**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<stdio.h>

#include<stdlib.h>

#define MAX 5

int s[MAX];

int top = -1;

void push(int item);

int pop();

void palindrome();

void display();

void main()

{

            int choice, item;

            while(1)

            {

                        printf("\n\n\n\n~~~~~~Menu~~~~~~ : ");

                        printf("\n=>1.Push an Element to Stack and Overflow demo ");

                        printf("\n=>2.Pop an Element from Stack and Underflow demo");

                        printf("\n=>3.Palindrome demo ");

                        printf("\n=>4.Display ");

                        printf("\n=>5.Exit");

                        printf("\nEnter your choice: ");

                        scanf("%d", &choice);

                        switch(choice)

                        {

                                    case 1:             printf("\nEnter an element to be pushed: ");

                                                            scanf("%d", &item);

**push(item);**

                                                            break;

                                    case 2:             **item = pop();**

                                                            if(item != -1)

                                                                        printf("\nElement popped is: %d", item);

                                                            break;

                                    case 3:             **palindrome();**

                                                            break;

                                    case 4:             **display();**

                                                            break;

                                    case 5:             exit(1);

                                    default:            printf("\nPlease enter valid choice ") ;

                                                            break;

                    }

            }

}

void push(int item)

{

            if(top == MAX-1)

            {

                        printf("\n~~~~Stack overflow~~~~");

                        return;

            }

            top = top + 1 ;

            s[top] = item;

}

int pop()

{

            int item;

            if(top == -1)

            {

                        printf("\n~~~~Stack underflow~~~~");

                        return -1;

            }

            item = s[top];

            top = top - 1;

            return item;

}

void display()

{

            int i;

            if(top == -1)

            {

                        printf("\n~~~~Stack is empty~~~~");

                        return;

            }

            printf("\nStack elements are:\n ");

            for(i=top; i>=0 ; i--)

                        printf("| %d |\n", s[i]);

}

void palindrome()

{

            int flag=1,i;

            printf("\nStack content are:\n");

            for(i=top; i>=0 ; i--)

                        printf("| %d |\n", s[i]);

            printf("\nReverse of stack content are:\n");

            for(i=0; i<=top; i++)

                        printf("| %d |\n", s[i]);

            for(i=0; i<=top/2; i++)

            {

                        if( s[i] != s[top-i] )

                        {

                                    flag = 0;

                                    break;

                        }

            }

            if(flag == 1)

            {

                        printf("\nIt is palindrome number");

            }

            else

            {

                        printf("\nIt is not a palindrome number");

            }

}