

## תרגיל בית 2 (שפת C)

בתרגיל זה אין להשתמש בהקצאות זיכרון דינמיות.

### הנחיות כלליות לפיתרון התרגילים בקורס:

- התרגילים הם לעבודה ביחידים. מותר להתייעץ אך ורק בעל פה, אסור בתכלית האיסור שחומר כתוב/מודפס/אלקטרוני יעבור בין אנשים. בנוסף, על חלק מהתרגילים תיבחנו פרונטלית ועליכם להבין כל דבר בקוד!
- המנעו ממספרי קסם: מספרים שמופיעים באמצע הקוד בלי משמעות מיוחדת (לדוגמה נניח שמספר הרשומות בתרגיל אחר הוא מקסימום 50 ואז בכל מקום בקוד כתוב 50. לעומת זאת, 0 לתחילת מערך לא נחשב מספר קסם - הפעילו הגיון בריא) והשתמשו במקום זאת בפקודות מאקרו (`#define` או אם כבר למדתם על כך ב `const`).
- אין להשתמש ב-`variable length arrays`, וכן במשתנים סטאטיים ו / או גלובליים.
- כל התרגילים בקורס צריכים להתקמפל ולרוץ באתר `c9.io` (האתר מריץ מערכת הפעלה אובונטו) עם שורת הקמפול:

```
gcc -Wall -Wvla -Werror -g ...
```

עבור תרגילי C ושורת הקמפול:

```
g++ -Wall -Wvla -Werror -g -D_GLIBCXX_DEBUG -std=c++11 ...
```

עבור תרגילי C++

או במידה ומצורף Makefile עם ה `Makefile` המצורף

- יש להקפיד על סגנון תכנות טוב כמו שלמדתם. לדוגמה, להימנע מחזרות קוד (לכתוב פונקציות שצריך), שמות משתנים עם משמעות, בהירות הקוד, להקפיד להשתמש בקבועים שצריך ולא במספרי קסם וכו'.
- עליכם להגיש קובץ ששמו מספר ת.ז. שלכם כמו שהיא מופיעה באתר המודל נקודה `.zip`. לדוגמה, אם מספר ת.ז. שלי הוא 12345678 אז שם הקובץ יהיה:

12345678.zip

### שאלה 1

ענו בקצרה בקובץ ה-`README` על השאלות הבאות:

1. מה ידפיס הקוד הבא? מדוע? הסבירו תשובתכם.

```
#include <stdio.h>
```

```
void PrintSizeOfArray(int a[])
{
    printf("%zu\n", sizeof(a));
    ++a[2];
}
```

```
int main()
{
    int it;
    int a[5] = {0};
    PrintSizeOfArray(a);
    for (it = 0 ; it < sizeof(a)/sizeof(int) ; ++it)
        printf("%d\n", a[it]);
    return 0;
}
```

```
}
```

2. שינינו את הקוד מהסעיף הקודם כך שהמערך a 'נעטף' בתוך struct. התקבל הקוד הבא:

```
#include <stdio.h>

typedef struct
{
    int a[5];
} ArrStruct;

void PrintSizeOfArray(ArrStruct arrStruct)
{
    printf("%zu\n", sizeof(arrStruct.a));
    ++arrStruct.a[2];
}

int main()
{
    int it, sizeA;
    ArrStruct arrStruct;
    sizeA = sizeof(arrStruct.a)/sizeof(int);
    for (it = 0 ; it < sizeA ; ++it)
        arrStruct.a[it] = 0;
    PrintSizeOfArray(arrStruct);
    for (it = 0 ; it < sizeA ; ++it)
        printf("%d\n", arrStruct.a[it]);
    return 0;
}
```

מה יודפס כעת? הסבירו.

במידה והמערך היה מאד גדול, מי משני המימושים יעיל יותר מבחינת זמן ריצה? מדוע?

## שאלה 2

בקובץ MyString.h שנמצא באתר מצויות חתימות של שתי פונקציות המתוארות בסעיפים הבאים. ממשו את הפונקציות בקובץ בשם MyString.c.

**שימו לב:** בשאלה זו אין להיעזר בפונקציות מוכנות מהספרייה string.h או מכל ספרייה סטנדרטית אחרת. שימוש בפונקציות אלו (אפילו שימוש בפונקציה strlen()) יביא לפסילת התשובה.

סעיף א:

ממשו את הפונקציה בעלת החתימה הבאה

```
int extendedSubStr(int isCyclic, int step, const char* str1, const char* str2);
```

הפונקציה תבצע חיפוש של תת המחרוזת str2 בתוך המחרוזת str1. אם נמצאה תת מחרוזת כזו, הפונקציה תחזיר את האינדקס בו מתחילה תת המחרוזת; אחרת יוחזר -1.

אופן החיפוש של הפונקציה ייקבע על ידי הפרמטרים isCyclic ו-step.

· אם הערך של isCyclic הוא אפס, יתבצע חיפוש תת מחרוזת עד סוף str1.

• אם הערך שונה מאפס נתייחס ל-str1 כאל מחרוזת ציקלית. במקרה זה יתכן כי המחרוזות str2 ארוכה יותר מ-str1, לדוגמא:

```
const char* str1 = "abc";  
const char* str2 = "abca";
```

• הפרמטר step מורה לפונקציה באיזה מספר דילוגים לחפש את תת המחרוזות. הוא עשוי לקבל ערכים חיוביים או שליליים, אך לא אפס. לדוגמא

```
const char* str1 = "t3e2s1tcba";  
const char* str2 = "test";
```

במקרה זה str2 לא תימצא אם step==1, אבל תימצא אם step==2. כאשר הערך של step שלילי יש להחזיר -1

#### הנחות והערות:

- הפרמטרים step ו-isCyclic יכולים להופיע בכל הצירופים; למשל מחרוזת ציקלית עם חיפוש בצעדים של 3.
- חיפוש מחרוזת ריקה str2, או חיפוש בתוך מחרוזת ריקה str1, יחזירו -1.
- שימו לב כי בחיפוש ציקלי עשויים להתבצע על המחרוזות מספר סבבים.
- החיפוש הוא case sensitive, כלומר מבדילים בין אותיות גדולות לקטנות.
- הניחו כי הפונקציה מקבלת קלט חוקי.

סעיף ב:

ממשו את הפונקציה בעלת החתימה הבאה:

```
void sortDelim(char str[], const char* delim);
```

הפונקציה מקבלת שני פרמטרים: מחרוזת str המורכבת ממספר מילים וביניהן תוים מפרידים שונים, ורשימת תוים מפרידים בפרמטר delim. כל אחד מן התוים ב-delim יכול להפריד בין מילים במחרוזת.

משימתכם היא למיין את מחרוזת הקלט str, כך שלאחר המיון יופיעו המילים מופרדות על ידי התו '!'; לדוגמא:

```
char str[] = "aaa*test,hello.world*abcd.zzz";  
const char* delim = ",.*";
```

לאחר הקריאה

```
sortDelim(str,delim);
```

ערך המחרוזות str צריך להיות

```
"aaa;abcd;hello;test;world;zzz"
```

הנחיות והערות:

- המחרוזות מכילה רצף מילים מופרדות. לא נתון מראש כמה מילים יש במחרוזת.
- המיון צריך להיות לקסיקוגרפי ולפי קוד Ascii. בהינתן שתי מחרוזות, על השוואת המילים שלכם להתנהג כמו הפונקציה strcmp (אך כזכור אסור לכם להשתמש בפונקציה strcmp).
- מותר למיין את המערך בסיבוכיות ריבועית במספר המילים.
- המחרוזות אינה מכילה מילים ריקות (כלומר מספר תוים מפרידים רצופים).

בנוסף לקבצים MyString.c, MyString.h, README.txt עליכם להגיש גם קובץ MyStringTest.cpp (שימו לב שזה הוא קובץ cpp ולא קובץ c, ראו absdiff.zip באתר כדוגמא) אשר יכיל בדיקה של התרגיל ע"י ספריית Google Unit Test.

יש להשתמש בקובץ makefile שנמצא באתר.

בהצלחה