. 71
Answer ||| C
Solution |||
The result evaluating the postfix expression 105 + 606 / * 8 – is 142. The Postfix notation is used to represent algebraic expressions. The expressions written in postfix form are evaluated faster compared to infix notation as parenthesis are not required in postfix. We have discussed infix to postfix conversion. In this post, evaluation of postfix expressions is discussed.
Following is algorithm for evaluation postfix expressions.
1) Create a stack to store operands (or values).
2) Scan the given expression and do following for every scanned element.
…..a) If the element is a number, push it into the stack
…..b) If the element is an operator, pop operands for the operator from stack. Evaluate the operator and push the result back to the stack
3) When the expression is ended, the number in the stack is the final answer

19. Consider the following relation
Cinema (*theater*, *address*, *capacity*)
Which of the following options will be needed at the end of the SQL query SELECT P1.*address*
FROM Cinema P1
such that it always finds the addresses of the alters with maximum capacity?

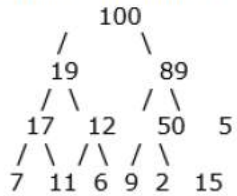
A.WHEREP1.*capacity*>=All (select P2.*capacity* from Cinema P2)
B. WHEREP1.*capacity*>=Any (select P2.*capacity* from Cinema P2)
C.WHEREP1.*capacity*> All (select max (P2.*capacity*) from Cinema P2)
D.WHEREP1.*capacity*> Any (select max (P2.*capacity*) from Cinema P2)
Answer ||| A
Solution |||
Option A is correct. When the ALL condition is followed by a list, the optimizer expands the initial condition to all elements of the list and strings them together with AND operators. When the **ANY** condition is followed by a list, the optimizer expands the initial condition to all elements of the list and strings them together with OR operators.
Logical Operators

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A || B) is true.

20. Consider the following array of elements.
⟨89,19,50,17,12,15,2,5,7,11,6,9,100⟩
The minimum number of interchanges needed to convert it into a max-heap is
A.4
B.5
C.2
D.3
Answer ||| D
Solution |||
⟨89, 19, 50, 17, 12, 15, 2, 5, 7, 11, 6, 9, 100⟩



Minimum number of swaps required to convert above tree
to Max heap is 3. Below are 3 swap operations.
Swap 100 with 15
Swap 100 with 50
Swap 100 with 89



The minimum number of interchanges needed to convert it into a max-heap is 3

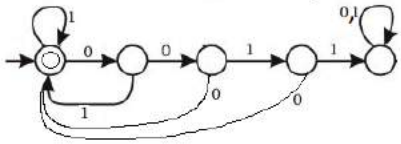
21. Two processes X and Y need to access a critical section. Consider the following synchronization construct used by both the processes

```
Process X                                   Process Y

/* other code for process X */              /* other code for process Y */
while(true)                                 while(true)
{                                           {
   varP = true;                                varQ = true;
   while(varQ == true)                         while(varP == true)
         {                                           {
       /* Critical  Section  */                    /* Critical  Section  */
             varP =  false;                              varQ =  false;
          }                                           }
}                                           }
/* other code for process X */              /* other code for process Y */
```

Here, var P and var Q are shared variables and both are initialized to false. Which one of the following statements is true?
A. The proposed solution prevents deadlock but fails to guarantee mutual exclusion
B. The proposedsolutionguaranteesmutualexclusionbutfailstopreventdeadlock
C. The proposed solution guarantees mutual exclusion and prevents deadlock
D. The proposed solution fails to prevent deadlock and fails to guarantee mutual exclusion
Answer ||| A
Solution |||
The proposed solution prevents deadlock but fails to guarantee mutual exclusion . When both processes try to enter critical section simultaneously, both are allowed to do so since both shared variables varP and varQ are true.So, clearly there is NO mutual exclusion. Also, deadlock is prevented because mutual exclusion is one of the four conditions to be satisfied for deadlock to happen. Hence, answer is A
In concurrent programming, a critical section is a part of a multi-process program that may not be concurrently executed by more than one of the program's processes. In other words, it is a piece of a program that requires mutual exclusion of access.

22. Let L be the language represented by the regular expression $\Sigma^*0011\Sigma^*$where = {0, 1}. What is the minimum number of states in a DFA that recognizes $\bar{L}$ (complement of L)?
A.4
B.5
C.6
D.8
Answer ||| B
Solution |||
The minimum number of states in a DFA that recognizes is 5. The given regular expression matches with all strings that contain 0011. The complement should match with all strings except the strings with 0011 as substring. Below is DFA for the complement.



23. Consider a software program that is artificially seeded with 100 faults. While testing this program, 159 faults are detected, out of which 75 faults are from those artificially seeded faults. Assuming that both real and seeded faults are of same nature and have same

distribution , the estimated number of undetected real faults is_____.
A. 28
B. 38
C. 48
D. 58
Answer ||| A
Solution |||
Total faults detected = 159
Real faults detected among all detected faults = 159 – 75 = 84
Since probability distribution is same, total number of real
Faults is (100/75)*84 = 112
Undetected real faults = 112- 84 = 28

24. Consider a machine with a byte addressable main memory of 2^{20} bytes, block size of 16 bytes and a direct mapped cache having 2^{12} cache lines. Let the addresses of two consecutive bytes in main memory be $(E201F)^{16}$ and $(E2020)^{16}$. What are the tag and cache line address (in hex) for main memory address $(E201F)^{16}$?
A. E, 201
B. F, 201
C. E, E20
D. 2, 01F
Answer ||| A
Solution |||
Block Size = 16 bytes
Block Offset = 4
No. of sets or cache lines = 2^{12}
Number of index bits = 12
Size of main memory = 2^{20}
Number of tag bits = 20 - 12 - 4 = 4
Let us consider the hex address E201F
Tag lines = First 4 bits = E (in hex)
Cache lines = Next 12 bits = 201 (In Hex)
line address (in hex) for main memory address $(E201F)^{16}$ is E, 201

25. Consider a CSMA / CD network that transmits data at a rate of 100 Mbps (10^8 bits per second) over a 1 km (kilo meter) cable with n repeaters. If the minimum frame size required for this network is 1250 bytes, what is the signal speed (km /sec) in the cable?
A.8000
B.10000
C.16000
D.20000
Answer ||| D
Solution |||
Data should be transmitted at the rate of 100 Mbps.
Transmission Time >= 2*Propagation Time
=> 1250*8 / (100 * 10^6) <= 2*length/signal speed
=> Signal speed <= (2 * 10^3 * 100 * 10^6) / (1250 * 8)
<= 2 * 10 * (10 ^ 3) km/sec
<= 20000

26. The velocity *v* (in kilometer / minute) of a motor bike which starts from rest, is given at fixed intervals of time *t* (in minutes) as follows:

t	2	4	6	8	10	12	14	16	18	20
v	10	18	25	29	32	20	11	5	2	0

The approximate distance (in kilometers) rounded to two places of decimals covered in 20 minutes using Simpson's 1/3rd rule is_____.
A. 309.33
B. 105.34
C. 59.34
D. 110.45
Answer ||| A
Solution |||
In numerical analysis, Simpson's rule is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the following approximation:

$$\int_a^b f(x)\,dx \approx \tfrac{b-a}{6}\left[f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\right].$$

Simpson's rule also corresponds to the three-point Newton-Cotes quadrature rule.

```
Float Simpsons( float (*f)(float x), float a, float b, int n) {
float h = (b - a) / n;
float x;
float r;
char m = 0;
float s = 0.0;
for (x = a; x <= b; x+=h) {
r = f(x);
if (x == a || x == b) {
s += r;
} else {
m = !m;
s += r * (m+1) * 2.0;
}
}
return s * (h/3.0);
}
```

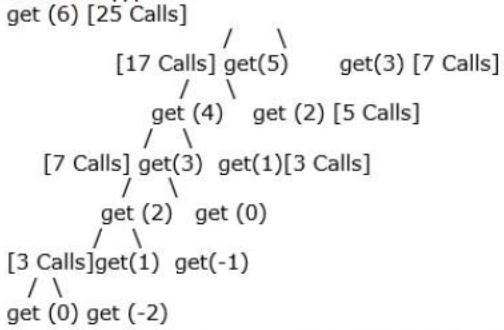
Result using Simpson's 1/3rd rule is 309.33

27. Assume that a merge sort algorithm in the worst case takes 30 seconds for an input of size 64. Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?
A.256
B.512
C.1024
D.2048
Answer ||| B
Solution |||
Time complexity of merge sort is Θ (n Log n)
c * 64Log64 is 30
c * 64 * 6 is 30
c is 5/64
For time 6 minutes
5/64* n Log n = 6*60
n Log n = 72*64 = 512 * 9
n = 512

28. Consider the following recursive C function.

```
void get(int n)
 {
       if (n<1) return;
       get(n-1);
       get(n-3);
       printf("%d", n);
 }
```

If get (6) function is being called in main () then how many times will they get ()
Function be invoked before returning to the main ()?
A.15
B.25
C.35
D.45
Answer ||| B
Solution |||



We can verify the same by running below program. 1 # include int count = 0; void get (int n) { count++; if (n < 1) return; get(n-1); get(n-3); } int main() { get(6); printf("%d ", count); } [/sourcecode] Output: 25

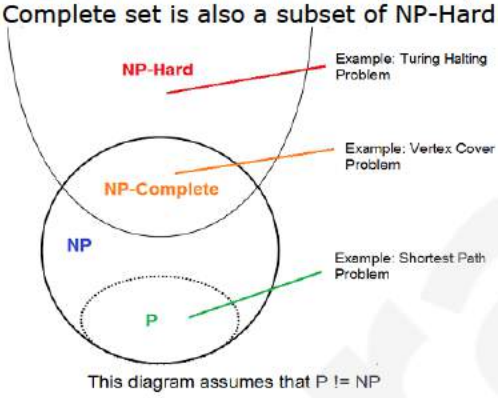
29. Consider a B+ tree in which the search key is 12 bytes long, block size is 1024 bytes, record pointer is 10 bytes long and block pointer is 8 bytes long. The maximum number of keys that can be accommodated in each non-leaf node of the tree is_____.
A. 40
B. 50
C. 60
D. 70
Answer ||| B
Solution |||
Let m be the order of B+ tree
m(8)+(m-1)12 <= 1024
[Note that record pointer is not needed in non-leaf nodes]
m <= 51
Since maximum order is 51, maximum number of keys is 50
A B+ tree is an n-array tree with a variable but often large number of children per node. A B+ tree consists of a root, internal nodes and leaves. The root may be either a leaf or a node with two or more children.
A B+ tree can be viewed as a B-tree in which each node contains only keys (not key-value pairs), and to which an additional level is added at the bottom with linked leaves.
The primary value of a B+ tree is in storing data for efficient retrieval in a block-oriented storage context in particular, file systems. This is primarily because unlike binary search trees, B+ trees have very high fan-out (number of pointers to child nodes in a node, typically on the order of 100 or more), which reduces the number of I/O operations required to find an element in the tree. The maximum number of keys that can be accommodated in each non-leaf node of the tree is 50.

30. Given the function F = P′ + QR, where F is a function in three Boolean variables P, Q and R and P′ =! P, consider the following statements.
(S1) F=∑ (4, 5, 6)
(S2)F=∑ (0, 1, 2, 3, 7)
(S3)F=∏ (4, 5, 6)
(S4)F=∏ (0, 1, 2, 3, 7)
Which of the following is true?
A. (S1)-False, (S2)-True, (S3)-True, (S4)-False
B. (S1)-True, (S2)-False, (S3)-False, (S4)-True
C. (S1)-False, (S2)-False, (S3)-True, (S4)-True
D. (S1)-True, (S2)-True, (S3)-False, (S4)- False
Answer ||| A
Solution |||
Option A (S1)-False, (S2)-True, (S3)-True, (S4)-False is correct. After drawing K map of F= P` +QR, we can find out S2 and S3 are TRUE. Karnaugh maps are used to simplify real-world logic requirements so that they can be implemented using a minimum number of physical logic gates. A sum-of-products expression can always be implemented using AND gates feeding into an OR gate, and a product-of-sums expression leads to OR gates feeding an AND gate. Karnaugh maps can also be used to simplify logic expressions in software design. Boolean conditions, as used for example in conditional statements, can get very complicated, which makes the code difficult to read and to maintain. Once minimized, canonical sum-of-products and product-of-sums expressions can be implemented directly using AND and OR logic operators.
Karnaugh maps are used to facilitate the simplification of Boolean algebra functions. Take the Boolean function described by the following truth table.
Truth table of a function

	A	B	C	D	f(A, B, C, D)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

31. Language L_1 is polynomial time reducible to language L_2. Language L_3 is polynomial time reducible to L_2, which in turn is polynomial time reducible to language L_4. Which of the following is / are true?
I. if $L_4 \in P$, then$L_2 \in P$
II. If $L_1 \in P$ or$L_3 \in P$, then$L2 \in P$
III. $L_1 \in P$, if and onlyif$L_3 \in P$

IV. if $L_4 \in P$, then $L_1 \in P$ and $L_3 \in P$
A. II only
B. III only
C. I and IV only
D. I only
Answer ||| C
Solution |||
Option C I and IV only is correct. P is set of problems that can be solved by a deterministic Turing machine in **P**olynomial time.
NP is set of decision problems that can be solved by a **N**on-deterministic Turing Machine in **P**olynomial time. P is subset of NP (any problem that can be solved by deterministic machine in polynomial time can also be solved by non-deterministic machine in polynomial time).
Informally, NP is set of decision problems which can be solved by a polynomial time via a "Lucky Algorithm", a magical algorithm that always makes a right guess among the given set of choices (Source Ref 1).
NP-complete problems are the hardest problems in NP set. A decision problem L is NP-complete if:
1) L is in NP (Any given solution for NP-complete problems can be verified quickly, but there is no efficient known solution).
2) Every problem in NP is reducible to L in polynomial time (Reduction is defined below).
A problem is NP-Hard if it follows property 2 mentioned above, doesn't need to follow property 1. Therefore, NP-Complete set is also a subset of NP-Hard set.



32. Consider the following C program.

```
#include<stdio.h>
int f1(void);
int f2(void);
int f3(void);
int x = 10;

int main( )
 {
   int x = 1;
   x += f1( ) + f2( ) + f3( ) + f2( );
   printf("%d", x);
   return 0;
 }

int f1() { int x = 25; x++; return x;}
int f2() { static int x = 50; x++; return x;}
int f3() { x *= 10; return x};
```

The output of the program is_____.
A. 230
B. 250
C. 260
D. 270
Answer ||| A
Solution |||
x += f1() + f2() + f3() + f2();
x = x + f1() + f2() + f3() + f2();
f1() returns 26
f2() returns 51
f3() returns 100
second call to f2() returns 52
[Note x is static in f2()]
x = 1 + 26 + 51 + 100 + 52 = 230. The output of the program is 230

33. Consider the following C program.

```
#include<stdio.h>
int main( )
{
   static int a[ ] = {10, 20, 30, 40, 50};
   static int *p[ ] = {a, a+3, a+4, a+1, a+2};
   int **ptr = p;
   ptr++;
   printf("%d%d", ptr-p,**ptr);
}
```

The output of the program is_____.
A. 140
B. 150
C. 160
D. 170
Answer ||| A
Solution |||
The value of prt-p is 1 and value of **ptr is 40
C is an imperative (procedural) language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. C was therefore useful for many applications that had formerly been coded in assembly language, as in system programming.
Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers. The output of the program is 140

34. Which of the following languages are context-free?
$L_1 = \{a^m b^n a^n b^m | m,n \geq 1\}$
$L_2 = \{a^m b^n a^m b^n | m,n \geq 1\}$
$L_3 = \{a^m b^n \mid m=2n+1\}$
A. L_1 and L_2 only
B. L_1 and L_3 only
C. L_2 and L_3 only
D. L_3 only
Answer ||| B
Solution |||
We can build a push down automata for L1 and L3, but cannot build push down automata for L@. Note that a PDA can use a stack. L1 and L3 can be identified using a single stack, but L2 can't be.
A context-free language (CFL) is a language generated by some context-free grammar (CFG). Different CF grammars can generate the same CF language. It is

important to distinguish properties of the language (intrinsic properties) from properties of a particular grammar (extrinsic properties).
The set of all context-free languages is identical to the set of languages accepted by pushdown automata, which makes these languages amenable to parsing. Indeed, given a CFG, there is a direct way to produce a pushdown automaton for the grammar (and corresponding language), though going the other way (producing a grammar given an automaton) is not as direct. Thus, option B is correct.

35. Consider the following policies for preventing deadlock in a system with mutually exclusive resources.
I. Processes should acquire all their resources at the beginning of execution. If any resource is not available, all resources acquired so far are released
II. There sources are numbered uniquely, and processes are allowed to request for resources only in increasing resource numbers
III. There sources are numbered uniquely, and processes are allowed to request for resources only in decreasing resource numbers
IV. There sources are numbered uniquely. A process is allowed to request only for are source with resource number larger than its currently held resources
Which of the above policies can be used for preventing deadlock?
A. Anyone of I and III but not II or IV
B. Anyone of I, III, and IV but not II
C. Anyone of II and III but not I or IV
D. Anyone of I, II, III, and IV
Answer ||| D
Solution |||
If Ist is followed, then hold and wait will never happen. II, III and IV are similar. If any of these is followed, cyclic wait will not be possible.
In concurrent programming, a deadlock is a situation in which two or more competing actions are each waiting for the other to finish, and thus neither ever does. In a transactional database, a deadlock happens when two processes each within its own transaction updates two rows of information but in the opposite order. Option D Anyone of I, II, III, and IV is correct.

36. In the network 200.10.11.144/27, the fourth octet (in decimal) of the last IP address of the network which can be assigned to a host is_____.
A. 158
B. 168
C. 178
D. 188
Answer ||| A
Solution |||
The last octet of network address is 10010000. The first three bits are fixed as 100, the remaining bits can get maximum value as 11111. So the maximum possible IP address is 10011111 which is 159. The question seems to by asking about network address. The maximum possible network address that can be assigned is 200.10.11.158/31 which has last octet as 158
Two versions of the Internet Protocol (IP) are in use: IP Version 4 and IP Version 6. Each version defines an IP address differently. Because of its prevalence, the generic term IP address typically still refers to the addresses defined by IPv4. The gap in version sequence between IPv4 and IPv6 resulted from the assignment of number 5 to the experimental Internet Stream Protocol in 1979, which however was never referred to as IPv5. fourth octet (in decimal) of the last IP address of the network which can be assigned to a host is 158

37. Consider a network connecting two systems located 8000 kilometers apart. The bandwidth of the network is 500×10^6 bits per second. The propagation speed of the media is 4×10^6 meters per second. It is needed to design a Go-Back-*N* sliding window protocol for this network. The average packet size is 10^7 bits. The network is to be used to its full capacity. Assume that processing delays at nodes are negligible. Then, the minimum size in bits of the sequence number field has to be _____.
A. 4
B. 6
C. 8
D. 12
Answer ||| C
Solution |||
Propagation time = (8000 * 1000)/ (4 * 10^6) = 2 seconds
Total round trip propagation time = 4 seconds
Transmission time for one packet = (packet size) / (bandwidth)
= (10^7) / (500 * 10^6) = 0.02 seconds
Total number of packets that can be transferred before an
Acknowledgement comes back = 4 / 0.02 = 200
Maximum possible window size is 200.
In Go-Back-N, maximum sequence number should be one more than
Window size.
So total 201 sequence numbers are needed. 201 different sequences
Numbers can be represented using 8 bits. the minimum size in bits of the sequence number field has to be 4.
Topic ||| Computer Networks||Flow and Error Control Techniques||Flow and Error Control Techniques

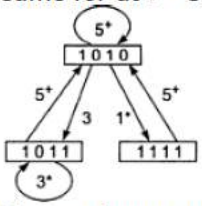
38. Consider the following reservation table for a pipeline having three stages S_1, S_2 and S3.

Time →

	1	2	3	4	5
S_1	X				X
S_2		X		X	
S_3			X		

The minimum average latency (MAL) is_____.
A. 3
B. 2
C. 1
D. 0
Answer ||| A
Solution |||
First we find forbidden latency which is distance between each pair or X in the same row= (2, 4) ... FL indicate another task cannot initiated after this permissible latency = (1, 3) we may initiate another task after this.
Now we can find collision vector using FL...
This collision vector will b known as initial state of the pipeline

collision vector = cycles (4, 3, 2, 1) (one bit for each cycle) and bit at 2 and 4 will b one because of collision= (1010)
now we can construct state diagram using permissible latency.
At 1...we will shift right 1 bit to collision vector (it is initial state) and perform OR operation to result with collision vector =
1010 after one right shift 0101
1010+0101=1111
same for at 3...1011
same for at >=5...1010



Now we have considered these points to find MAL
from the state diagram we can determine optimal latency cycle which result in MAL.
we have to find simple cycle which is a latency cycle in which each state appears only once
some simple cycle are greedy cycle which is one whose edges are all made with minimum latencies from their respective starting state
so here we find latency cycle (3) and (1, 5)
so the answer will b min of (3 which have constant latency or avgas latency of (1, 5) = (1+5)/2=3)
so the answer is 3

39. Consider the following code sequence having five instructions $I_1$to I_5. Each of these instructions has the following format.
OPRi, Rj, Rk
where operation OP is performed on contents of registers Rj and Rk and the result is stored in register Ri.
I_1:ADDR1, R2, R3
I_2:MULR7, R1, R3
I_3:SUBR4, R1, R5
I_4:ADDR3, R2, R4
I_5:MULR7, R8, R9
Consider the following three statements.
S1: There is a nanti-dependence between instructions $I_2$and I_5
S2: There is a nanti-dependence between instructions I_2 and I_4
S3:With in an instruction pipeline a nanti-dependence always creates one or more stalls
Which one of above statements is / are correct?
A. OnlyS1istrue
B. OnlyS2istrue
C. OnlyS1andS3aretrue
D. OnlyS2andS3aretrue
Answer ||| B
Solution |||
The given instructions can be written as below:
I1: R1 = R2 + R3
I2: R7 = R1 * R3
I3: R4 = R1 - R5
I4: R3 = R2 + R4
I5: R7 = R8 * R9
An anti-dependency, also known as write-after-read (WAR), occurs when an instruction requires a value that is later updated.
S1: There is an anti-dependence between instructions I2 and I5
False, I2 and I5 don't form any write after read situation. They both write R7
S2: There is an anti-dependence between instructions I2 and I4
True, I2 reads R3 and I4 writes it.
S3: Within an instruction pipeline an anti-dependence always
creates one or more stalls.
Anti-dependency can be removed by renaming variables. See following example.
1) B = 3
2) A = B + 1
3) B = 7
Renaming of variables could remove the dependency.
1) B = 3
N. B2 = B
2) A = B2 + 1
3) B = 7
So, Option B is true.

40. Consider the following two C code segments. Y and X are one and two dimensional arrays of size n and n × n respectively, where 2≤n≤10. Assume that in both code segments, elements of Y are initialized to 0 and each element X[i][j] of array X is initialized to i + j. Further assume that when stored in main memory all elements of X are in same main memory page frame.

Code segment 1:
```
//initialize elements of Y to 0
//initialize elements X[i][j] of X to i+j

   for(i = 0; i < n; i++)
      Y[i] += X[0][i];
```

Code Segment 2:
```
//initialize elements of Y to 0
//initialize elements X[i][j] of X to i+j

    for(i = 0; i < n; i++)
      Y[i] += X[i][0];
```

Which of the following statements is / are correct?
S1: Final contents of array Y will be same in both code segments
S2: Elements of array X accessed inside the for loop shown in code segment 1 are contiguous in main memory
S3: Elements of array X accessed inside the for loop shown in code segment 2 are contiguous in main memory
A. Only S2 is correct
B. Only S3 is correct
C. Only S1 and S2 are correct
D. Only S1 and S3 are correct
Answer ||| C
Solution |||
In C, 2D arrays are stored in row major order. Therefore, S2 is correct, but S3 is not correct. hence option C is correct.
An array keeps track of multiple pieces of information in linear order, a one-dimensional list. However, the data associated with certain systems (a digital image, a board game, etc.) lives in two dimensions. To visualize this data, we need a multi-dimensional data structure, that is,

a multi-dimensional array. A two-dimensional array is really nothing more than an array of arrays (a three-dimensional array is an array of arrays of arrays). Think of your dinner. You could have a one-dimensional list of everything you eat: (lettuce, tomatoes, steak, mashed potatoes, cake, ice cream) Or you could have a two-dimensional list of three courses, each containing two things you eat: (lettuce, tomatoes) and (steak, mashed potatoes) and (cake, ice cream).

41. Consider the following partial Schedule S involving two transactions T1 and T2. Only the read and the write operations have been shown. The read operation on data item P is denoted by read (P) and the write operation on data item P is denoted by write (P).

Time instance	Transaction-id	
	T1	*T2*
1	*read(A)*	
2	*write(A)*	
3		*read(C)*
4		*write(C)*
5		*read(B)*
6		*write(B)*
7		*read(A)*
8		*commit*
9	*read(B)*	

Schedule *S*

Suppose that the transaction *T1* fails immediately after time instance 9. Which one of the following statements is correct?
A. *T2* must be aborted and then both *T1* and *T2* must be re-started to ensure transaction atomicity
B. Schedule *S* is non-recoverable and cannot ensure transaction atomicity
C. Only *T2* must be aborted and then re-started to ensure transaction atomicity
D. Schedule *S* is recoverable and can ensure atomicity and nothing else needs to bed one
Answer ||| B
Solution |||
T2 reads value of 'A' which is written by T1 and T2 is committed before T1
A transaction symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database. Transactions in a database environment have two main purposes:
To provide reliable units of work that allow correct recovery from failures and keep a database consistent even in cases of system failure, when execution stops (completely or partially) and many operations upon a database remain uncompleted, with unclear status.
To provide isolation between programs accessing a database concurrently. If this isolation is not provided, the programs' outcomes are possibly erroneous.
A database transaction, by definition, must be atomic, consistent, isolated and durable. Database practitioners often refer to these properties of database transactions using the acronym ACID.
Transactions provide an "all-or-nothing" proposition, stating that each work-unit performed in a database must either complete in its entirety or have no effect whatsoever. Further, the system must isolate each transaction from other transactions, results must conform to existing constraints in the database, and transactions that complete successfully must get written to durable storage.
So Option B Schedule *S* is non-recoverable and cannot ensure transaction atomicity
is correct.

42. If the following system has non-trivial solution,
$px+ qy+ rz= 0$
$qx+ ry+ pz= 0$
$rx+ py+ qz= 0,$
then which one of the following options is TRUE?
A. p- q+ r= 0orp= q= -r
B. p+ q- r= 0orp= -q= r
C. p+ q+ r= 0orp= q= r
D. p- q+ r= 0orp= -q= -r
Answer ||| C
Solution |||
|A| should be 0 for non-trivial solution.
A solution for example that is not trivial. Often, solutions or examples involving the number zero are considered trivial. Nonzero solutions or examples are considered nontrivial. For example, the equation x + 5y = 0 has the trivial solution (0, 0). the adjective trivial is frequently used for objects (for example, groups or topological spaces) that have a very simple structure. The noun triviality usually refers to a simple technical aspect of some proof or definition. The origin of the term in mathematical language comes from the medieval tritium curriculum. The antonym nontrivial is commonly used by engineers and mathematicians to indicate a statement or theorem that is not obvious or easy to prove. Option C is true.

43. Consider the following C program:

```
#include<stdio.h>
int main( )
{
    int i, j, k = 0;
    j = 2 * 3 / 4 + 2.0 / 5 + 8 / 5;
    k -= --j;
    for(i = 0; i < 5; i++)
      {
         switch(i + k)
         {
            case 1:
            case 2: printf("\n%d", i+k);
            case 3: printf("\n%d", i+k);
            default: printf("\n%d", i+k);
         }
      }
    return 0;
}
```

The number of times printf statement is executed is_____.
A. 8
B. 9
C. 10
D. 11
Answer ||| C
Solution |||
The following statement makes j = 2
j = 2 * 3 / 4 + 2.0 / 5 + 8 / 5;
The following statement makes k = -1

k -= --j;
There is one important thing to note in switch is, there is no break. Let count of printf statements be 'count'
For i = 0, the value of i+k becomes -1, default block
Is executed, count = 1
For i = 1, the value of i+k becomes 0, default block
Is executed, count = 2
For i = 2, the value of i+k becomes 1, all blocks are
Executed as there is no break, count = 5
For i = 3, the value of i+k becomes 2, three blocks
After case 1: are executed, count = 8
For i = 4, the value of i+k becomes 3, two blocks
Are executed, count = 10, hence result is Option C

44. If for non-zero $x,\ af(x)+bf\left(\frac{1}{x}\right)=\frac{1}{x}-25$ where a≠ b then $\int_1^2 f(x)dx$ is

A. $\frac{1}{a^2-b^2}\left[a(\ln 2-25)+\frac{47b}{2}\right]$

B. $\frac{1}{a^2-b^2}\left[a(2\ln 2-25)-\frac{47b}{2}\right]$

C. $\frac{1}{a^2-b^2}\left[a(2\ln 2-25)+\frac{47b}{2}\right]$

D. $\frac{1}{a^2-b^2}\left[a(\ln 2-25)-\frac{47b}{2}\right]$

Answer ||| A
Solution |||
Option A is correct. In mathematical analysis, and applications in geometry, applied mathematics, engineering, and natural sciences, a function of a real variable is a function whose domain is the real numbers $\mathbb{R}$, more specifically the subset of $\mathbb{R}$ for which the function is defined.
The "output", also called the "value of the function", could be anything: simple examples include a single real number, or a vector of real numbers (the function is "vector valued"). Vector-valued functions of a single real variable occur widely in applied mathematics and physics, particularly in classical mechanics of particles, as well as phase paths of dynamical systems. But we could also have a matrix of real numbers as the output (the function is "matrix valued"), and so on. The "output" could also be other number fields, such as complex numbers, quaternion's, or even more exotic hyper complex numbers.
A real-valued function of a real variable is a function that takes as input a real number, commonly represented by the variable x, for producing another real number, the value of the function, commonly denoted f(x). For simplicity, in this article a real-valued function of a real variable will be simply called a function. To avoid any ambiguity, the other types of functions that may occur will be explicitly specified.

45. Let G be a connected undirected graph of 100 vertices and 300 edges. The weight of a minimum spanning tree of *G* is 500. When the weight of each edge of *G* is increased by five, the weight of a minimum spanning tree becomes_____.
A. 800
B. 995
C. 1000
D. 456
Answer ||| B
Solution |||
Since there are 100 vertices, there must be 99 edges in Minimum Spanning Tree (MST). When weight of every edge is increased by 5, the increment in weight of MST is = 99 * 5 = 495 so new weight of MST is 500 + 495 which is 995.
A graph is connected when there is a path between every pair of vertices. In a connected graph, there are no unreachable vertices. A graph that is not connected is disconnected. A graph with just one vertex is connected.

46. Two hosts are connected via a packet switch with 10^7 bits per second links. Each link has a propagation delay of 20 microseconds. The switch begins forwarding a packet 35 microseconds after it receives the same. If 10000 bits of data are to be transmitted between the two hosts using a packet size of 5000 bits, the time elapsed between the transmission of the first bit of data and the reception of the last bit of the data in micro seconds is _____.
A. 1325
B. 1575
C. 1657
D. 3467
Answer ||| B
Solution |||
Sender host transmits first packet to switch, the transmission time is $5000/10^7$ which is 500 microseconds. After 500 microseconds, the second packet is transmitted. The first packet reaches destination in 500 + 35 + 20 + 20 + 500 = 1075 microseconds. While the first packet is traveling to destination, the second packet starts its journey after 500 microseconds and rest of the time taken by second packet overlaps with first packet. So overall time is 1075 + 500 = 1575

47. For the processes listed in the following table, whichofthefollowingschedulingschemeswillgivethelowestaverageturnaroundtime?

Process	Arrival Time	Processing Time
A	0	3
B	1	6
C	4	4
D	6	2

A. First Come First Serve
B. Non- pre emptive Shortest Job First
C. Shortest Remaining Time
D. Round Robin with Quantum value two
Answer ||| C
Solution |||
Turnaround time is the total time taken between the submission of a program/process/thread/task (Linux) for execution and the return of the complete output to the customer/user. Turnaround Time = Completion Time -

Arrival Time. FCFS = First Come First Serve (A, B, C, D) SJF = Non-preemptive Shortest Job First (A, B, D, C) SRT = Shortest Remaining Time (A(3), B(1), C(4), D(2), B(5)) RR = Round Robin with Quantum value 2 (A(2), B(2), A(1),C(2),B(2),D(2),C(2),B(2)

Pr	Arr.Time	P.Time	FCFS	SJF	SRT	RR
A	0	3	3-0=3	3-0=3	3-0=3	5-0=5
B	1	6	9-1=8	9-1=8	15-1=14	15-1=14
C	4	4	13-4=9	15-4=11	8-4=4	13-4=9
D	6	2	15-6=9	11-6=5	10-6=4	11-6=5
Average			7.25	6.75	6.25	8.25

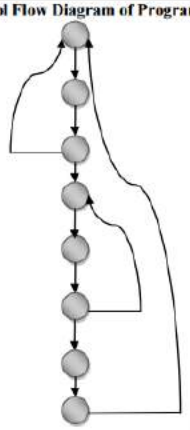
Shortest Remaining Time produces minimum average turn-around time. Option C is correct.

48. Consider three software items: Program-*X*, Control Flow Diagram of Program-*Y* and Control Flow Diagram of Program – *Z* as shown below

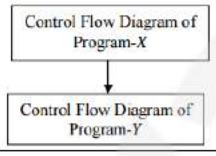
Program-*X*:

```
sumcal(int maxint, int value)
{
    int result=0, i=0;
    if (value <0)
    {
      value = -value;
    }
    while((i<value) AND (result
<= maxint))
    {
      i=i+1;
      result = result + 1;
    }
    if(result <= maxint)
    {
      printf(result);
    }
    else
    {
      printf("large");
    }
    printf("end of program");
}
```

Control Flow Diagram of Program-*Y*:



Control Flow Diagram of Program-*Z*:



The values of McCabe's Cyclamate complexity of Program –*X,* Program -*Y*, and Program -*Z respectively* are
A. 4, 4, 7
B. 3, 4, 7
C .4, 4, 8
D. 4, 3, 8
Answer ||| A
Solution |||
The cyclamate complexity of a structured program[a] is defined With reference to the control flow graph of the program, a directed Graph containing the basic blocks of the program, with an edge Between two basic blocks if control may pass from the first to the Second. The complexity M is then defined as.
M = E – N + 2P,
Where
E = the number of edges of the graph.
N = the number of nodes of the graph.
P = the number of connected components.
For first program X, E = 11, N = 9, P = 1, So M = 11-9+2*1 = 4
For second program Y, E = 10, N = 8, p = 1, So M = 10-8+2*1 = 4
For Third program X, E = 22, N = 17, p = 1, So M = 22-17+2*1 = 7, hence option A.

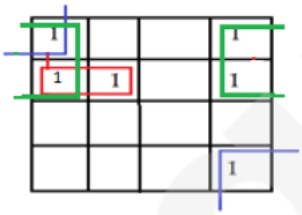
49. Consider the equation $(43)_x = (y3)_8$ where X and Y are unknown. The number of possible solutions is_____
A. 3
B. 4
C. 5
D. 6
Answer ||| C
Solution |||
3 + 4x = 3 + 8y where 0 <= y <= 7
and x >= 5 (because the number represented in base x is 34)
x = 2y and 0 <= y <= 7
The following are possible solutions
y = 3, 4, 5, 6, 7
x = 6, 8, 10, 12, 14, The number of possible solutions is 5.

50. Let *R* be a relation on the set of ordered pairs of positive integers such that $((p,q),(r,s)) \in R$ if and only if $p-s=q-r$. Which one of the following is true about *R*?
A. Both reflexive and symmetric
B. Reflexive but not symmetric
C. Not reflexive but symmetric
D. Neither reflexive nor symmetric
Answer ||| C
Solution |||
((p, q), (r, s)) ∈ R if and only if p–s = q–r
(p, q) is not related to (p, q)
as p-q is not same as q-p.
The relation is symmetric because if p–s = q–r, then s-q = s-p.
Let R be a binary relation on a set A.
R is reflexive if for all x ∈ A, xRx.
R is symmetric if for all x,y ∈ A, if xRy, then yRx.
R is transitive if for all x,y, z ∈ A, if xRy and yRz, then xRz.
R is an equivalence relation if A is nonempty and R is reflexive, symmetric and transitive.
In terms of digraphs, reflexivity is equivalent to having at least a loop on each vertex; symmetry means any arrow from one vertex to another will always be accompanied by another arrow in the opposite direction; and transitivity is the same as saying there must be a direct arrow from one vertex to another if one can walk from that vertex to the other through a list of arrows, travelling always along the direction of the arrows. R is Not reflexive but symmetric

51. Suppose X_i for i=1,2,3 are independent and identically distributed random variables whose probability mass functions are $Pr[X_i=0]= Pr[X_i = 1]=1/2$ for i=1,2,3. Define another random variable $Y = X_1X_2 \oplus X_3$, where $\oplus$ denotes XOR. Then $Pr[Y=0|X_3=0]$ = _____.
A. 0.75
B. 0.65
C. 0.98
D. 1
Answer ||| A
Solution |||

It is given X3 = 0. Y can only be 0 when X1 X2 is 0. X1 X2 become 0 for X1 = 1, X2 = 0, X1 = X2 = 0 and X1 = 0, X = 1 So the probability is = 0.5*0.5*3 = 0.75.
In probability and statistics, a probability distribution assigns a probability to each measurable subset of the possible outcomes of a random experiment, survey, or procedure of statistical inference. Examples are found in experiments whose sample space is non-numerical, where the distribution would be a categorical distribution; experiments whose sample space is encoded by discrete random variables, where the distribution can be specified by a probability mass function; and experiments with sample spaces encoded by continuous random variables, where the distribution can be specified by a probability density function. More complex experiments, such as those involving stochastic processes defined in continuous time, may demand the use of more general probability measures.
In applied probability, a probability distribution can be specified in a number of different ways, often chosen for mathematical convenience:
by supplying a valid probability mass function or probability density function
by supplying a valid cumulative distribution function or survival function
by supplying a valid hazard function
by supplying a valid characteristic function
by supplying a rule for constructing a new random variable from other random variables whose joint probability distribution is known.

52. The total number of prime implicates of the function $f(w, x, y, z)=\Sigma(0,2,4,5,6,10)$is_____.
A. 2
B. 3
C. 4
D. 5
Answer ||| B
Solution |||



Red, Blue and Green together make total three primes implicit.
As we know "A prime implicit of a function is an implicit that cannot be covered by a more general (more reduced - meaning with fewer literals) implicit. (Wikipedia)" we've to see only prime implicates. We can make group of four 1's as in the figure (green group), a group of two 1's in blue and red. As we can't reduce it further so this is minimal representation of problem and hence total number of prime implicit =3. So, answer (B) part.

53. Suppose c = ⟨c[0],...,c[k−1]⟩ is an array of length k, where all the entries are from the set{0,1}. For any positive integers a and n, consider the following pseudo code.
DOSOMETHING(c, a, n)

```
DOSOMETHING(c, a, n)
z ← 1
for i ← 0 to k − 1
    do z ← z^2 mod n
     if c[i] = 1
          then z ← (z × a) mod n
return z
```

If k=4, c=⟨1,0,1,1⟩, a =2 and n =8, then the output of DOSOMETHING(c, a, n)is_____.
A. 0
B. 1
C. 2
D. 3

Answer ||| A
Solution |||
DOSOMETHING (c, a, n)
z ← 1
for i ← 0 to k – 1
do z ← z^2 mod n
if c[i] = 1
then z ← (z × a) mod n
return z
If k = 4, c = ⟨1, 0, 1, 1⟩, a = 2 and n = 8, then the output of DOSOMETHING(c, a, n) is _____.
For i = 0, z = 1 mod 8 = 1, since c[0] = 1, z = 1*2 mod 8 = 2
For i = 1, z = 2*2 mod 8 = 1, since c[1] = 0, z remains 4
For i = 2, z = 16 mod 8 = 0
Once z becomes 0, none of the statements inside DOSOMETHING() can make it non-zero.

54. Let $f(n) = n$ and $g(n) = n^{(1+\sin n)}$, where n is a positive integer. Which of the following statements is / are correct?
I. $f(n)= 0(g(n))$
II. $f(n)=\Omega(g(n))$
A. Only I
B. Only II
C. Both I and II
D. Neither I nor II
Answer ||| D
Solution |||
The value of sine function varies from -1 to 1.
For sin = -1 or any other negative value,
I become false.
For sin = 1 or any other negative value,
II becomes false, Result is Option D.

55. Consider the following grammar G

$$S \rightarrow F \mid H$$
$$F \rightarrow p \mid c$$
$$H \rightarrow d \mid c$$

where S,F, and Hare non-terminal symbols , p, d, and care terminal symbols. Which of the following statement (s) is / are correct?
S1. LL(1) can parse al strings that are generated using grammar G
S2. LR(1) can parse all strings that are generated using grammar G
A. Only S1
B. Only S2

C. Both S1 and S2
D. Neither S1 nor S2
Answer ||| D
Solution |||
The given grammar is ambiguous as there are two possible leftmost derivations for string "c".
First Leftmost Derivation
S → F
F → c
Second Leftmost Derivation
S → H
H → c
An Ambiguous grammar can neither be LL (1) nor LR (1) , so the Result is Neither S1 nor S2

***