DSA 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.

#include <stdio.h>

#include <stdlib.h>

#define maxSize 3

typedef struct {

int top;

char arr[maxSize];

}STACK;

int isFull(STACK\* s){

if(s->top==maxSize-1)

return 1;

return 0;

}

int isEmpty(STACK\* s){

if(s->top==-1)

return 1;

return 0;

}

void push(STACK\* s,char x){

s->arr[++s->top]=x;

}

char pop(STACK\* s){

return s->arr[s->top--];

}

void display(STACK\* s){

int k=s->top;

for(;k!=-1;)

printf("%c ",s->arr[k--]);

printf("\n");

}

int main()

{

STACK\* s;

char ch;

s->top=-1;

int opt=1;

while(opt!=4){

printf("Enter your choice:\n1.Push\n2.Pop\n3.Display\n4.Exit\n");

scanf("%d",&opt);

switch(opt){

case 1: if(isFull(s)){

printf("The stack is full\n");

break;

}

printf("Enter the char ");

scanf(" %c",&ch);

push(s,ch);

break;

case 2:if(isEmpty(s)){

printf("The stack is empty\n");

break;

}

printf("The popped value is %c\n",pop(s));

break;

case 3:if(isEmpty(s)){

printf("The stack is empty!\n");

break;

}

printf("The stack elements are ");

display(s);

break;

case 4:printf("Thankyou!\n");

exit(0);

default: printf("Invalid option!");

}

}

return 0;

}

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

#include <stdio.h>

#define maxSize 32

int stack[maxSize];

int top=-1,count=0;

int pop(int stack[]){

return stack[top--];

}

int push(int stack[],int x){

stack[++top]=x;

}

int main()

{

int d,count;

printf("Enter the decimal number ");

scanf("%d",&d);

for(count=0;d!=1;d/=2,count++)

push(stack,d%2);

printf("The binary form of %d is 1",d);

for(;count!=0;count--)

printf("%d",pop(stack));

return 0;

}

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

#include <stdio.h>

#define maxSize 32

char stack[maxSize];

int top=-1,count=0;

char pop(char stack[]){

return stack[top--];

}

void push(char stack[],char x){

stack[++top]=x;

}

int palindrome(char stack[],char s[]){

int i;

if(count%2==0)

i=count/2;

else i=count/2+1;

for(;i<count;i++)

if(s[i]!=pop(stack))

return 0;

return 1;

}

int main()

{

char s[100];

int d;

printf("Enter the string ");

scanf("%s",s);

while(s[count]!='\0')

count++;

for(int i=0;i<count/2;i++)

push(stack,s[i]);

if(palindrome(stack,s))

printf("%s is a palindrome",s);

else

printf("%s is not a palindrome",s);

return 0;

}