Program 3) Develop 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<stdlib.h>
#include<stdio.h>
#include<string.h>
#define max_size 5
int stack[max_size],top=-1,flag=1;
int i,temp,item,rev[max_size],num[max_size];
void push();
void pop();
void display();
void pali();
int main(){
   int choice;
   printf("\n\n--------STACK OPERATIONS--------\n");
   printf("1.Push\n");
   printf("2.Pop\n");
   printf("3.Palindrome\n");
   printf("4.Display\n");
   printf("5.Exit\n");
   printf(" ");
   while(1){
      printf("\nEnter your choice:\t");
```

```
scanf("%d",&choice);
switch(choice){
   case 1: push();
   break;
   case 2: pop();
   if(flag)
   printf("\nThe poped element: %d\t",item);
   temp=top;
   break;
   case 3: pali();
   top=temp;
   break;
   case 4: display();
   break;
   case 5: exit(0);
   break;
   default: printf("\nInvalid choice:\n");
   break;
}
}
//return 0;
}
void push() //Inserting element into the stack
{
   if(top==(max_size-1)){
      printf("\nStack Overflow:");
   }else{
      printf("Enter the element to be inserted:\t");
      scanf("%d",&item);
      top=top+1;
      stack[top]=item;
```

```
    }
    temp=top;
}void pop() //deleting an element from the stack
{
    if(top==-1){
        printf("Stack Underflow:");
        flag=0;
    }else{
        item=stack[top];
        top=top-1;
    }
}
    void pali(){
        i=0;
        if(top==-1){
            printf("Push some elements into the stack first\n");
        }else{
            while(top!=-1){
                rev[top]=stack[top];
                pop();
            }
            top=temp;
            for(i=0;i<=temp;i++){
                if(stack[top--]==rev[i]){
                    if(i==temp){
                        printf("Palindrome\n");
                        return;
                    }}}
                    printf("Not Palindrome\n");
        }}
        void display(){
```

```
int i;
top=temp;
if(top==-1){
   printf("\nStack is Empty:");
}else{
   printf("\nThe stack elements are:\n" );
   for(i=top;i>=0;i--){
      printf("%d\n",stack[i]);
   }}}
```