

IAT-2 Scheme & solution

Descriptive:  $4Q * 5M = 20M$

Multiple Choice:  $5Q * 2M = 10M$

Multiple Choice:  $20Q * 1M = 20M$

Total Marks = 50M

Descriptive:  $4Q * 5M = 20M$

1Q. Write a c function to reverse a singly linked list.

Solution:

**Input :** head node of the linked list

**Begin:**

**If** (head != NULL) then

prevNode ← head

head ← head.next

curNode ← head

prevNode.next ← NULL

**While** (head != NULL) do

head ← head.next

curNode.next ← prevNode

prevNode ← curNode

curNode ← head

**End while**

head ← prevNode

**End if**

**End**

2Q. Write a c function to delete a node at the front of a doubly linked list.

Solution:

```
NODE* delete_front(NODE *head)
{
    NODE *p;
    if(head==NULL)
        printf("\nList is Empty (QUEUE)");
    else
    {
        p
        =
        h
        e
        a
        d
        ;
        h
        e
        a
        d
        =
        h
        e
        a
        d
        -
        >
        r
        i
        g
        h
```

```

t
;
h
e
a
d
-
>
r
i
g
h
t
-
>
l
e
f
t
=
N
U
L
L
;
f
r
e
e
(
p
)
;
printf("\nFront(first)node is deleted");

return head;
}

```

```

NODE* delete_end(NODE *head)
{
    NODE *p, *q;
    p=head;
    while(p->right!=NULL)
    {
        p=p->right;    //Go upto -1 node which you want to delete
    }
    q=p->left;

    q->right=NULL; p->left=NULL;
    free(p);//Delete last node...
    printf("\nLast(end) entry is deleted");
    return head;
}

```

3Q. Write a c function to perform insertion operation on circular queue.

```
void insert()
{
    if((rear == MAX-1&&front==0)||((front=rear+1)))
        printf("\nQueue is Full");

    else {
        rear = (rear + 1) % MAX;
        q[rear]=item;
        count++;
    }
}
```

4Q. Write a c program to solve tower of Hanoi problem.

Solution:

```
#include<stdio.h>
#include<conio.h>

void tower(int n, int source, int temp,int destination)
{
    if(n == 0)
        return;
    tower(n-1, source, destination, temp);
    printf("\nMove disc %d from %c to %c", n, source, destination);
    tower(n-1, temp, source, destination);
}

void main()
{
    int n;
    clrscr();
    printf("\nEnter the number of discs: \n");
    scanf("%d", &n);
    tower(n, 'A', 'B', 'C');
    printf("\n\nTotal Number of moves are: %d", (int)pow(2,n)-1);
    getch();
}
```

Multiple Choice:

5Q\*2M=10M

1Q. Ackerman's function is defined on the non-negative integers. The value of  $a(2,4)$  is

- a) 11
- b) 8
- c) 7
- d) 20

2Q. What is the value of the postfix expression  $2\ 3\ +\ 4\ 6\ 5\ -\ +\ *$

- a) 19
- b) 21
- c) -4
- d) 25

3Q. What is the corresponding postfix expression for the given infix expression?

$(a+(b*c(d/e^f)*g)*h)$

- a)  $ab*cdef/\wedge*g-h+$
- b)  $abcdef/\wedge*g*h*+$
- c)  $abcd*\wedge ed/g*-h*+$
- d)  $abc*de\wedge fg/\wedge*-h+$

4Q. The prefix form of  $A-B/(C * D \wedge E)$  is?

- a)  $-\wedge ACBDE$
- b)  $-ABCD*\wedge DE$
- c)  $-A/B*C\wedge DE$
- d)  $-A/BC*\wedge DE$

5Q. Consider the following operations performed on a stack of size 5.

Push(9); pop(); Push(3); Push(6); pop(); Push(10); pop(); pop(); Push(25);

After completion of all the operations, the no. of elements present on stack are

- a) 2

b)1

c)3

d) 5

Multiple Choice:

20Q\*1M=20M

1. Which of the following data structure is used to convert postfix expression to infix expression?
  - a) **Stack**
  - b) Queue
  - c) Linked List
  - d) Heap
2. To convert the postfix expression into the infix expression we use stack and scan the postfix expression from left to right.
  - a) **True**
  - b) False
3. .... form of access is used to add remove nodes from a stack.
  - A. **LIFO**
  - B. FIFO
  - C. Both A and B
  - D. None of these
4. Which of the following is an application of stack?
  - A. finding factorial
  - B. tower of Hanoi
  - C. infix to postfix
  - D. **all of the above**
5. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a ?
  - a) **Queue**
  - b) Stack
  - c) Tree
  - d) Linked list

6. A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is?
- Queue
  - Circular queue
  - Deque**
  - Priority queue
7. A normal queue, if implemented using an array of size MAX\_SIZE, gets full when
- Rear = MAX\_SIZE - 1**
  - Front = (rear + 1) mod MAX\_SIZE
  - Front = rear + 1
  - Rear = front
8. Which of the following is not an advantage of priority queue?
- Easy to implement
  - Processes with different priority can be efficiently handled
  - Applications with differing requirements
  - Easy to delete elements in any case**
9. Minimum number of moves required to solve a tower of hanoi problem with n disks is
- 
- $2^n$
  - $2^n - 1$**
  - $n^2$
  - $n^2 - 1$
10. Prefix notation is also known as?
- Reverse Polish Notation
  - Polish Reverse Notation
  - Polish Notation**
  - Reverse Notation
11. Which of the following is false about a doubly linked list?
- We can navigate in both the directions
  - It requires more space than a singly linked list
  - The insertion and deletion of a node take a bit longer
  - Implementing a doubly linked list is easier than singly linked list**
12. What is a memory efficient double linked list?
- Each node has only one pointer to traverse the list back and forth**
  - The list has breakpoints for faster traversal
  - An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list
  - A doubly linked list that uses bitwise AND operator for storing addresses
13. Which of the following is not a disadvantage to the usage of array?
- Fixed size

- b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
- c) Insertion based on position
- d) **Accessing elements at specified positions**

14. Which of the following points is/are not true about Linked List data structure when it is compared with array?

- 1. Arrays have better cache locality that can make them better in terms of performance
- 2. It is easy to insert and delete elements in Linked List
- 3. Random access is not allowed in a typical implementation of Linked Lists
- 4. **Access of elements in linked list takes less time than compared to arrays**

15. Which of the following data structures can be used for parentheses matching?

- 1. n-ary tree
- 2. queue
- 3. priority queue
- 4. **stack**

16. A variant of a linked list in which the last node of the list points to the first node of the list is

- a) Singly linked list
- b) Doubly linked list
- c) **Circular linked list**
- d) None of the above

17. In doubly linked list traversal can be performed ?

- a. Only in forward direction
- b. Only in reverse direction
- c. **In both directions**
- d. None of the above

18. RLINK in the linked list is the pointer pointing to the ...

- A. **successor node**
- B. predecessor node
- C. head node
- D. last node

19. In a linked list, insertion can be done as .....

- A. beginning
- B. end
- C. middle
- D. **all of the above**

20. The disadvantage in using a circular linked list is .....

- A. **it is possible to get into an infinite loop**



- B. last node points to the first node.
- C. time consuming
- D. requires more memory space.