Lab No.3

//stack\_utils.h

#include <stdio.h>

#include <stdlib.h>

#define MAX (5)

#define TRUE (1)

#define FALSE (0)

#define SUCCESS (1)

#define FAILED (0)

typedef struct stack

{

char item[MAX];

int top;

}stack;

int isEmpty(stack\*);

int isFull(stack\*);

int push(stack\*, char);

char pop(stack\*);

void display(stack\*);

stack\* new\_stack();

int isEmpty(stack \*s)

{

if(s->top == -1) return TRUE;

return FALSE;

}

int isFull(stack \*s)

{

if(s->top == MAX - 1) return TRUE;

return FALSE;

}

int push(stack \*s, char elem)

{

if(isFull(s))

return FAILED;

s->item[++s->top] = elem; return SUCCESS;

}

char pop(stack \*s)

{

if(isEmpty(s)) return FAILED;

return(s->item[s->top--]);

}

void display(stack \*s)

{

if(isEmpty(s)) return;

int i;

for(i = 0; i <= s->top; i++)

printf("%c ", s->item[i]);

printf("\n");

}

stack\* new\_stack()

{

stack\* s = (stack \*)malloc(sizeof(stack)); s->top = -1;

return s;

}

Q1)

#include "stack\_utils.h"

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char const \*argv[])

{

int ch;

char elem;

stack\* s = new\_stack();

while(TRUE)

{

printf("\n1.Display Stack\n2.Push to stack\n3.Pop from stack\n4.Exit\nChoice: ");

scanf("%d", &ch);

if(ch == 4) break;

else if(ch == 1)

{

printf("Stack contents: \n");

display(s);

continue;

}

else if(ch == 2)

{

printf("Element to push: ");

scanf(" %c", &elem);

int pushed = push(s, elem);

if(pushed == SUCCESS)

printf("Pushed Succesfully!\n");

else

printf("Push Failed! Stack is full\n");

continue;

}

else if(ch == 3)

{

char popped = pop(s);

if(popped != FAILED)

printf("Popped element: %c\n", popped);

else

printf("Stack is empty!\n");

continue;

}

}

free(s);

return 0;

}

2)

#include <stdio.h>

#include <stdlib.h>

#include "stack\_utils.h"

void print\_bin(int);

void print\_bin(int dec)

{

stack\* rems = new\_stack();

while(dec != 0)

{

push(rems,dec%2 + '0'); dec = dec/2;

}

while(isEmpty(rems) == FALSE)

{

printf("%c", pop(rems));

}

}

int main(int argc, char const \*argv[])

{

int dec;

printf("Enter a decimal number (0-31): ");

scanf("%d", &dec);

if(dec > 31)

{

printf("Number requires greater than 5 bits\n");

printf("You can change the max allowed bits by changing the value of MAX defined

in stack\_utils.h\n");

return 0;

}

printf("Binary equivalent of the number: ");

print\_bin(dec);

printf("\n");

return 0;

}

3)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "stack\_utils.h"

int isPalindrome(char\*);

int isPalindrome(char\* str)

{

int i;

stack\* s = new\_stack();

for(i = 0; i < strlen(str); i++)

{

push(s, str[i]);

}

i = 0;

while((s->top <= i) == FALSE)

{

char elem = pop(s);

if(elem != str[i]) return FALSE;

i++;

}

return TRUE;

}

int main(int argc, char const \*argv[])

{

char\* str = (char \*)calloc(5, sizeof(char));

printf("Enter a string(max 5 characters): ");

scanf("%s", str);

if(isPalindrome(str))

printf("The entered string is a palindrome.\n");

else

printf("The entered string is not a palindrome.\n");

return 0;

}