

1. An ADT is defined to be a mathematical model of a user-defined type along with the collection of all _____ operations on that model

- A. Cardinality
- B. Assignment
- C. Primitive**
- D. Structured

2. A linear collection of data elements where the linear node is given by means of pointer is called

- (A) Linked list
- (B) Node list
- (C) Primitive list**
- (D) None of these

3. Representation of data structure in memory is known as:

- A. Storage structure
- B. File structure
- C. abstract data type**
- D. None of the above

4. The use of pointers to refer elements of a data structure in which elements are logically adjacent is

- A. Stack
- B. Queue
- C. Pointers
- D. Linked allocation**

5. When new data are to be inserted into a data structure, but there is not available space; this situation is usually called

- A. Overflow**
- B. Underflow

C is memory full

D memory full

6. Mention the various operations in the List

Operations on List ADT

Insert(x,6) - Insert the element x after the position 6.

Delete(x) – The element x will be deleted.

Find(x) - Returns the position of x.

Previous(i) – Returns the position of its predecessor(i-1).

Next(i) - Returns the position of its successor(i+1).

PrintList() - Contents of the list is displayed.

MakeEmpty() – Makes the list empty.

7. How to insert an element in a list?

Insert()

{

int i,data,pos;

printf("\nEnter the data to be inserted:\t");

scanf("%d",&data);

printf("\nEnter the position at which element to be inserted:\t");

scanf("%d",&pos);

if (pos==n)

printf ("Array overflow");

for(i = n-1 ; i >= pos-1 ; i--)

list[i+1] = list[i];

list[pos-1] = data;

n=n+1;

}

8. How to delete a element in a list?

```
void Delete()  
{  
    int i, pos ;  
    printf("\nEnter the position of the data to be deleted:\t");  
    scanf("%d",&pos);  
    printf("\nThe data deleted is:\t %d", list[pos-1]);  
    for(i=pos-1;i<n-1;i++)  
        list[i]=list[i+1];  
    n=n-1;  
}
```

9. Pick up the nearest phonebook and open it to the first page of names. We're looking to find the first "Smith". Look at the first name. Is it "Smith"? Probably not (it's probably a name that begins with 'A'). Now look at the next name. Is it "Smith"? Probably not. Keep looking at the next name until you find "Smith".

This is an example of a sequential search. You start at the beginning of a sequence and went through each item one by one, in the order they existed in the list, until you found the item you were looking for.

```
#include <stdio.h>  
  
int LINEAR_SEARCH(int inp_arr[], int size, int val)  
{  
    for (int i = 0; i < size; i++)  
        if (inp_arr[i] == val)  
            return i;  
    return -1;  
}
```

```

int main(void)
{
    int arr[] = { 10, 20, 30, 40, 50, 100, 0 };
    int key = 100;
    int size = 10;
    int res = LINEAR_SEARCH(arr, size, key);
    if (res == -1)
        printf("ELEMENT NOT FOUND!!");
    else
        printf("Item is present at index %d", res);

    return 0;
}

```

10. How to traverse an array means to access each element (item) stored in the array so that the data can be checked or used as part of a process.

```

#include <stdio.h>

main()
{
    int Array[] = {1,3,5,7,8};
    int item = 10, k = 3, n = 5;
    int i = 0, j = n;
    printf("The original array elements are :\n");
    for(i = 0; i<n; i++)
    {
        printf("Array[%d] = %d \n", i, Array[i]);
    }
}

```