

Ayush Goyal
190905522

DSA Lab 3 (Session 2)

Q1) 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.

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

#define MAX 4

typedef struct {
    char data[MAX];
    int top;
} stack;

bool isEmpty(stack* s){
    if(s->top == -1){
        return true;
    }
    else return false;
}

bool isFull(stack* s){
    if(s->top == MAX-1){
        return true;
    }
    else return false;
}

void push(stack* s, char c){
    if(isFull(s)){
        printf("Stack is full.\n");
        return;
    }
    s->top++;
    s->data[s->top] = c;
}

char pop(stack* s){
    if(!isEmpty(s)){
        return(s->data[s->top--]);
    }
}

void display(stack* s){
    if(isEmpty(s)){
```

```

        printf("Stack is empty\n");
        return;
    }
    int count = s->top;
    while(count > -1){
        printf("%c\n",s->data[count--]);
    }
}

int main(){
    stack st;
    stack* s = &st;
    s->top = -1;
    int n=0;
    char ch;
    do{
        printf("\nEnter:\t 1 to push \t 2 to pop\t3 to display\t4 to exit.");
        scanf("%d",&n);
        switch(n){
            case 1 : printf("Enter char to push : ");
                     scanf(" %c",&ch);
                     push(s,ch);
                     break;
            case 2 : pop(s);
                     break;
            case 3 : display(s);
                     break;
            case 4 : break;
        }
    }while(n != 4);
}

```

```
student@dslab: ~/Desktop/DSLAbAyush
File Edit View Search Terminal Help
student@dslab:~/Desktop/DSLAbAyush$ gcc l3q1.c -o l3q1
student@dslab:~/Desktop/DSLAbAyush$ ./l3q1

Enter:  1 to push      2 to pop      3 to display  4 to exit.1
Enter char to push : 1

Enter:  1 to push      2 to pop      3 to display  4 to exit.1
Enter char to push : 2

Enter:  1 to push      2 to pop      3 to display  4 to exit.1
Enter char to push : 3

Enter:  1 to push      2 to pop      3 to display  4 to exit.1
Enter char to push : 3

Enter:  1 to push      2 to pop      3 to display  4 to exit.3
3
3
2
1

Enter:  1 to push      2 to pop      3 to display  4 to exit.2
Enter:  1 to push      2 to pop      3 to display  4 to exit.2
Enter:  1 to push      2 to pop      3 to display  4 to exit.3
2
1

Enter:  1 to push      2 to pop      3 to display  4 to exit.4
student@dslab:~/Desktop/DSLAbAyush$
```

Q2) Convert a given decimal number to binary using stack.

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

#define MAX 10

typedef struct {
    int data[MAX];
    int top;
} stack;

void push(stack* s, int c){
    s->top++;
    s->data[s->top] = c;
}

int pop(stack* s){
    return(s->data[s->top--]);
}
```

```

void display(stack* s){
    int count = s->top;
    while(count > -1){
        printf("%d",s->data[count--]);
    }
}

int main(){
    stack st;
    stack* s = &st;
    s->top = -1;
    int n,r;
    printf("\nEnter decimal number : ");
    scanf("%d",&n);
    int q=n;
    while(q!=0){
        r = q%2;
        push(s,r);
        q = q/2;
    }
    printf("\nEquivalent binary number is : ");
    display(s);
    printf("\n");
}

```

```

student@dslab: ~/Desktop/DSLAbAyush
File Edit View Search Terminal Help
student@dslab:~/Desktop/DSLAbAyush$ gcc l3q2.c -o l3q2
student@dslab:~/Desktop/DSLAbAyush$ ./l3q2

Enter decimal number : 6

Equivalent binary number is : 110
student@dslab:~/Desktop/DSLAbAyush$ ./l3q2

Enter decimal number : 15

Equivalent binary number is : 1111
student@dslab:~/Desktop/DSLAbAyush$ ./l3q2

Enter decimal number : 16

Equivalent binary number is : 10000
student@dslab:~/Desktop/DSLAbAyush$

```

Q3) Determine whether a given string is palindrome or not using stack.

```

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

#define MAX 10

typedef struct {
    char data[MAX];
    int top;
} stack;

void push(stack* s, char e){
    s->top++;
    s->data[s->top] = e;
}

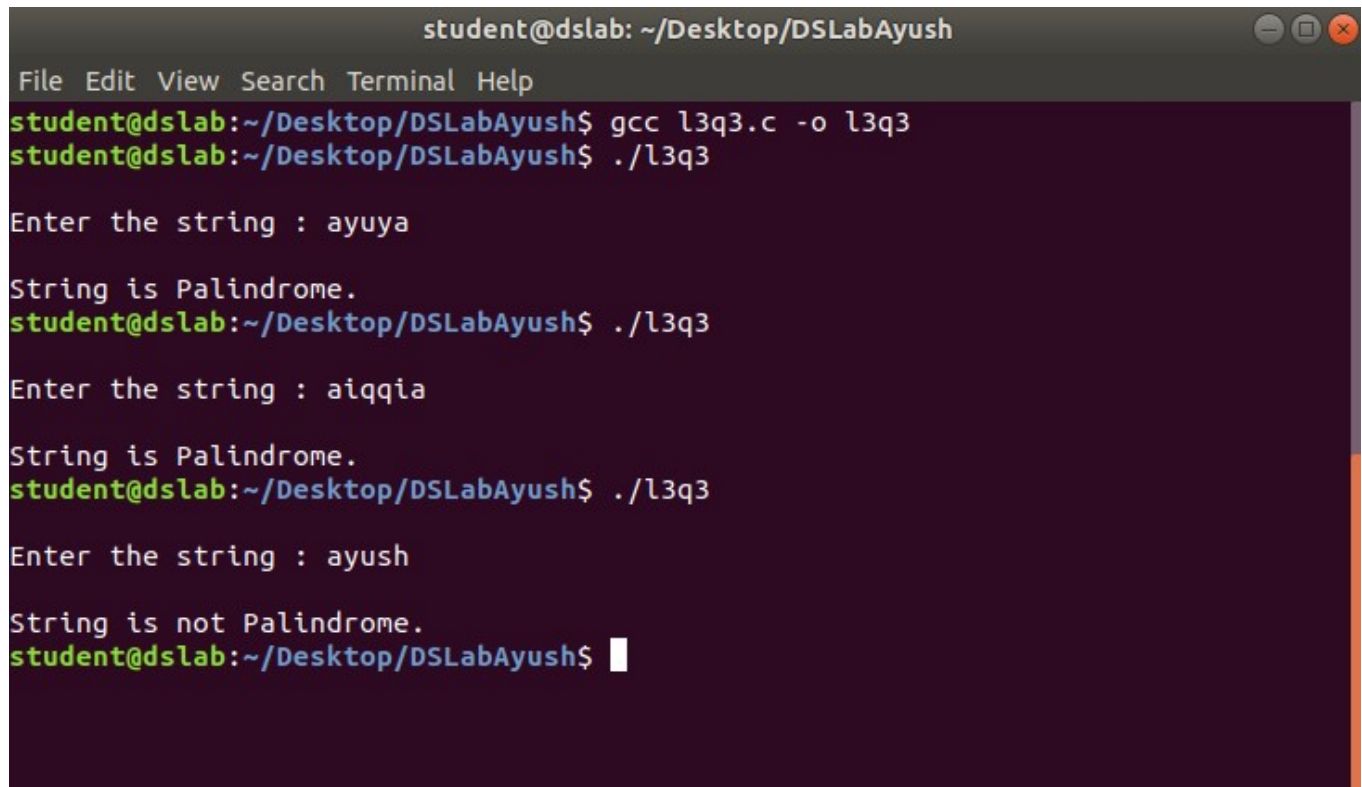
char pop(stack* s){
    return(s->data[s->top--]);
}

int isPalindrome(stack* s, char str[]){
    int l = strlen(str);
    //printf("strlen = %d",l);
    int i, m = l/2;
    for(i=0;i<m;i++){
        push(s,str[i]);
    }
    if(l%2!=0){
        i++;
    }
    char e;
    while(str[i] != '\0'){
        e = pop(s);
        //printf("%c\t%c\t",e,str[i]);
        if(e!=str[i])
            return 0;
        i++;
    }
    return 1;
}

int main(){
    stack st;
    stack* s = &st;
    s->top = -1;
    char str[20];
    printf("\nEnter the string : ");
    scanf("%s",str);
    if(isPalindrome(s,str))
        printf("\nString is Palindrome.\n");
    else
        printf("\nString is not Palindrome.\n");
}

```

```
    return 0;  
}
```



A terminal window titled "student@dslab: ~/Desktop/DSLAbAyush" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the compilation and execution of a C program named l3q3.c. The program checks if input strings are palindromes. The first two inputs, "ayuya" and "aiqqia", are correctly identified as palindromes. The third input, "ayush", is correctly identified as not a palindrome.

```
student@dslab: ~/Desktop/DSLAbAyush  
File Edit View Search Terminal Help  
student@dslab:~/Desktop/DSLAbAyush$ gcc l3q3.c -o l3q3  
student@dslab:~/Desktop/DSLAbAyush$ ./l3q3  
Enter the string : ayuya  
String is Palindrome.  
student@dslab:~/Desktop/DSLAbAyush$ ./l3q3  
Enter the string : aiqqia  
String is Palindrome.  
student@dslab:~/Desktop/DSLAbAyush$ ./l3q3  
Enter the string : ayush  
String is not Palindrome.  
student@dslab:~/Desktop/DSLAbAyush$
```