```
/*3. 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*/

#define MAX 50
#include<math.h>
#include<stdio.h>
#include<stdlib.h>

int stk[MAX], top = -1;

int overflow(void)                  /* Function for Overflow operation */
{
      if (top == MAX - 1)           /* Overflow if top is max size */
            return 1;
      else
            return 0;
}

int underflow(void)                 /* Function for underflow operation */
{
      if (top == -1)                /* Underflow if top is -1 */
            return 1;
      else
            return 0;
}

void push(int ele)                  /* Function for push operation */
{
      if (overflow())
      {
            printf("Stack Overflow\n");
            return;
      }
      else
            stk[++top] = ele;       /* Push ele to top of stk[] */
}

int pop(void)                       /* Function for push operation */
{
      if (underflow())
      {
            printf("Stack Underflow\n");
            return -1;
      }
      else
      {
            return stk[top--];    /*Return popped ele from top of stk[] */
      }
```

```
}

void display(void)                  /* Function for display operation */
{
      int i;
      if (underflow())
      {
            printf("Stack is Empty\n");
            return;
      }
      else
      {
            for (i = top;i >= 0;i--)
            {                 /* Display content of stk[] upto top */
                  printf("||  %d  ||\n", stk[i]);
                  printf(" ========\n");
            }
            printf("\n");
      }
}

void check_palindrome(int pal)
{
      int i, len=0,count=0;
      len = log10(pal) + 1;
      for (i = 0;i < len / 2;i++)
      {
            push(pal % 10);
            pal = pal / 10;
      }
      if ((len % 2)!= 0)
      {
            pal = pal / 10;
      }
      for (i = 0;i < len / 2;i++)
      {
            if (pal % 10 == pop())
                  count++;
            pal = pal / 10;
      }

      if (count == len/2)
            printf("Stack is Palindrome\n");  /*if same count and len
then palindrome*/
      else
            printf("Stack is Not a Palindrome\n");

}

int main()
{
      int  ele,ch,pal;
      while (1)
      {
            printf("\n**********Stack Operations Menu**********\n");
            printf("1. Push Operation\n");
            printf("2. Pop Operation\n");
            printf("3. Check Palindrome\n");
```

```
            printf("4. Display Stack\n");
            printf("5. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1: printf("Enter an element to push\n");
                  scanf("%d", &ele);
                  push(ele);
                  break;
            case 2: ele = pop();
                  printf("Popped element is:   %d\n",ele);
                  break;
            case 3: printf("Enter a number to check Palindrome\n");
                  scanf("%d", &pal);
                  if (log10(pal) > 2)
                  {
                        check_palindrome(pal);
                  }
                  else
                        printf("Enter minimum of 3 digits  number\n");
                  break;
            case 4: display();
                  break;
            case 5: exit(0);
            default: printf("Enter the valid choice\n\n");
                  break;
            }
      }
      return 0;
}
```