**מעבדה 8**

**מגישים:**

טל יהושע, ת"ז 314774860

משה פרץ, ת"ז 318263977

**תרגיל מס' 1:**

**תוכנית:**

**קובץ Lab8c.C**

#include <stdio.h>

extern int PermutationArray(int arr1[], int arr2[], int size);

int main()

{

int arr1[] = { 1,4,7,9,9,0 };

int arr2[] = { 9,0,7,1,9,4 };

int arr3[] = { 1,4,4,9,7,0 };

if (PermutationArray(arr1, arr3, 6) == 1)

printf("\narr1 permutation of arr3 \n");

else

printf("\narr1 is not permutation of arr3 \n");

if (PermutationArray(arr1, arr2, 6) == 1)

printf("\narr1 permutation of arr2 \n");

else

printf("\narr1 is not permutation of arr2 \n");

return 0;

}

**קובץ Lab8q1.asm**

;lab8

.MODEL SMALL

.STACK 100H

.DATA

isPermutation DW 10 DUP(?)

.CODE

PUBLIC \_PermutationArray

\_PermutationArray PROC NEAR

PUSH BP ;preserve BP

MOV BP,SP

PUSH CX ;preserve CX

PUSH BX ;preserve BX

MOV AX,1 ;isEqual-> ax=1 else ax=0

MOV SI,[BP+4] ;getting the first array address

LEA DI,isPermutation ;Getting the isPermutation

MOV CX,10 ;the counter array length is 10

ZERO: ;resetting all the counter array to 0 for seconde use

MOV [DI],WORD PTR 0

ADD DI,2

LOOP ZERO

LEA DI,isPermutation

MOV BX,[BP+8] ;getting the first array size

MOV CX,BX

firstCount: ;counting the numbers in the first array

PUSH CX

MOV CX,2

REP ADD DI,WORD PTR [SI] ;Moving to the correct position in the counter array

INC WORD PTR [DI] ;adding 1 in the same location

REP SUB DI,[SI]

ADD SI,2 ;moving to the next position in the first array

POP CX

LOOP firstCount

MOV SI,[BP+6] ;getting the seconde array address

MOV CX,BX

secondCount: ;matching the numbers in the second array

PUSH CX

MOV CX,2

REP ADD DI,WORD PTR [SI] ;Moving to the correct position in the counter array

DEC WORD PTR [DI] ;Decreasing the equal numbers in the counter array

REP SUB DI,[SI]

ADD SI,2 ;moving to the next position in the second array

POP CX

LOOP secondCount

MOV CX,10

;checking if the the counter array equal to 0,if it is the arrays are premutation

isPermutationCheck:

CMP WORD PTR [DI],0 ; checking if the counter array

JE NEXT

MOV AX,0

JMP notPermutation

NEXT:

ADD DI,2

LOOP isPermutationCheck

notPermutation:

POP BX

POP CX

POP BP

RET

\_PermutationArray ENDP

END

**פלטים:**

