1. Design, Develop and Implement a menu driven Program in C for the following Array operations

a. Creating an Array of N Integer Elements

b. Display of Array Elements with Suitable Headings

c. Inserting an Element (ELEM) at a given valid Position (POS)

d. Deleting an Element at a given valid Position(POS)

e. Exit.

Support the program with functions for each of the above operations.

Algorithm:

```
Step 1: Start the Program
Step 2: Declare the Variables
Step 3: Declare the functions with the prototypes
Step 4: Get the value of ‘n’ using scanf() function.
Step 5: Use switch case for each function
Step 6: Case 1: Create an array with ‘n’ elements
Step 7: Using for loop scan ‘n’ elements of an array
Step 8: Case 2: Display ‘n’ elements of an array using printf() function in the for loop
Step 9: Case 3: Insert the element into an array
Step 10: Check whether the element position is equal to or greater than ‘n’. If so, then display the
array is
full.
Step 11: Otherwise, get the element position and its value to be inserted. Then display the array
Step 12: Case 4: Check the condition whether ‘n’ is equal to ‘0’. If so, then display the array is
empty.
Step 13: Otherwise, get the position of the element to be deleted. Then display the array
Step 14: Case 5: Quit the Switch case program by calling exit() function.
Step 15: Stop the Program execution.
```

Program:

```
#include<stdio.h>

#include<stdlib.h>

int a[10], pos, elem;

int n = 0;

void create();

void display();

void insert();

void del();

void main()

{

int choice;

while(1)

{

printf("\n\n~~~~MENU~~~~");
```

```
printf("\n=>1. Create an array of N integers");
printf("\n=>2. Display of array elements");
printf("\n=>3. Insert ELEM at a given POS");
printf("\n=>4. Delete an element at a given POS");
printf("\n=>5. Exit");
printf("\nEnter your choice: ");
scanf("%d", &choice);
switch(choice)
{
case 1: create();
break;
case 2: display();
break;
case 3: insert();
break;
case 4:del();
break;
case 5:exit(1);
break;
default:
printf("\nPlease enter a valid choice:");
}
}
}
void create()
{
int i;
printf("\nEnter the number of elements: ");
scanf("%d", &n);
printf("\nEnter the elements: ");
```

```
for(i=0; i<n; i++)
{
scanf("%d", &a[i]);
}
}
void display()
{
int i;
if(n == 0)
{
printf("\nNo elements to display");
return;
}
printf("\nArray elements are: ");
for(i=0; i<n;i++)
printf("%d\t ", a[i]);
}
void insert()
{
int i;
if(n == 5)
{
printf("\nArray is full. Insertion is not possible");
return;
}
do
{
printf("\nEnter a valid position where element to be inserted: ");
scanf("%d", &pos);
}while(pos > n);
```

```
printf("\nEnter the value to be inserted: ");
scanf("%d", &elem);
for(i=n-1; i>=pos ; i--)
{
a[i+1] = a[i];
}
a[pos] = elem;
n = n+1;
display();
}
void del()
{
int i;
if(n == 0)
{
printf("\nArray is empty and no elements to delete");
return;
}
do
{
printf("\nEnter a valid position from where element to be deleted: ");
scanf("%d", &pos);
}while(pos>=n);
elem = a[pos];
printf("\nDeleted element is : %d \n", elem);
for( i = pos; i< n-1; i++)
{
a[i] = a[i+1];
}
n = n-1;
```

```
display();
}
```

Result :

Thus, the ‘C’ Program for Design, Development and implementing a menu driven Program to perform creation of an array, display, insertion and deletion is executed and the output is verified successfully.

2. Design, Develop and Implement a menu driven Program in C for the following operations
on STACK of Integers (Array Implementation of Stack with maximum size MAX)
a. Push an Element on to Stack
b. Pop an Element from Stack
c. Demonstrate how Stack can be used to check Palindrome
d. Demonstrate Overflow and Underflow situations on Stack
e. Display the status of Stack
f. Exit
Support the program with appropriate functions for each of the above operations

```
#include <stdio.h>
#include <stdlib.h>
#include<conio.h>
int s[5],top=-1;
void push()
{
   if(top==4)
      printf("\nStack overflow!!!!");
   else
   {
      printf("\nEnter element to insert:");
      scanf("%d",&s[++top]);
   }
}
void pop()
{
   if(top==-1)
```

```
        printf("\nStack underflow!!!");
    else
        printf("\nElement popped is: %d",s[top--]);
}
void disp()
{
    int t=top;
    if(t==-1)
        printf("\nStack empty!!");
    else
        printf("\nStack elements are:\n");
    while(t>=0)
        printf("%d ",s[t--]);
}
void pali()
{
    int num[5],rev[5],i,t;
    for(i=0,t=top;t>=0;i++,t--)
        num[i]=rev[t]=s[t];
    for(i=0;i<=top;i++)
        if(num[i]!=rev[i])
        break;
    if(i==top+1)
        printf("\nIt is a palindrome");
    else
        printf("\nIt is not a palindrome");
}
```

```
int main()
{
    int ch;
    do
    {
        printf("\n...Stack operations.....\n");
        printf("1.PUSH\n");
        printf("2.POP\n");
        printf("3.Palindrome\n");
        printf("4.Display\n");
        printf("5.Exit\n________________\n");
        printf("Enter choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:push();break;
            case 2:pop();break;
            case 3:pali();break;
            case 4:disp();break;
            case 5:exit(0);
            default:printf("\nInvalid choice");
        }
    }
    while(1);
    return 0;
}
```