פקולטה: מדעי הטבע

מחלקה: מדעי המחשב

שם הקורס: מבוא למחשבים ושפת C

קוד הקורס: 2-7028510

תאריך בחינה: 15.2.2017

משך הבחינה: שעתיים

שם המרצה: ד"ר אופיר פלא

חומר עזר: פתוח

שימוש במחשבון: לא

<u>הוראות כלליות:</u>

- הניסוח הוא בלשון זכר מטעמי נוחות ומתייחס לכולם/ן!
- אין בחירה במבחן. יש לענות על כל השאלות.
- ניתן להסתמך על כל סעיף במבחן גם אם לא פתרתם אותו על מנת לפתור סעיף אחר במבחן.
- ניתן לרשום "לא יודע/ת" על סעיף ולזכות ב 20% מהנקודות המוקנות לסעיף הספציפי.
- במידה ולסעיף ניתנה תשובה ובנוסף נרשם לגבי הסעיף "לא יודע/ת" אזי הניקוד שיינתן לסעיף יהיה 0 מבלי שהתשובה תיקרא.
- אם לא רשום דבר בסעיף או שזה כלל לא נמצא ההנחה היא שנרשם "לא יודע/ת" עבור אותו סעיף.
- יורדו נקודות על פתרון נכון שאינו אופטימלי בהתייחס לנלמד בקורס.
- לכל אורך הבחינה הניחו כי:

sizeof (char) = 1
sizeof (int) = 4
sizeof (double) = 8
sizeof (void *) = 8
sizeof (size_t) = 8

<u>שאלה 1 (15 נק')</u>

סטודנט למד כי ניתן לייעל גישה לאיברי מערך דו-מימדי, על ידי גישה חד מימדית. לצורך כך הוא כתב את הפונקציה ותכנית הבדיקה הבאה.

```
#include <stdio.h>

#define COLS 5
#define ROWS 3

int GetNthItem (int arr[ROWS][COLS], size_t n)
{
    return *((int*)(arr+n));
}

int main ()
{
    int arr[ROWS][COLS] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
    printf ("%d\n", GetNthItem(arr,2));
    return 0;
}
```

א. [10 נק'] מה יהיה הפלט של ריצת התכנית, נמקו.

ב. [5 נק'] תקנו את הפונקציה GetNthElement כך שהיא תעבוד היטב – תחזיר את האיבר ה-n מתחילת המערך בזיכרון, כאילו המערך היה חד מימדי. כלומר, ה-output של התוכנית צריך להיות 3. (אין לשנות את הפרוטוטייפ של הפונקציה).

<u>שאלה 2 (15 נק')</u>

מתכנת כתב מקרו בכדי ליעל את פעולת ההחלפה של מחרוזות אחת בשנייה. לצורך בדיקת הרעיון שלו הוא עשה שימוש במקרו בתכנית הבדיקה הבאה:

```
#include <stdio.h>

#define STR_SWAP(a,b) {char* t=a; a=b; b=t;}
int main ()
{

    char str1 [] = "B";
    char str2 [] = "A";
```

```
    printf ("%s %s\n", str1, str2);
    STR_SWAP (str1,str2);
    printf ("%s %s\n", str1, str2);
    return 0;
}
```

א. [5 נק'] להפתעתו, התקבלה שגיאת קומפילציה, הסבירו מדוע.

ב. [10 נק'] בכדי להתגבר על הבעיה הוא החליף את המקרו בפונקציה הבאה:

```
void STR_SWAP (char a[], char b[])
{
    char *t=a;
    a=b;
    b=t;
}
```

מה יהיה הפלט של קימפול והרצת התכנית (במידה והקימפול נכשל, אין צורך לפרט מהו פלט הריצה, ובמידה והקימפול מצליח יש לפרט במדוייק מהו הפלט של ריצת התכנית). הסבר.

<u>שאלה 3 (20 נק')</u>

בהנחה ש sizeof שלstruct שווה לסכום מרכיביו, ובהתייחס לקטע הקוד הבא:

```
1. #include <stdlib.h>
2. struct B
3. {
4.     int *_arr;
5. };
6. struct A
7. {
8.     struct B *_pb;
9. };
10.
11.
12.   int main()
13.   {
14.       struct A a;
15.       struct B *pb = (struct B*)malloc(sizeof (struct B));
16.
17.       int arr[5] = {1,2,3,4,5};
18.       pb->_arr = arr;
19.       a._pb = pb;
20.
21.
22.       struct A *pa = (struct A*)malloc(sizeof (struct A));
23.       *pa = a;
24.
```

```
25.
26.       //<free code>
27.
28.       return 0;
29.   }
```

א. [16 נק'] הטבלה הבאה מכילה כתובות. עליכם לציין לכל כתובת את המיקום שלה בזיכרון (בהתייחס לשמם ולשלב הריצה כאשר התוכנית סיימה לבצע את מספר השורה שבסוגריים): מחסנית, גלובלי, ערימה דינמית, איזור הקוד, לא מוגדר. לדוגמא הכתובת &a לאחר שורה 14 היא כתובת במחסנית. הניחו שכל ההקצאות הדינמיות מצליחות.

כתובת	מיקום
a._pb (14)	
&pb (15)	
pb (15)	
&(pb->_arr[3]) (15)	
&(pb->_arr[3]) (18)	
arr+3 (18)	
&pa (23)	
pa->_pb (23)	

ב. [4 נק'] הוסיפו קוד לשחרור כל הזכרון בתכנית.

<u>שאלה 4 (50 נק')</u>

א. (30 נק') בשאלה זו יש לממש את הפונקציה swap_slow_sort אשר חתימתה ותיעודה זהים לפונקציה qsort מהספרייה הסטנדרטית שלמדנו בכיתה. השוני הוא במימוש. בשאלה זו אתם מתבקשים לממש מיון אשר מחליף רנדומית כל פעם זוג איברים במערך עד שהמערך הופך לממויין.

חתימת הפונקציה:

```
void swap_slow_sort(void *base, size_t nmemb, size_t size,
                    int (*compar)(const void *, const void *));
```

תיעוד הפונקציה:

```
The swap_slow_sort() function sorts an array with nmemb elements of size
size. The base argument points to the start of the array. The contents of
the array are sorted in ascending order according to a comparison function
pointed to by compar, which is called with two arguments that point to the
objects being compared. The comparison function must return an integer
less than, equal to, or greater than zero if the first argument is
considered to be respectively less than, equal to, or greater than the
second. If two members compare as equal, their order in the sorted array
is undefined.
```

הנחיות נוספות:

מותר להשתמש בפונקציה memswap_ptr שראיתם בשיעור:

```
/// Swaps two blocks of memory.
/// We assume the blocks don't overlap.
///
/// \param p1,p2 - the blocks to swap.
/// \param size  - the size of the blocks.
void memswap_ptr(void* p1, void* p2, size_t size);
```

השורה הבאה מגרילה מספר בין 1 ל 10 (כולל):

```
return 1 + (rand() % 10);
```

הקפידו על כל כללי התכנות הנכונים שלמדתם בשיעור, כולל כתיבת פונקציות עזר וכו'!

אין צורך להכליל את הקבצים הדרושים לשימוש בפונקציות הספרייה הסטנדרטית.

ב. (20 נק') כתבו Google test לקוד שכתבתם בסעיף א. שימו לב, אין צורך לכתוב את ההכללות אלא אך ורק את ה TEST macros. כמו כן מספיק לבדוק את הקוד עם מערך של ints כאשר INT_MIN הוא הערך הקטן ביותר ש int יכול לקבל ו INT_MAX הוא הערך הגדול ביותר ש int יכול לקבל. יש לכתוב קוד מלא אך אם יש חזרה על קוד אפשר לסמן שורות ולכתוב שהן חוזרות.

בהצלחה!