תרגיל 4 – מצביעים והקצאת זיכרון דינאמי

הגשה עד 9/1/17 בשעה 23:50

הוראות ההגשה ודגשים מיוחדים מופיעים בסוף התרגיל! חובה לקרוא ולפעול לפיהן.

התרגיל מורכב מחלק תאורטי, חלק מעשי וחלק הבדיקות .

חלק א' (תאורטי) יש להגיש כקובץ וורד.

חלק ב' (מעשי) יש להגיש בקובץ .C אחד (בחלק המעשי מדובר בתוכנית אחת (!) - פונקציה main אחת – המפעילה הרבה פונקציות אחרות!)

חלק ג' (הבדיקות) יש להגיש בקובץ .C נוסף.

לכווץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.

חלק א' – תאורטי (המענה בקובץ טקסט וורד בלבד!) – סה"כ 14 נקודות

שאלה 1 – שאלות כללית – 9 נקודות

א. השלימו את הקוד (**1 נקודה**):

איזה הצהרה יש להוסיף לתוכנית הבאה במקום המסומן על מנת לוודא שהיא תדפיס למסך: "Hello World" בזמן ההרצה שלה?

```
#include<stdio.h>
int main() {
    char str[] = "Hello World";
    char temp[25];
    char *ps, *pt;
    ps = str;
    pt = temp;
    while(*ps)
        *pt++ = *ps++;

    /* Add a statement here */
    printf("%s\n", temp);
    return 0;
}
```

ג. נתון מערך מצביעים למחרוזות (**8 נקודות**) :

char* arrPoint[]={"father","mother","sister","brother",NULL}

char** ppArr= arrPoint;

ומצביע לתחילתו ppArr . בעזרת מצביע ppArr איך ניתן להדפס את כל המחרוזות:

- על פי התבנית מחרוזת %s .
- על פי התבנית התווים %c.

הפלט בשני המקרים אמור להיראות כך:

father mother sister brother

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלפון

שאלה 2 – מה הפלט? - 3 נקודות

א. בהנחה שהתא הראשון במערך נשמר בכתובת 1000 בזיכרון, מה יהיה פלט הקטע הבא:

```
#include<stdio.h>
int main(){
    int a[3][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
    printf("%u, %u, %u\n", a[0]+1, *(a[0]+1), *(*(a+0)+1));
    return 0;
}
```

שאלה 3 – מהי השגיאה? - 2 נקודות

בכל סעיף הסבירו היכן נמצאת השגיאה ואיך (אם בכלל) ניתן לתקנה.

א.

```
#include<stdio.h>
int main() {
    int *p;
    *p=100;
    return 0;
}
```

ב.

```
#include<stdio.h>
int main() {
    int array[] = {10, 20, 30, 40, 50};
    int j;
    for(j=0; j<5; j++) {
        printf("%d\n", array);
        array++;
    }
    return 0;
}
```

חלק ב' – מעשי (המענה בקובץ .C אחד)- סה"כ 81 נקודות

המערכת:

בכניסה למערכת התוכנית תציג למשתמש את התפריט **(2 נקודות)** הבא :

1 - Reduce String

2 - UpdateString

3 - Dynamic1DArray

4 - Dynamic2DArray

5 - Exit

ההרצה תעבוד עד לבחירת אופציה 5 , כאשר התוכנית תסתיים ותדפיס את ההודעה הבאה :

Have a nice day !

סעיף 1 – ניהול מחרוזת דינאמית (צימצום מחרוזת) – סה"כ 21 נקודות

על מנת להפעיל את סעיף זה, על המשתמש יהיה להקליד בפונקציה הראשית 1.

כתוב פונקציה ( )***BuildString*** * char הקולטת מחרוזת מהמשתמש (מחרוזת זו תהיה לכל היותר 80 תווים באורכה) ומייצרת מערך דינמי של תווים בגודל המדויק לקבלת המחרוזת ומעתיקה אותה לתוך המערך שנוצר, הפונקציה מחזירה את כתובת המערך.

כתוב פונקציה (char** str)***FreeString*** void המקבלת כתובת של המצביע למערך דינאמי ומשחררת את הזיכרון שהמערך תופס בזיכרון.

כתוב פונקציה (char * st1, char * st2)***ReduceString*** * char שמקבלת את שני המצביעים למחרוזות שנוצרו ע"י הפעלת הפונקציה ***BuildString*** ומוחקת את כל המופעים (במידה וישנם) של המחרוזת st1 בתוך המחרוזת st2. הפונקציה מייצרת מערך דינמי חדש st3 המתאים למחרוזת החדשה ומחזירה את הכתובת שלו.

הפונקציה לא משנה את המחרוזות st1, st2.

להלן מספר דוגמאות :

מחרוזת הראשונה : abcd

מחרוזת השנייה:Xyabczewrrrabcdkllabx

מחרוזת החדשה :Xyabczewrrrkllabx (Xyabczewrrr~~abcd~~kllabx:)

מחרוזת הראשונה : aaa

מחרוזת השנייה:aaaaaxaaaaaaayaaaaa

מחרוזת החדשה : aaxayaa (~~aaa~~aax~~aaaaaaa~~ay~~aaa~~aa)

מחרוזת הראשונה : ababab

מחרוזת השנייה:xyzabababababz

מחרוזת החדשה : xyzababz (xyz~~ababab~~ababz)

בהפעלת האופציה הזו בתפריט, המשתמש יתבקש להזין את תוכן שתי המחרוזות ובהתאם לכך התוכנית תבנה אותן ותפעיל את הפונקציה ReduceString , הפונקציה תחזיר את המחרוזת החדשה והפונקציה הראשית בתוכנית תדפיס אותה, לאחר מכן התוכנית תשחרר את כל הזיכרון הדינאמי שנתפס.

סעיף 2 – ניהול מחרוזת דינאמית (עידכון מחרוזת)- סה"כ 16 נקודות

על מנת להפעיל את סעיף זה, על המשתמש יהיה להקליד בפונקציה הראשית 2.

כתוב פונקציה (char * str)***ChangeString*** * char שמקבלת מחרוזת המורכבת ממספרים, אותיות קטנות ורווחים. המספרים חיוביים ומורכבים מספרה אחת או יותר . הפונקציה תיצור ותחזיר את המחרוזת החדשה אשר נדרש שתהיה מורכבת רק מאותם המספרים עם השינוי הבא :

- ✓ הספרה 2 תוחלף ב22, הספרה 3 תוחלף ב333 וכו.
- ✓ הספרה 0 לא תופיע כלל.

- ✓ בין מספר למספר יופיע סימן מינוס (-) לכל רצף של רווחים , וכוכבית (*) לכל רצף של אותיות קטנות.

לדוגמה ,

הפונקציה תקבל מחרוזת הבאה:

42 qqq231 hh 425 abc 1023

הפונקציה תחזיר:

444422-*223331-*-44442255555-*-122333

בהפעלת אופציה זו, לאחר הזנת הקלט הנדרש (=המחרוזת) על-ידי המשתמש (מחרוזת זו תהיה לכל היותר 80 תווים באורכה), התוכנית תיצור את המחרוזת ע"י פונקציה המתאימה לכך והתוכנית תפעיל את הפונקציה ChangeString, הפונקציה תחזיר את המחרוזת החדשה והפונקציה הראשית תדפיס אותה, בסוף הפעולה תשחרר את כל הזיכרון הדנאמי (ניתן להשתמש בפונקציות שהגדרתם קודם).

הפונקציה לא משנה את הפרמטר שלה.

סעיף 3 – ניהול מערך חד-ממדי דינאמי- סה"כ 21 נקודות

על מנת להפעיל סעיף זה המשתמש יצטרך להקליד בפונקציה הראשית 3.

כתוב פונקציה void ***Build1DArray***(int** arr, int* count) הקולטת מהמשתמש את גודל המערך (כמות האיברים), יוצרת מערך דינאמי, קולטת לתוכו את הערכים ומחזירה את המערך.

כתוב פונקציה void ***Print1DArray***(int* arr, int size) המקבלת כפרמטר מערך, כולל הגודל שלו, ומדפיסה את אברי המערך בפורמט הבא:

{ 3 , 1 , 4 , 5 }

מערך ריק יודפס:

{ }

כתוב פונקציה void ***Reduce1DArray***(int** arr, int* count) המקבלת מצביע למערך (ע"י מצביע למצביע) ואת גודל המערך (כמות האיברים, ע"י המצביע) . הפונקציה תצמצם את המערך כך כדי לא יהיו בו ערכים חוזרים ותחזיר את המערך המעודכן .

לאחר בחירת האופציה בתפריט התוכנית תבקש להכניס את כמות האיברים , תקצה זיכרון בהתאם, תקבל את כל הערכים מהמשתמש (תמלא את הנתונים של המערך) , תדפיס אותו , לאחר מכן תשלח את המערך לפונקציה צימצום, תדפיס את המערך המעודכן ואז תשחרר את כל הזיכרון.

סעיף 4 – ניהול מערך דו-ממדי דינאמי – סה"כ 21 נקודות

על מנת להפעיל סעיף זה המשתמש יצטרך להקליד בפונקציה הראשית 4.

כתוב פונקציה void ***BuildMatrix***(int*** matrix,int row,int column). הפונקציה תקבל מצביע למערך דו-מימדי דינאמי, כמות השורות וכמות העמודות. הפונקציה תקצה זיכרון בהתאם, ותמלא את המטריצה בערכים.

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהנייד

כתוב פונקציה void ***FreeMatrix***(int*** matrix,int row). הפונקציה תקבל מצביע למערך דו-מימדי דינאמי וכמות השורות, ותשחרר את כל הזיכרון שהמטריצה תפסה.

כתוב פונקציה void ***PrintMatrix***(int** matrix,int row,int column). הפונקציה תקבל מערך דו-מימדי דינאמי, כמות השורות וכמות העמודות. הפונקציה תדפיס את איברי המטריצה בתצוגת טבלת דו-ממדית .

למשל,

1 2 4 7

8 7 6 1

כתוב פונקציה void ***Transpose***(int*** matrix,int* row,int* column) המשחלפת את המטריצה כך שכל עמודה תהפוך לשורה וההיפך , הפונקציה תשנה את המערך הקיים ותחזיר את המטריצה החדשה.

שימו לב שיש ליצור מערך חדש ולשחרר את הזיכרון של המערך הישן .

למשל:

1	**2**	**3**	→	**1**	**4**
4	**5**	**6**		**2**	**5**
				3	**6**

כתוב פונקציה void ***RemoveColumn*** (int** matrix,int row,int column ,int colNumber), המקבלת את מערך דו-מימדי דינאמי, כמות השורות ,כמות העמודות ומספר העמודה ומורידה אותה מהמטריצה הקיימת. במידה ומספר העמודה אינו תואם את ממדי המטריצה, תוצג הודעה מתאימה והפונקציה תסתיים.

לאחר בחירת האופציה בתפריט התוכנית תבקש להכניס ערכים לשני מימדי המטריצה, תקצה זיכרון בהתאם תקבל את כל הערכים מהמשתמש (תמלא את הנתונים של המטריצה) , תדפיס אותה , לאחר מכן תשלח את המטריצה לפונקצית שיחלוף , תדפיס את המטריצה המעודכנת , תבקש מהמשתמש את מספר העמודה , תוריד אותה מהמטריצה , תדפיס את המטריצה שוב ואז תשחרר את כל הזיכרון.

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלאפון

חלק ג' – מעשי unit test (המענה בקובץ .C נפרד) – סה"כ 15 נקודות

סעיף 1 Unit test- 10 נקודות

ע"מ לבדוק הקצאה \ שחרור זכרון, יש להשתמש ב- Best Practices מבחינת הקצאת הזכרון:

```
char* pointer = NULL; /* Best Practice, point to null when not assigned /
allocated */

/* do stuff */

pointer = (char *)malloc(1024);/* malloc does not always work, check it. */

if(pointer == NULL) {
/*Help, warn or exit*/
}

/* do stuff */

if(pointer) {
    free(pointer);
    pointer = NULL; /* Best Practice, point to null when not assigned /
allocated */
}

/* do stuff */

if(pointer) {
/* tested memory allocation*/
}
```

לחלק זה יש לכתוב 4 בדיקות:

1. בדיקת הקצאת זכרון: "unit_test_malloc"(3 נקודות)
2. בדיקת שחרור זכרון: "unit_test_free" (3 נקודות)
3. בדיקת פונק' ReduceString לפי הדוגמא הראשונה: "unit_testReduce1" (2 נקודות):
   מחרוזת הראשונה : abcd
   מחרוזת השנייה:Xyabczewrrrabcdkllabx
   מחרוזת החדשה :Xyabczewrrrkllabx (:Xyabczewrrr~~abcd~~kllabx)
4. בדיקת פונק' ReduceString לפי הדוגמא השניה: "unit_testReduce2" (2 נקודות):
   מחרוזת הראשונה : aaa
   מחרוזת השנייה:aaaaaxaaaaaaayaaaaa
   מחרוזת החדשה : aaxayaa (~~aaa~~aax~~aaaaaa~~ay~~aaa~~aa)

הנחיות כלליות לבדיקות יחידה:

- יש להשתמש בספרית MinUnit לביצוע בדיקות היחידה (http://www.jera.com/techinfo/jtns/jtn002.html)
- יש להציג את הנתונים הנשלחים לפונק' בבדיקה.

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהנייד

סעיף 2 Unit test- 5 נקודות

לחלק זה יש לכתוב יש לכתוב בדיקת יחידה: "unit_test_Change":
הפונקציה תקבל מחרוזת הבאה:
42 qqq231 hh 425 abc 1023
הפונקציה תחזיר:
444422-*223331-*-44442255555-*-122333

הנחיות כלליות:

- יש להשתמש בספרית MinUnit לביצוע בדיקות היחידה (http://www.jera.com/techinfo/jtns/jtn002.html)
- יש להציג את הנתונים הנשלחים לפונק' בבדיקה.

כתוב פונקציה RunUnitTests אשר נקראת מתוך ה-main ומבצעת את כל הבדיקות הדרושות.

סעיף בונוס (מסלול סייבר) – (10 נקודות)

יש לממש את הפונקציה ***Reduce1DArray*** באופן ***גנרי***. הפונקציה מקבלת מערכים מאחד מהטיפוסים הבאים, ומבצעת צמצום שלהם תוך שימוש במצביעים לפונקציות. הטיפוסים האפשריים הם: int, char, float. יש ליצור לאפשרות בחירה חדשה בתפריט ואז להפעיל את הפונקציה המדוברת.

הערות:

1. **אין להשתמש בתרגיל בחומר שטרם נלמד, או שנלמד לאחר נושא התרגיל אלא אם נכתב במפורש בתרגיל שמותר (מלבד סעיף סייבר).**
2. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום שצריך להכניס מספר שלם – נכניס מספר שלם (ולא שבר או אות). אנחנו לא מתחייבים שהוא יהיה חיובי או א-שלילי, או בטווח מסוים! – אלא אם כן נאמר אחרת בשאלה עצמה.
3. בכל פעם שהמשתמש מקליד קלט שגוי התוכנית מבקשת קלט חוזר.
4. אחרי כל הדפסה יש לבצע ירידת שורה.
5. בתרגיל יש להשתמש בספריות stdlib-ו,malloc, stdio בלבד!
6. יש להקפיד על תכנות נכון:
   a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: define, const או enum, בהתאם לצורך.
   b. יש לרשום הערות .
   c. יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!
   d. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן.
   e. לפני בקשת קלט (scanf) יש להדפיס למשתמש הוראה (printf) איזה קלט מבוקש.
   f. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו').
   g. יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה.
7. בהצלחה ☺

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלפון

הנחיות הגשה

- תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!
- חלק א' יש להגיש כקובץ וורד , חלק ב' (מעשי) יש להגיש בקובץ .C אחד ,חלק ג' (הבדיקות) יש להגיש בקובץ .C נוסף. לכווץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
- ההגשה היא אך ורק דרך המערכת!
- **שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית, סבטלנה, במייל: sceassign2016@gmail.com**
- **שאלות לגבי unit test (בדיקת יחידה) יש לשלוח אך ורק לדוד בן ישי במייל:**
- **davidbe4@ac.sce.ac.il**
- כל הסעיפים נכתבים כחלק של אותה תוכנית וזה מצריך פונקציה ראשית אחת בלבד. דאגו לתעד היטב את הקוד שלכם, כך שיהיה ברור מה מבצע כל חלק!
- בתחילת הקובץ יש להוסיף את התיעוד הבא:

```
/* Assignment: 2
        Campus: Ashdod / Beer Sheva (תבחרו את המתאים)
        Author: Israel Israeli, ID: 01234567
*/
```

כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.

- הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומחלה חריפה!) ובצרוף אישורים מתאימים. כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה לפחות יומיים לפני חלוף הדד-ליין!
- ההגשה היא עד התאריך האחרון לתרגיל: 9/1/17 יום ב' בשעה **23:50**. הגשה מאוחרת אפילו בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את זמנכם בהתאם!

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלאפון