EX NO: 2	ARRAY IMPLEMENTATION OF LIST ADT

AIM:

To write a program for array implementation of list ADT

Algorithm:

Step1: Create nodes first, last; next, prev and cur
then set the value as NULL.
Step 2: Read the list operation type.
Step 3: If operation type is create then process the following steps.

- Allocate memory for node cur.
- Read data in cur's data area.
- Assign cur node as NULL.
- Assign first=last=cur.

Step 4: If operation type is Insert then process the following steps.

- Allocate memory for node cur.
- Read data in cur's data area.
- Read the position the Data to be insert.
- Availability of the position is true then assing
- cur's node as first and first=cur.
- If availability of position is false then do following steps.
    - Assign next as cur and count as zero.
    - Repeat the following steps until count less than
    - postion.
    - 1 .Assign prev as next
    - Next as prev of node.
    - Add count by one.
    - If prev as NULL then display the message INVALID
    - POSITION.
    - If prev not qual to NULL then do the following steps.
    - Assign cur's node as prev's node.
    - Assign prev's node as cur.

Step5: If operation type is delete then do the following steps.

- Read the position .
- Check list is Empty .If it is true display the message List empty.
- If position is first. Assign cur as first.
- Assign First as first of node.
- Reallocate the cur from memory.
- If position is last.
- Move the current node to prev. cur's node as Null.
- Reallocate the Last from memory.
- Assign last as cur.
- If position is enter Mediate.
- Move the cur to required postion.
- Move the Previous to cur's previous position
- Move the Next to cur's Next position.
- Now Assign previous of node as next.
- Reallocate the cur from memory.

step 6: If operation is traverse.

- Assign current as first.
- Repeat the following steps untill cur becomes NULL

Program:

```
#include<stdio.h>
 #include<conio.h>
        #define MAX 10
        void create();
        void insert();
        void deletion();
        void search();
        void display();
        int a,b[20], n, p, e, f, i, pos;
        void main()
        {
        //clrscr();
        int ch;
        char g='y';
        do
        {
        printf("\n main Menu");
        printf("\n 1.Create \n 2.Delete \n 3.Search \n 4.Insert \n 5.Display\n 6.Exit
        \n"); printf("\n Enter your Choice");
        scanf("%d", &ch);
        switch(ch)
        {
        case 1: create(); break;
        case 2: deletion(); break;
        case 3:search();break;
        case 4:insert();break;
        case 5:display();break;
        case 6:exit();break;
        default:
        printf("\n Enter the correct choice:");
        }
        printf("\n Do u want to continue:::");
        scanf("\n%c", &g);
        }
        while(g=='y'||g=='Y');
        getch();
        }
        void create()
        {
        printf("\n Enter the number of nodes");
        scanf("%d", &n);
        for(i=0;i<n;i++)
        {
        printf("\n Enter the Element:",i+1);
```

```
scanf(“%d”, &b[i]);
}
}
void deletion()
{
printf(“\n Enter the position u want to delete::”);
scanf(“%d”, &pos);
if(pos>=n)
{
printf(“\n Invalid Location::”);
}
else
{
for(i=pos+1;i<n;i++)
{
b[i-1]=b[i];
}
n—;
}
printf(“\n The Elements after deletion”);
for(i=0;i<n;i++)
{
printf(“\t%d”, b[i]);
}
}
void search()
{
printf(“\n Enter the Element to be searched:”);
scanf(“%d”, &e);
for(i=0;i<n;i++)
{
if(b[i]==e)
{
printf(“Value is in the %d Position”, i);
}
else
{
printf(“Value %d is not in the list::”, e);
continue;
}
}
}
void insert()
{
printf(“\n Enter the position u need to insert::”);
scanf(“%d”, &pos);
if(pos>=n)
 {
 printf(“\n invalid Location::”);
```

```
 }
 else
 {
 for(i=MAX-1;i>=pos-1;i—)
 {
 b[i+1]=b[i];
 }
 printf(“\n Enter the element to insert::\n”);
 scanf(“%d”,&p);
 b[pos]=p;
 n++;
 }
 printf(“\n The list after insertion::\n”);
display();
}
void display()
{
printf(“\n The Elements of The list ADT are:”);
for(i=0;i<n;i++)
{
printf(“\n\n%d”, b[i]);
}
}
```

Output:

```
Main Menu
1.Create    2.Delete    3.Search    4.Insert    5.Display    6.Exit
Enter your Choice 1
Enter the number of nodes 3
Enter the Element 8 9 7

Main Menu
1.Create    2.Delete    3.Search    4.Insert    5.Display    6.Exit
Enter your Choice 2
Enter the position u want to delete 2
The Elements after deletion 8 7

Main Menu
1.Create    2.Delete    3.Search    4.Insert    5.Display    6.Exit
Enter your Choice 4
Enter the position u need to insert 2
Enter the element to insert 6
The list after insertion 8 6 7

Main Menu
1.Create    2.Delete    3.Search    4.Insert    5.Display    6.Exit
Enter your Choice 6
```

Result:

Thus the program for array implementation of list ADT is executed sucessfully & verified