190905514
MOHAMMAD TOFIK

WEEK 3 LAB 3

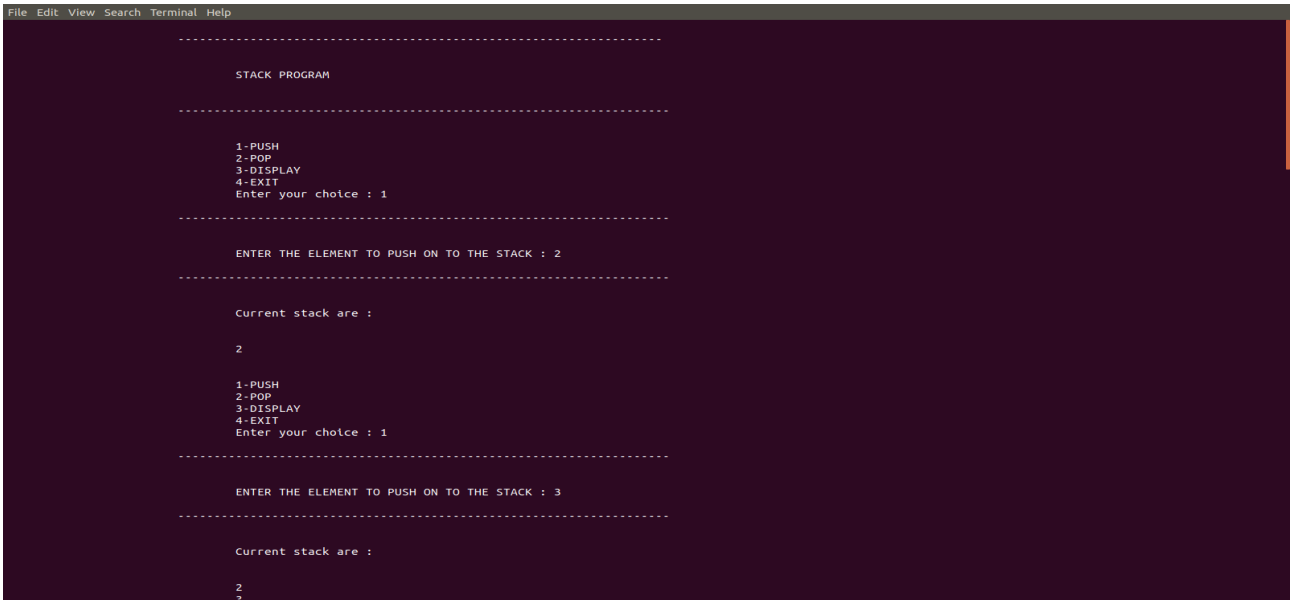
1) Implement a menu driven program to define a stack of characters. Include push, pop and display functions. Also include functions for checking error conditions such as underflow and overflow (ref. figure 1) by defining isEmpty and isFull functions. Use these function in push, pop and display functions appropriately. Use type defined structure to define a STACK containing a character array and an integer top. Do not use global variables.

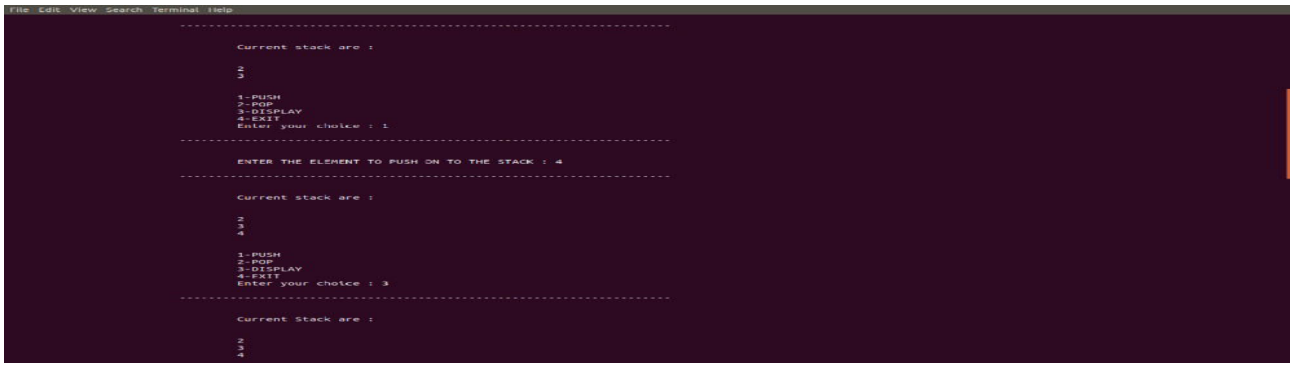
Program1.c

```
#include <stdio.h>
#include <stdlib.h>


#define SIZE 10
#define UNDERFLOW '\0'


typedef enum
{
   NO = 0,
   YES = 1,
} BOOL;


BOOL isStackFull (int tos)
{
   if (tos == SIZE - 1)
      return YES;
   return NO;
}

BOOL isStackEmpty (int tos)
{
   if (tos == -1)
      return YES;
   return NO;
}

void push (char *stack, int item, int *tos)
{
   if (isStackFull (*tos))
   {
      printf("\n\t\t\t\tSTACK IS FULL\n");
      return;
   }
   (*tos) += 1;
   *(stack + (*tos)) = item;
}

char pop (char *stack, int *tos)
{
   if (isStackEmpty (*tos))
   {
      printf("\n\t\t\t\tSTACK IS EMPTY : ");
      return UNDERFLOW;
   }
```

```
    return *(stack + ((*tos)--));
}
void display(char *stack,int tos)
{
    printf("\n\n");
    char *pi;
    for(pi=stack; pi<=stack+tos; ++pi)
        printf("\n\t\t\t\t%c",*pi);
    printf("\n\n");
}
int main(int argc, const char * argv[])
{
    int tos = -1;
    char *stack = (char *)malloc(sizeof(char *));
    char ch = '3';
    printf("\n\t\t\t----------\n\n");
    printf("\n\t\t\t\tSTACK PROGRAM \n\n");
    printf("\n\t\t\t----------\n\n");
    while (ch == '1' || ch == '2' || ch == '3')
    {
        printf("\n\t\t\t\t1-PUSH");
        printf("\n\t\t\t\t2-POP");
        printf("\n\t\t\t\t3-DISPLAY");
        printf("\n\t\t\t\t4-EXIT");
        printf("\n\t\t\t\tEnter your choice : ");
        scanf(" %c", &ch);
        printf("\n\t\t\t----------\n\n");
        if (ch == '1')
        {
            char item;
            printf("\n\t\t\t\tENTER THE ELEMENT TO PUSH ON TO THE STACK : ");
            scanf(" %c", &item);
            printf("\n\t\t\t----------\n\n");
            push(stack, item, &tos);
            if (!isStackFull(tos))
            {
                printf("\n\t\t\t\tCurrent stack are : ");
                display(stack, tos);
            }
        }
        else if (ch == '2')
        {
            char item = pop(stack, &tos);
            if (item != UNDERFLOW)
            {
                printf("\n\t\t\t\tPopped item is  = %c", item);
                printf("\n\n");
                printf("\n\t\t\t\tCurrent Stack are: ");
                display(stack, tos);
            }
        }
        else if (ch == '3')
        {
            printf("\n\t\t\tCurrent Stack are : ");
            display(stack, tos);
        }
        else
            exit(0);
    }
    return 0;
}
```

OUTPUT :


File Edit View Search Terminal Help
STACK PROGRAM
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 1
ENTER THE ELEMENT TO PUSH ON TO THE STACK : 2
Current stack are :
2
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 1
ENTER THE ELEMENT TO PUSH ON TO THE STACK : 3
Current stack are :
2


File Edit View Search Terminal Help
Current stack are :
2
3
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 1
ENTER THE ELEMENT TO PUSH ON TO THE STACK : 4
Current stack are :
2
3
4
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 3
Current Stack are :
2
3
4


File Edit View Search Terminal Help
2
3
4
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 2
Popped item is = 4
Current Stack are:
2
3
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 3
Current Stack are :
2
3
1-PUSH
2-POP
3-DISPLAY
4-EXIT
Enter your choice : 2

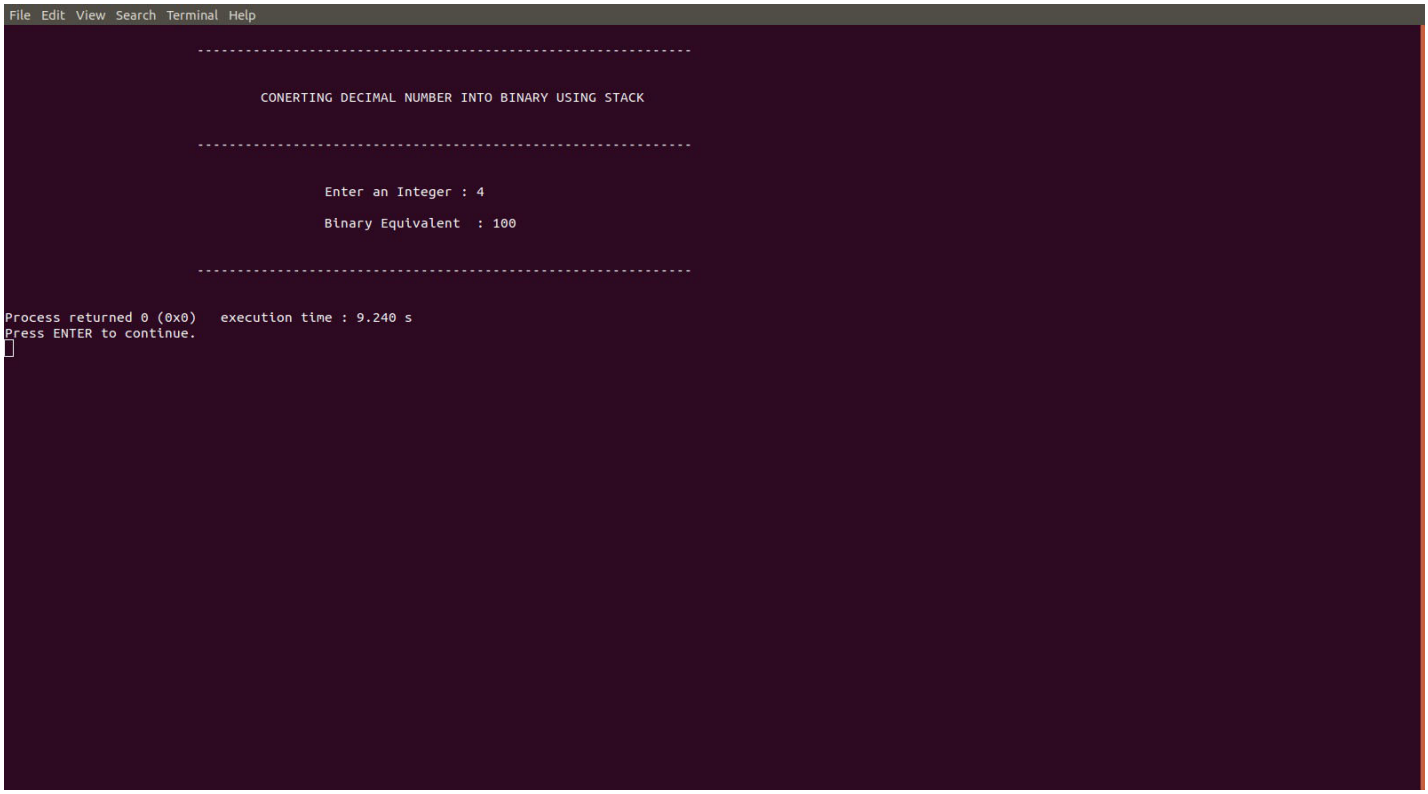
2) Convert a given decimal number to binary using stack.

Program2.c

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 100

int isEmpty(int top, int stack_arr[]);
void push(int x, int *top, int stack_arr[]);
int pop(int *top, int stack_arr[]);
void decimaltobinary(int number);

int main()
{
   int number;
   printf("\n\t\t\t\t----------------------------------------------------------\n\n");
   printf("\n\t\t\t\tCONERTING DECIMAL NUMBER INTO BINARY USING STACK \n\n");
   printf("\n\t\t\t\t----------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tEnter an Integer : ");
   scanf("%d",&number);
   printf("\n\t\t\t\t\tBinary Equivalent  : ");
   decimaltobinary(number);

   return 0;
```

```
}

void decimaltobinary(int number)
{
    int stack[MAX], top=-1, rem;
    while(number!=0)
    {
        rem = number%2;
        push(rem, &top, stack);
        number/=2;
    }
    while(top!=-1)
        printf("%d", pop(&top, stack));
    printf("\n\n");
    printf("\n\t\t\t\t----------------------------------------------------------------\n\n");
}


void push(int x, int *top, int stack_arr[])
{
    if(*top == (MAX-1))
        printf("\n\t\t\t\t\t\tStack is full :");
    else
    {
        *top=*top+1;
        stack_arr[*top] = x;
    }
}

int pop(int *top, int stack_arr[])
{
    int x;
    if(*top == -1)
    {
        printf("\n\t\t\t\t\t\tStack is empty : ");
        exit(1);
    }
    else
    {
        x = stack_arr[*top];
        *top=*top-1;
    }
    return x;
}
```

OUTPUT :



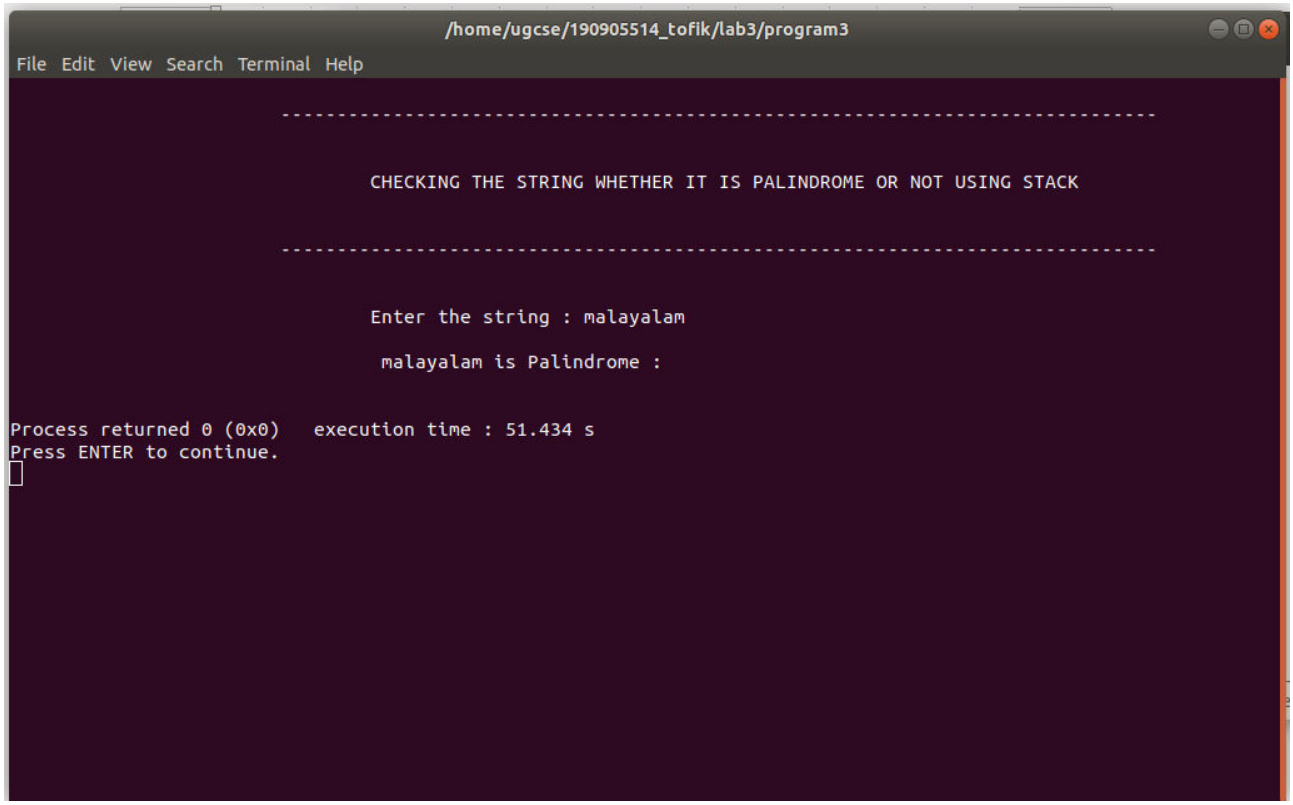
3) Determine whether a given string is palindrome or not using stack.

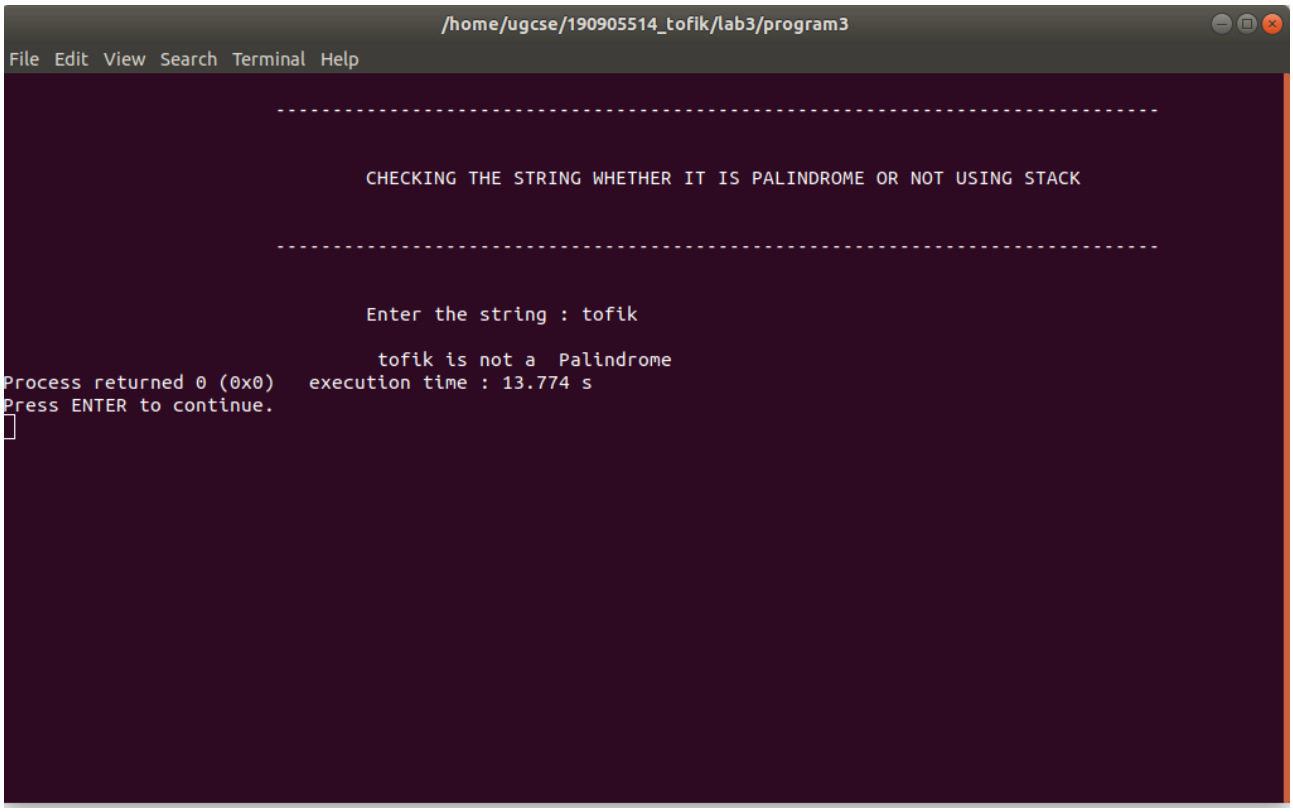
Program3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 1000
#define EMPTY '\0'
typedef enum {
   PRACTICE = 0,
   ALLISWELL = 1,
} BOOLEAN;
BOOLEAN isStackFull (int tos) {
   if (tos == SIZE - 1)
      return ALLISWELL;
   return PRACTICE;
}
BOOLEAN isStackEmpty (int tos) {
   if (tos == -1)
      return ALLISWELL;
   return PRACTICE;
}
void push (char *stack, int item, int *tos) {
   if (isStackFull (*tos)) {
      printf("\n\t\t\t\tSTACK IS FULL : \n\n");
      return;
   }
   (*tos) += 1;
   *(stack + (*tos)) = item;
}
char pop (char *stack, int *tos) {
   if (isStackEmpty (*tos)) {
```

```
        printf("\n\t\t\t\tSTACK IS EMPTY\n\n");
        return EMPTY;
    }
    return *(stack + ((*tos)--));
}
void display (char *stack, int tos) {
    printf("\n");
    char *pi;
    for (pi = stack; pi <= stack + tos; ++pi)
        printf("\n\t\t\t\t%c", *pi);
    printf("\n\n");
}
BOOLEAN isPalindrome (char *str) {
    int tos = -1, i;
    char *stack = (char *)malloc(sizeof(char *));

    for (i = 0; i < strlen(str); ++i)
        push(stack, str[i], &tos);

    for (i = 0; i < strlen(str)/2; ++i)
        if (pop(stack, &tos) != str[i])
            return ALLISWELL;
    return PRACTICE;
}
int main(int argc, const char * argv[]) {

    char *str = (char *)malloc(SIZE * sizeof(char));
    printf("\n\t\t\t----------------------------------------------------------------------------\n\n");
    printf("\n\t\t\t\tCHECKING THE STRING WHETHER IT IS PALINDROME OR NOT USING STACK \n\n");
    printf("\n\t\t\t----------------------------------------------------------------------------\n\n");
    printf("\n\t\t\t\tEnter the string : ");
    scanf("%s", str);

    if (!isPalindrome (str))
        printf("\n\t\t\t\t %s is Palindrome :  \n\n", str);

    else
        printf("\n\t\t\t\t %s is not a  Palindrome ", str);

    return 0;
}
```

OUTPUT :


/home/ugcse/190905514_tofik/lab3/program3
File Edit View Search Terminal Help
CHECKING THE STRING WHETHER IT IS PALINDROME OR NOT USING STACK
Enter the string : malayalam
malayalam is Palindrome :
Process returned 0 (0x0) execution time : 51.434 s
Press ENTER to continue.


/home/ugcse/190905514_tofik/lab3/program3
File Edit View Search Terminal Help
CHECKING THE STRING WHETHER IT IS PALINDROME OR NOT USING STACK
Enter the string : tofik
tofik is not a Palindrome
Process returned 0 (0x0) execution time : 13.774 s
Press ENTER to continue.