DSA Lab

Week 3 Assignment Submission

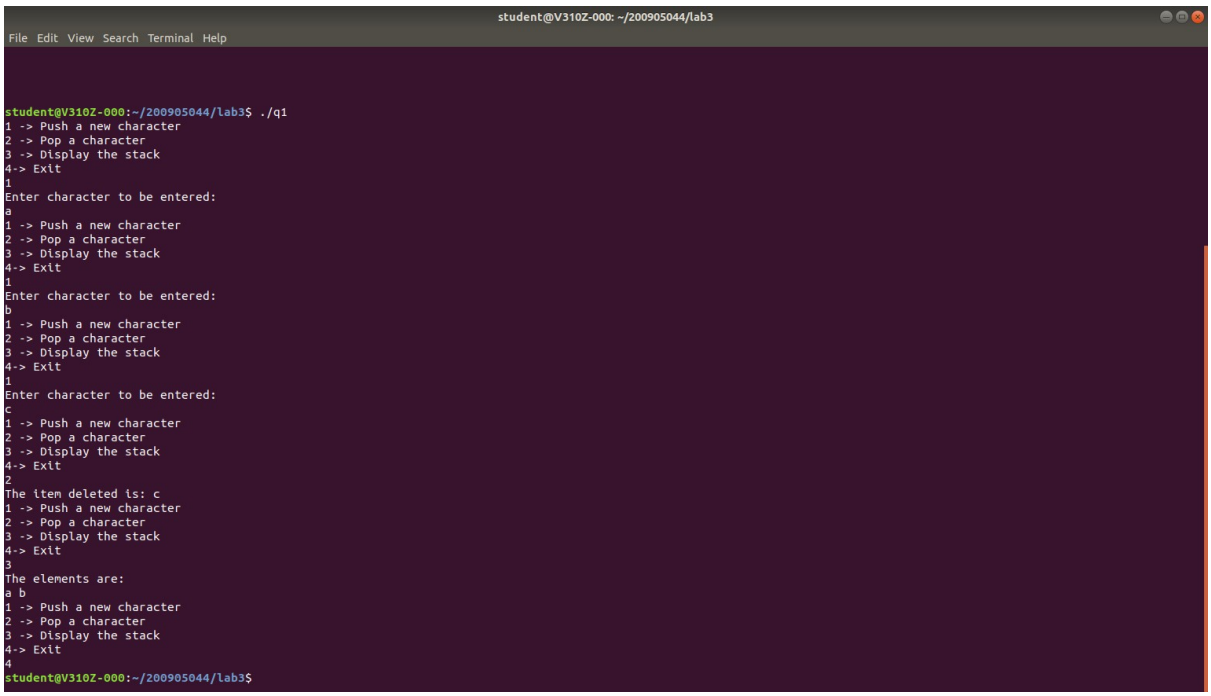
Swamiraju Satya Praveen Varma
200905044
Section B , B1
10

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 <stdlib.h>
#include <string.h>
#define MAX_ARRAY_SIZE 50

typedef struct
{
	char arr[MAX_ARRAY_SIZE];
	int top;
}stack;
void isEmpty(stack *s)
{
	if(s->top==-1)
	{
		printf("Error: Stack is empty.");
		exit(0);
	}
}
void isFull(stack *s)
{
	if(s->top==MAX_ARRAY_SIZE-1)
	{
		printf("Error: Stack is full.");
		exit(0);
	}
}
void push(stack *s, char item)
{
	isFull(s);
	s->top++;
```

```
	s->arr[s->top]=item;
}
char pop(stack *s)
{
	isEmpty(s);
	char temp;
	temp=s->arr[s->top--];
	return temp;
}
void display(stack *s)
{
	isEmpty(s);
	printf("The elements are: \n");
	for(int i=0;i<=s->top;i++)
	{
		printf("%c ",s->arr[i]);
	}
	printf("\n");
}

int main()
{
	stack s;
	s.top=-1;
	char item;
	int choice,x;
	while(1)
	{
		printf("1 -> Push a new character\n2 -> Pop a character\n3 -> Display the
stack\n4-> Exit\n");
		scanf("%d",&choice);
		switch(choice)
		{
			case 1:
			{
				printf("Enter character to be entered: \n");
				scanf("%d",&x);
				scanf("%c",&item);
				push(&s,item);
				break;
			}
			case 2:
			{
				char temp;
				temp=pop(&s);
				printf("The item deleted is: %c\n",temp);
				break;
```

```
                        }
                        case 3:
                        {
                                display(&s);
                                break;
                        }
                        case 0:
                        default:
                        {
                                exit(0);
                        }
                }
        }
        return 0;
}
```

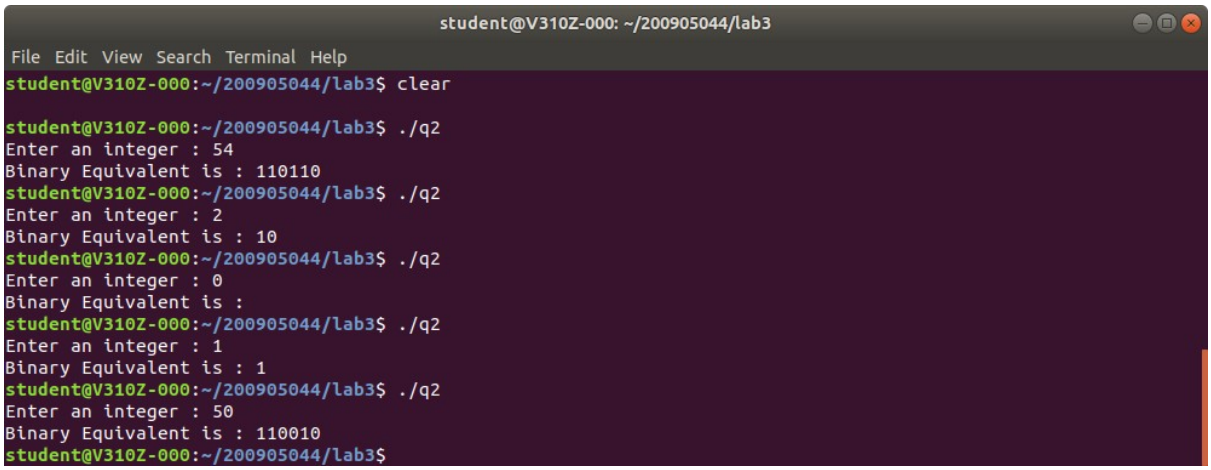


Q2: Convert a given decimal number to binary using stack.

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

int isEmpty(int top, int stack_arr[]);
void push(int x, int *top, int stack_arr[]);
```

```
int pop(int *top, int stack_arr[]);
void DecToBin(int num);

int main()
{
	int num;
	printf("Enter an integer : ");
	scanf("%d",&num);
	printf("Binary Equivalent is : ");
	DecToBin(num);

	return 0;

}/*End of main()*/

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


void push(int x, int *top, int stack_arr[])
{
	if(*top == (MAX-1))
		printf("Stack Overflow\n");
	else
	{
		*top=*top+1;
		stack_arr[*top] = x;
	}
}/*End of push()*/

int pop(int *top, int stack_arr[])
{
	int x;
	if(*top == -1)
	{
		printf("Stack Underflow\n");
		exit(1);
	}
	else
```

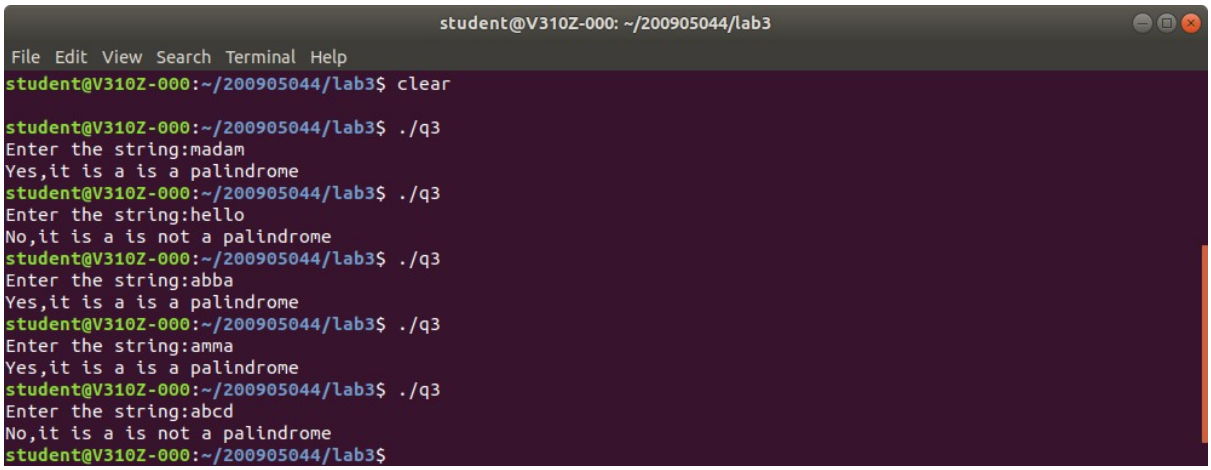
```
        {
                x = stack_arr[*top];
                *top=*top-1;
        }
        return x;
}/*End of pop()*/
```



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

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

char* stack;
int top = -1;

// push function
void push(char ele)
{
    stack[++top] = ele;
}

// pop function
char pop()
{
    return stack[top--];
}

// Function that returns 1
// if str is a palindrome
int isPalindrome(char str[])
{
    int length = strlen(str);

    // Allocating the memory for the stack
```

```
    stack = (char*)malloc(length * sizeof(char));

    // Finding the mid
    int i, mid = length / 2;

    for (i = 0; i < mid; i++) {
        push(str[i]);
    }

    // Checking if the length of the string
    // is odd, if odd then neglect the
    // middle character
    if (length % 2 != 0) {
        i++;
    }

    // While not the end of the string
    while (str[i] != '\0') {
        char ele = pop();

        // If the characters differ then the
        // given string is not a palindrome
        if (ele != str[i])
            return 0;
        i++;
    }

    return 1;
}


int main()
{
    char str[10];

    printf("Enter the string:");

    scanf("%s",str);

    if (isPalindrome(str)) {
        printf("Yes,it is a is a palindrome\n");
    }
    else {
        printf("No,it is not a palindrome\n");
    }

    return 0;
}
```


student@V310Z-000: ~/200905044/lab3
File Edit View Search Terminal Help
student@V310Z-000:~/200905044/lab3$ clear
student@V310Z-000:~/200905044/lab3$ ./q3
Enter the string:madam
Yes,it is a is a palindrome
student@V310Z-000:~/200905044/lab3$ ./q3
Enter the string:hello
No,it is a is not a palindrome
student@V310Z-000:~/200905044/lab3$ ./q3
Enter the string:abba
Yes,it is a is a palindrome
student@V310Z-000:~/200905044/lab3$ ./q3
Enter the string:amma
Yes,it is a is a palindrome
student@V310Z-000:~/200905044/lab3$ ./q3
Enter the string:abcd
No,it is a is not a palindrome
student@V310Z-000:~/200905044/lab3$

THANK YOU