1. Which data structure is used for implementing recursion?

A) Stack
B) Queue
C) List
D) Array

Ans: Stack

Recursion is a technique of problem-solving where a function is called again and again on smaller inputs until some base case i.e. smallest input which has a trivial solution arrives and then we start calculating the solution from that point. The stack data structure is used for implementing recursion. The activation records of a function call, which include the address of the calling function, local variables, and arguments, are stored in a LIFO (Last In First Out) order in the stack. This is because the activation records are to be stored in LIFO order.

2. Which data structure is based on the Last In First Out (LIFO) principle?

A) Tree

B) Linked List

C) Stack

D) Queue

Ans: The data structure that is based on the Last In First Out (LIFO) principle is called a **stack**. A stack is a linear data structure that follows the LIFO principle, meaning that the last element added to the stack is the first one to be removed[1]. You can think of a stack as a pile of plates, where you can only add or remove plates from the top of the pile.

3. Which of the following tree data structures is not a balanced binary tree?

A) Splay tree

B) B-tree

C) AVL tree

D) Red-black tree

Answer: B
Explanation: All the tree data structures given in options are balanced, but B-tree can have more than two children.

4. Which of the following is not the type of queue?

A) Priority queue

B) Circular queue

C) Single ended queue

D) Ordinary queue

The correct choice is (C) Single ended queue To explain: Queue always has two ends. So, single ended queue is not the type of queue.

5. Which algorithm is used in the top tree data structure?

A) Backtracking

B) Divide and Conquer

C) Branch

D) Greedy

Right answer is (B) Divide and Conquer The best explanation: Top tree is a type of data structure which is based on unrooted dynamic binary tree and is used to solve path related problems. It allows an algorithm called divide and conquer.

6. Consider the following processes with time slice of 4 milliseconds (I/O requests are ignored):
Process A B C D
Arrival time 0 1 2 3
CPU Cycle 8 4 9 5
The average turnaround time of these processes will be

A) 19.25 milliseconds

B) 18.25 milliseconds

C) 19.5 milliseconds

D) 18.5 milliseconds

Answer : 18.25 milliseconds

Process	AT	BT	CT	TAT
A	0	8	20	20
B	1	4	8	7
C	2	9	26	24
D	3	5	25	22

A	B	C	D	A	C	D	C

0 4 8 12 16 20 24 25 26

Avg TAT = 20 + 7 + 24 + 22 / 4 = 18.25

7. Which scheduling policy is most suitable for a time shared operating system?

A) First Come First Serve

B) Shortest Job First

C) Round Robin

D) Elevator

Ans: According to the web search results, the most suitable scheduling policy for a time shared operating system is **Round Robin (RR)**. This is because RR is a **preemptive** algorithm, which means it can interrupt a running process and switch to another one if needed. This allows the system to respond quickly to user requests and minimize the waiting time of processes. RR also ensures that every process gets a fair share of the CPU time, as it allocates a fixed amount of time (called a time slice) to each process in a circular order.

8. An operating system using banker's algorithm for deadlock avoidance has ten dedicated devices (of same type) and has three processes P1, P2 and P3 with maximum resource requirements of 4, 5 and 8 respectively. There are two states of allocation of devices as follows:

State 1	Processes	P1	P2	P3
	Devices allocated	2	3	4
State 2	Processes	P1	P2	P3
	Devices allocated	0	2	4

Which of the following is correct?

A. State 1 is unsafe and State 2 is safe
B. State 1 is safe and State 2 is unsafe
C. Both State 1 and State 2 are safe
D. Both State 1 and State 2 are unsafe

Exp:

Answer **A**

State 1:

Total number of devices allocated = 2+3+4 =9

No of devices available = 10-9 =1

- The no of devices still needed for P1 = 4-2 =2 >1 //Cannot allocate
- The no of devices still needed for P2 = 5-3 =2 >1 //Cannot allocate
- The no of devices still needed for P2 = 8-4 =4 >1 //Cannot allocate

State 1 is unsafe

state 2

Total number of devices allocated = 0+2+4 =6

No of devices available = 10-6 =4

- The no of devices still needed for P1 = 4-0 =4 =4 //Can allocate
- The no of devices still needed for P2 = 5-2 =3 <4 //Can allocate
- The no of devices still needed for P2 = 8-4 =4 =4 //Can allocate

State 2 is safe

9. In which one of the following page replacement policies, Belady's anomaly may occur?

(A) FIFO
(B) Optimal
(C) LRU
(D) MRU

Answer: (A)

Explanation: Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm.

10. A disk is advertised with a seek time of 3 ms, 512 bytes per sector and 128 sectors per track. The disk rotates at 5, 200 rpm.

i. Determine the average rotational delay for the disk.

ii. Determine the time required to read a 4 Mbyte file. You are to assume that the file occupies sectors on adjacent tracks.

Exp:

5200 rotation --------> 60 seconds

1 rotation --------> $\frac{60}{5200}$ = 11.53 msec

i) Avg rotational delay = $\frac{11.53}{2}$ = 5.76 msec

1 track data = 128 × 512 = 65536 Byte

now,

1 track data wiil be fetched in 1 rotation

65536 byte -------> 11.53 msec

4 Mbyte -------> $11.53 \frac{5365536}{} \times 222$ = 737.92 msec

Number of tracks required to store 4 Mbyte = $\frac{22265536}{}$ = 64

Since data are stored in adjacent track, seek time and rotation time will be considered for every track change

ii) Time to read 4 Mbyte = ((Seek time + Avg rotation time) × 64) + Time to transfer 4 Mbyte

= ((3 msec + 5.76 msec) × 64) + 737.92 msec

= 560.64 msec + 737.92 msec

= 1298.56 msec

11. Following proposition statement is

F1 : P → ~P

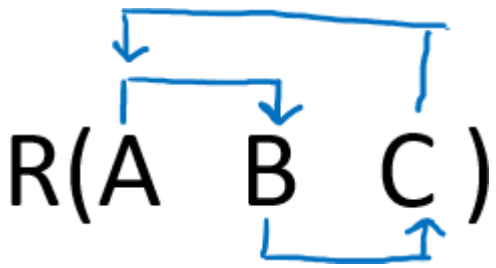
F2 : (P → ~P) V (~P → P)

F1:	**F2:**
P → ~P	**(P → ~P) V (~P → P)**
= ~P V ~P	**= (~P V ~P) V (P V P)**
= P + P	**= (~P) V (P)**
= P	**= ~P V P**
	= 1
Satisfiable	**Valid**

12. Given a relation R(A, B, C) and Functional Dependency set FD = { A → B, B → C, and C → A}, determine given R is in which normal form?

Exp:

Solution: Let us construct an arrow diagram on R using FD to calculate the candidate key.



From the above arrow diagram on R, we can see that all the attributes are determined by all the attributes of the given FD, hence we will check all the attributes (i.e., A, B, and C) for candidate keys

Let us calculate the closure of A

A + = ABC (from the closure method we studied earlier)

Since closure A contains all the attributes of R, hence A is the Candidate key.

Let us calculate the closure of B

B + = BAC (from the closure method we studied earlier)

Since closure B contains all the attributes of R, hence B is the Candidate key.

Let us calculate the closure of C

C + = CAB (from the closure method we studied earlier)

Since closure C contains all the attributes of R, hence C is the Candidate key.

Hence three Candidate keys are: **A B and C**

Since R has 3 attributes: - A B and C, Candidate Keys are A B and C, Therefore, prime attributes (part of candidate key) are A B C while there is no non-prime attribute

Given FD are { A → B, B → C, and C → A } and Super Key / Candidate Key is A B and C

NOTE: To solve such questions, we apply reverse engineering, i.e. 1st check BCNF, if not then 3NF, if not then 2NF, and so on.

a. FD: **A → B** satisfy the definition of BCNF, as A is Super Key, we check other FD for BCNF

b. FD: **B → C** satisfy the definition of BCNF, as B is Super Key, we check other FD for BCNF

c. FD: **C à A** satisfy the definition of BCNF, as C is Super Key

Since there were only three FD's and all FD: { A → B, B → C and C → A } satisfy BCNF, hence the highest normal form is BCNF.

Therefore R(A, B, C) is in BCNF.

13. Let R (A, B, C, D) be a relational schema with the following functional dependencies:

A → B, B → C,

C → D and D → B.

The decomposition of R into

(A, B), (B, C), (B, D)

(A) gives a lossless join, and is dependency preserving
(B) gives a lossless join, but is not dependency preserving

(C) does not give a lossless join, but is dependency preserving
(D) does not give a lossless join and is not dependency preserving

Answer: (A)

Explanation: Background :

- **Lossless-Join Decomposition:**
  Decomposition of R into R1 and R2 is a lossless-join decomposition if at least one of the following functional dependencies are in F+ (Closure of functional dependencies)
- `R1 ∩ R2 → R1`
- `OR`
- `R1 ∩ R2 → R2`
- **Dependency Preserving Decomposition:**
  Decomposition of R into R1 and R2 is a dependency preserving decomposition if closure of functional dependencies after decomposition is same as closure of of FDs before decomposition.
  A simple way is to just check whether we can derive all the original FDs from the FDs present after decomposition.

14. Consider the following schedules involving two transactions.

S1: r1(X);r1(Y);r2(X);r2(Y);w2(Y);w1(X)

S2: r1(X);r2(X);r2(Y);w2(Y);r1(Y);w1(X)

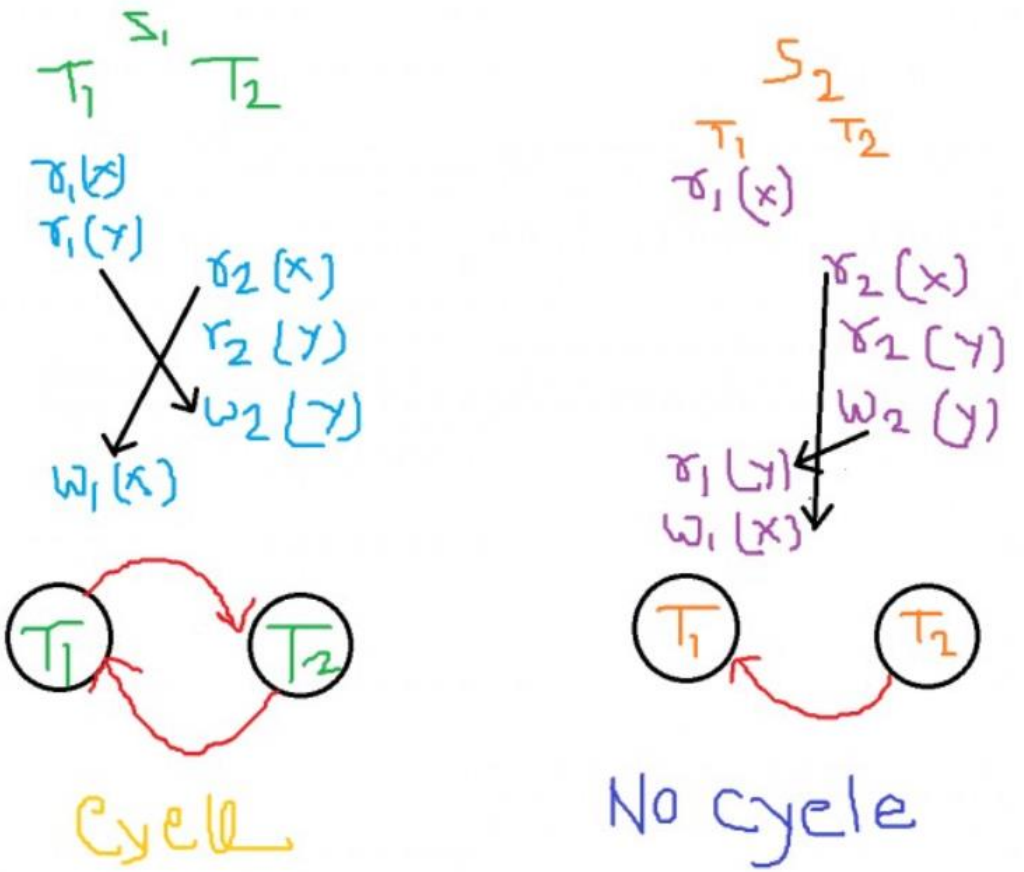
Which one of the following statements is correct with respect to above?

A) Both S1 and S2 are conflict serializable
B) Both S1 and S2 are not conflict serializable
C) S1 is conflict serializable and S2 is not conflict serializable
D) S1 is not conflict serializable and S2 is conflict serializable

F)

Option D. S1 is not conflict serializable and S2 is conflict serializable

Conflict Operations are : RW , WR, WW
Non-Conflict operation : RR
If there is a cycle found then it can't be conflict serializable. In S1 we found a cycle so its not conflict serializable but in S2 there's no cycle found so its conflict serializable.



15. Given the IP address 201.14.78.65 and the subset mask 255.255.255.224. What is the subset address?

A. 201.14.78.32
B. 201.14.78.64
C. 201.14.78.65
D. 201.14.78.224

Answer : 201.14.78.64

At every router basically we have 3 entries in a routing table

1. Network Id / Subnet address

2. Subnet Mask

3 .Interface

So with the help of these we can figure out where to send a packet.

IP address 201.14.78.65 subset mask 255.255.255.224

IP address	11111111.11111111.11111111.11100000
Subnet Mask	11001001.00001110.01001110.01000001
Subnet address	11001001.00001110.01001110.01000000
	201 . 14 . 78 . 64

It is a Bitwise ANDing in above operation.

16. The number of flip-flops required to design a modulo – 272 counter is:

(A)9
(B)8
(C)27
(D)11

Answer: (A)

Explanation:
With n flip – flops we can design upto Modulo 2^n counter. 272 is greater than 256 i.e. 2^8 so we need little bit more; $2^9 = 512$ which is greater than 272. So 9 flip – flops are sufficient to design Modulo 272 counter. Option (B) is correct.

17. The occurrence of degeneracy while solving a transportation problem means that

A. Total supply equals total demand
B. Total supply does not equal total demand
C. The solution so obtained is not feasible
D. None of these

Ans: C

The occurrence of degeneracy while solving a transportation problem means that the solution so obtained is not feasible.

18. In a Linear Programming Problem, suppose there are three basic variables and 2 non-basic variables, then the possible number of basic solutions are

A. 6
B. 8
C. 10
D. 12

Ans: Total number of basic solutions are given by the equation

n!/m! * (n-m)!

where m=3 no of basic variables and n=3+2 =5 total no of variables

hence total soln =5!/3!2!=5x4/2=10

Ans is C

19. When two BCD numbers 0x14 and 0x08 are added what is the binary representation of the resultant number?
(A) 0x22
(B) 0x1c
(C) 0x16
(D) results in overflow

Answer: (A)

Explanation: BCD numbers are binary coded decimal numbers in which each digit of a number is separately represented as a 4-bit binary number.

```
BCD Representation of 0x14 = 0001  0100
BCD Representation of 0x08 = 0000  1000
                             ------------
                  Addition = 0001  1100


In BCD addition, if a digit > 9, add 6 to it.
                         so, 0001  1100
                            +0000  0110
                           --------------
                             0010  0010
```

So in BCD addition, the resultant number is 22.

20. A company needs to develop digital signal processing software for one of its newest inventions. The
software is expected to have 40000 lines of code. The company needs to determine the effort in
person-months needed to develop this software using the basic COCOMO model. The multiplicative

factor for this model is given as 2.8 for this software development on embedded systems, while the
exponentiation factor is given as 1.20. What is the estimation effort in person-months?

(A) 234.25 (B) 932.50 (C) 287.80 (D) 122.40

Explanation:
"Effort = a1 × (KLOC)^a2" -Rajib Mall (Fundamentals of Software Engineering)
KLOC = LOC/1000 = 40000/1000 = 40
Effort = 2.8 × 40^1.2 = 234.2

21. A computer handles several interrupt sources of which the following are relevant for this question.

* Interrupt from CPU temperature sensor (raises interrupt if CPU temperature is too high)
* Interrupt from Mouse (raises interrupt if mouse is moved or button is pressed)
* Interrupt from Keyboard (raises interrupt when a key is pressed or released)
* Interrupt from Hard Disk (raises interrupt when a disk read is completed)

Which one of these will be handled at **HIGHEST** priority?

(A) Interrupt from Hard Disk
(B) Interrupt from Mouse
(C) Interrupt from Keyboard
(D) Interrupt from CPU temperature sensor

Explanation:
"Higher priority interrupt levels are assigned to requests which, if delayed or interrupted,
could have serious consequences. Devices with high speed transfer such as magnetic disks are given
high priority, and slow devices such as keyboard receive low priority."
-Morris Mano (Computer System Architecture)
We know that mouse pointer movements are more frequent than keyboard ticks. So its obvious that its
data transfer rate is higher than keyboard.
Delaying a CPU temperature sensor could have serious consequences, overheat can damage CPU circuitry.
From the above information we can conclude that priorities are-
CPU temperature sensor > Hard Disk > Mouse > Keyboard

22. Which of the following is **NOT** desired in a good Software Requirement Specifications (SRS) document?

(A) Functional Requirements
(B) Non-Functional Requirements
(C) Goals of Implementation
(D) Algorithms for Software Implementation

Explanation:

"An SRS document should clearly document the following aspects of a system: Functional Requirements,
Non-Functional Requirements and Goals of implementation."
-Rajib Mall
(Fundamentals of Software Engineering)

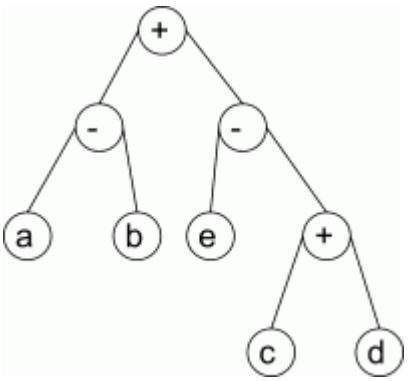
23. What does the following fragment of C program print?

```
char c[] = "DEVA2023";
char *p = c;
printf("%s", p + p[1] - p[3]);
```

(A) DEVA2023 (B) E2023 (C) 2023 (D) 023

Explanation:
p[1]='E'=69 (ASCII of E) and p[3]='A'=65 (ASCII of A)
so p[1]-p[3] = 69-65 = 4
p is address of the first element of array and p+4 is the address of fifth element which is '2'. Hence,
the output of program shows 2023 (Compiled on GCC compiler DEV-Cpp)

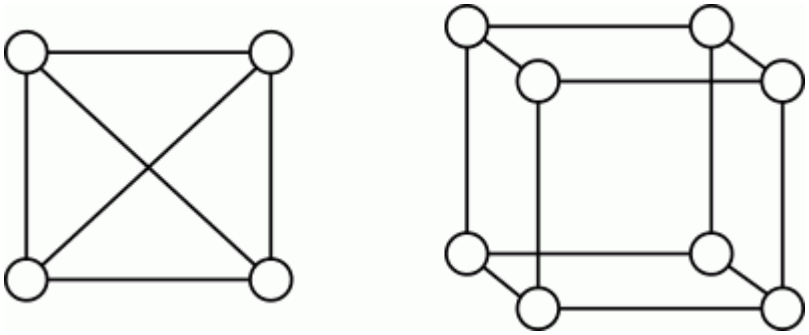
24. Consider evaluating the following expression tree on a machine with load-store architecture in which
memory can be accessed only through load and store instructions. The variables a,b,c,d and e initially
stored in memory. The binary operators used in this expression tree can be evaluate by the machine only
when the operands are in registers. The instructions produce results only in a register. If no
intermediate results can be stored in memory, what is the minimum number of registers needed to
evaluate this expression?



(A) 2 (B) 9 (C) 5 (D) 3

Explanation:
R1←c, R2←d, R2←R1+R2, R1←e, R2←R1-R2
Now to calculate the rest of the expression we must load a and b into the registers but we need the
content of R2 later.
So we must use another Register.
R1←a, R3←b, R1←R1-R3, R1←R1+R2

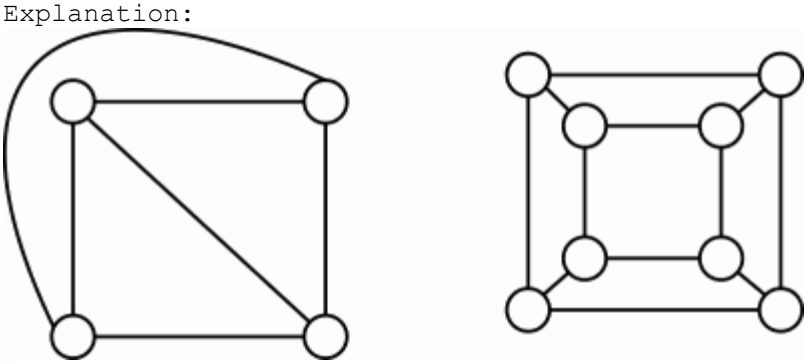
25. K4 and Q3 are graphs with the following structures.
K4 Q3



Which one of the following statements is **TRUE** in relation to these graphs?

(A) K4 is planar while Q3 is not
(B) Both K4 and Q3 are planar
(C) Q3 is planar while K4 is not
(D) Neither K4 nor Q3 are planar

Explanation:



Both can be made without intersecting edges