10010 – מבוא לתכנות מערכות

# תרגול 7: void\* + מצביעים לפונקציות + variadic functions

1. קבלת סכום ממחרוזת
   1. כתוב את הפונקציה getAsciiSum המקבלת מחרוזת ומחזירה את סכום ערכי האסקיי של התווים שבה.
   2. כתוב את הפונקציה הבאה:

int sum(const char\* str, int (\*getSum)(const char\*))

הפונקציה מקבלת מחרוזת ומצביע לפונקציה, ומפעילה את הפונקציה שהתקבלה על המחרוזת, ומחזירה את התוצאה.

* 1. קרא מה- main לפונקציה sum כאשר הפרמטר השני הוא הפונקציה getAsciiSum והדפס את התוצאה.
  2. קרא מה- main לפונקציה sum כאשר הפרמטר השני הוא הפונקציה strlen, המחזירה את סכום התווים במחרוזת, והדפס את התוצאה.

שאלה 2

כתוב פונקציה המקבלת מספר וזוגות של מערכי מספרים ואורכם. סיום הפרמטרים יהיה באמצעות NULL.

באמצעות qsort יש למיין את המערכים ובאמצעות bsearch הפונקציה תחזיר את כתובת המערך הראשון המכיל את המספר, או NULL אם אינו מופיע באף מערך.

ברגע שנמצא מערך מתאים, אין להמשיך למיין את שאר המערכים.

Starter לשאלה 2

#include <stdio.h>

#include <stdarg.h>

#include <stdlib.h>

void main()

{

int arr1[] = {1,37,74,13,51};

int arr2[] = {11,74,22,80};

int arr3[] = {21,84,16,35,79};

int\* arr;

int lookFor;

printf("Enter number to look for --> ");

scanf("%d", &lookFor);

}

שאלה 3

להלן קובץ header המכיל מבנה המחזיק נתונים עבור מערך שאיבריו מטיפוס void\*, וכן פונקציות הקשורות אליו:

#ifndef \_\_ARRAY\_H

#define \_\_ARRAY\_H

typedef struct

{

void\*\* arr;

int typeSize;

int logicalSize;

int phisicalSize;

void (\*print)(const void\*);

} Array;

void initArray(Array\* arr, int sizeType, int phisicalSize,

void (\*print)(const void\*));

void freeArray(Array\* arr);

void freeArrayAndData(Array\* arr);

void addToEnd(Array\* arr, void\* newValue);

void printArr(const Array\* arr);

#endif // \_\_ARRAY\_H

עליכם לממש את קובץ ה- C כך שכל המימושים יהיו לטיפוס כללי כלשהו (void\*).

להלן main לדוגמא:

#include <stdio.h>

#include <stdlib.h>

#include "array.h"

void printInt(const void\* obj)

{

printf("%d ", \*(int\*)obj);

}

void main()

{

Array intArr;

int n1=3, n2=8, n3=5;

initArray(&intArr, sizeof(int), 2, printInt);

addToEnd(&intArr, &n1);

addToEnd(&intArr, &n2);

addToEnd(&intArr, &n3);

printArr(&intArr);

freeArray(&intArr);

}

/\*

There are 3 values (4 spaces):

3 8 5

Press any key to continue . . .

\*/

פתרון שאלה 3

#include <stdio.h>

#include <stdlib.h>

#include "array.h"

void initArray(Array\* arr, int sizeType, int phisicalSize,

void (\*print)(const void\*))

{

arr->logicalSize = 0;

arr->phisicalSize = phisicalSize;

arr->typeSize = sizeType;

arr->print = print;

}

void freeArray(Array\* arr)

{

free(arr->arr);

}

void freeArrayAndData(Array\* arr)

{

int i;

for (i=0 ; i < arr->logicalSize ; i++)

free(arr->arr[i]);

free(arr->arr);

}

void addToEnd(Array\* arr, void\* newValue)

{

if (arr->logicalSize == arr->phisicalSize)

{

printf("double arr size...\n");

arr->phisicalSize \*= 2;

arr->arr = (void\*\*)realloc(arr->arr, arr->phisicalSize\*sizeof(void\*));

}

arr->arr[arr->logicalSize++] = newValue;

}

void printArr(const Array\* arr)

{

int i;

printf("There are %d values (%d spaces):\n", arr->logicalSize, arr->phisicalSize);

for (i=0 ; i < arr->logicalSize ; i++)

arr->print(arr->arr[i]);

printf("\n");

}