**מעבדה 11 – מת"מ -   
מגישים:  
אורי מלכא – 314862996  
אלן ציפין - 313206062  
  
שאלה 1 מיין:  
  
שאלה 1 מימוש:**

#include "Lab11Header.h"

//implementation of the private functions

void free\_char(void\* elem) //function to free element

{

free(elem);

}

int main() {

PSTACKNODE stack = NULL;

char key,\*c;

puts("Enter characters, separated by space. Press enter to stop\n");

do

{

key = getchar();

\_flushall();

c = (char\*)malloc(sizeof(char));

\*c = key;

push(c, &stack);

} while (key != '\n');

puts("Here are the characters in reverse order:\n");

while (!isempty(stack))

{

printf("%c", \*(char\*)top(stack)); //casting to receive the specific data type

pop(&stack,free\_char); //we send a pointer to the function

}

return 0;

}

#include "Lab11Header.h"

//general structure deceleration (node in a one way linked list)

struct node

{

void\* data;

struct node\* next;

};

//implementation of all the general functions

void\* top(PSTACKNODE stack)

{

return (stack->data); //send the address

}

void push(void\* key, PSTACKNODE\* stack)

{

PSTACKNODE newnode;

newnode = (PSTACKNODE)malloc(sizeof(STACKNODE)); //create new node we assume that memo allocation succeeded

newnode->data = key;

newnode->next = (\*stack); //we head to the head

(\*stack) = newnode; //update the head

}

void pop(PSTACKNODE\* stack, void(\*free\_data)(void\*)) {

PSTACKNODE oldnode;

oldnode = (\*stack);

(\*stack) = (\*stack)->next;

free\_data(oldnode->data); //use the pointer to the function to free general memo.

free(oldnode);

oldnode = NULL;

}

BOOL isempty(PSTACKNODE stack)

{

return (stack == NULL) ? TRUE : FALSE; //if empty return 1 else 0

}

**שאלה 1 הדר:**

#ifndef \_Lab11Imp

#define \_Lab11Imp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

typedef struct node STACKNODE, \*PSTACKNODE; //pointer to a general structure

typedef enum{FALSE,TRUE}BOOL; //to check if is empty or not

//General functions decelerations

void push(void\* key, PSTACKNODE\* stack); // a function to add to the head of a one way linked list.

void pop(PSTACKNODE\* stack, void(\*free\_data)(void\*)); // a function to remove from the head of a one way linked list.

BOOL isempty(PSTACKNODE stack); // check if the "stack" or the list is empty.

void\* top(PSTACKNODE stack); // return the adress to the data value inside the head of the linked list

#endif

**שאלה 2:**

#include<stdio.h>

#define N 5

void Int\_Sum(void\* sum, void\* num)

{

\*(int\*)sum += \*(int\*)num;

}

void Int\_Sub(void\* sum, void\* num)

{

\*(int\*)sum -= \*(int\*)num;

}

void Init(void\* p)

{

\*(int\*)p = 0;

}

int Int\_Comp(void\* a, void\* b)

{

if (\*(int\*)a == \*(int\*)b)

return 1;

return 0;

}

int Do\_It(void\* sum, void\*\* array, void (\*p\_Init)(void\*), void(\*p\_Sum)(void\*, void\*), void(\*p\_Sub)(void\*, void\*), int(\*p\_Comp)(void\*, void\*))

{

int i;

p\_Init(sum); //initialize sum

for (i = 0; i < N; i++) {

if (p\_Comp(sum, array[i])) return 1; // we check if we have a match between the sum until this point with the next element value

(i % 2) ? p\_Sub(sum, array[i]) : p\_Sum(sum, array[i]); // if the index is odd we subtract else we add

}

return 0;

}

int main()

{

int num[] = { 5,8,23,20,23 }, i, answer;

void\* p\_num[N];

for (i = 0; i < N; i++) {

p\_num[i] = &num[i]; //we initialize the general pointers array to the specific array elements.

}

answer = Do\_It(&answer, p\_num, Init, Int\_Sum, Int\_Sub, Int\_Comp);

if (answer == 1)

printf("The answer is 'Yes'");

else

printf("The answer is 'No'");

return 0;

}

