

**פקולטה:** מדעי הטבע

**מחלקה:** מדעי המחשב

**שם קורס:** מבוא למחשבים ושפת C

**שם המרצה:** ד"ר מוטי גבע, גב' חרות סטרמן

**קוד קורס:** 2-7028510-1/2/3/4/5

**תאריך בחינה:** 6 פברואר 2019

**סמסטר:** א' **מועד:** א'

**משך הבחינה:** שתיים

**חומר עזר:** פתוח.

#### **הוראות כלליות:**

- הניסוח הוא בלשון זכר מטעמי נוחות ומתייחס לכולם/!
- יש לרשום את התשובות בצורה מרוכזת במחברת הבחינה בלבד. טופס הבחינה לא ייבדק.
- המקום המוקצה לכל תשובה הוא על בסיס המקום שמופיע בטופס הבחינה (לרוב שורה בודדת בלבד).
- אין לחרוג מהמקום המוקצה לטובת כתיבת תשובות. במידה ותהיה חריגה מהמקום המוקצה, חלק התשובה שחורג לא ייבדק.
- 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 נרשמה תשובה, ההנחה היא שנרשם "לא יודע/ת" עבור אותו סעיף.

## שאלה 1 (20 נקודות)

נתון הקוד הבא:

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str[] = "hmmmm...";
6      const char * const ptr1[] = {"to be", "or not to be", "that is the question"};
7      char *ptr2 = "that is the question";
8
9      ptr1[2] = str;
10     strcpy(str, "Hamlet");
11     for (int i=0; i<sizeof(ptr1)/sizeof(*ptr1); ++i)
12         printf("%s ", ptr1[i]);
13     return 0;
14 }
```

א. בהתבסס על הקוד לעיל (10 נק'):

a. האם הקוד מצליח להתקמפל ולרוץ: כן \ לא (2 נק')

b. אם נכשל (8 נק' לחילופין עם סעיף c)

i. באיזה שלב: pre-processing / compiling / linking / run-time (2 נק')

ii. טעות בשורה (3 נק'): \_\_\_\_\_

iii. מה סיבת הטעות (3 נק'): \_\_\_\_\_

c. אם מצליח, כתבו את הפלט של התוכנית (8 נק' לחילופין עם סעיף b):

ב. אם נשנה את שורה מס' 9, לשורה הבאה (10 נק'):

(&ptr2)[3] = str;

a. האם הקוד מצליח להתקמפל ולרוץ: כן \ לא (2 נק')

b. אם נכשל (8 נק' לחילופין עם סעיף c)

i. באיזה שלב: pre-processing / compiling / linking / run-time (2 נק')

ii. טעות בשורה (3 נק'): \_\_\_\_\_

iii. מה סיבת הטעות (3 נק'): \_\_\_\_\_

c. אם מצליח, כתבו את הפלט של התוכנית (8 נק' לחילופין עם סעיף ב):

## שאלה 2 (20 נקודות)

נתון הקוד הבא השמור בקובץ בשם macro.c:

```
1  #include <stdio.h>
2  #define a(s) #s
3  int main()
4  {
5  #ifdef b
6      #if b < 0
7          printf("%s\n", a(2 b | ~ 2 b));
8      #else
9          printf("%d\n", 2 b | ~ 2 b);
10     #endif
11 #else
12     printf("%d\n", xor(1,2));
13 #endif
14     return 0;
15 }
```

## הערה

כפי שנלמד בכיתה, הדגל D- מגדיר מאקרו. להלן תמצות של המידע במדריך של gcc:

-D name=**definition**

The contents of **definition** are processed as if they appeared in a #define directive.

לדוגמה: אם מריצים את הקומפיילר `gcc -D a=4+3` (או לחילופין `'a=4+3' -D`), זה שקול לכתובה של השורה הבאה בתוך קבצי המקור של התוכנה:

```
#define a 4+3
```

**הערה נוספת:** כדאי להסתכל במצגת `1a_basic_C_and_building` שקף 55 (מופיע גם בסוף המבחן)

על בסיס שורות הקומפילציה של `gcc` בכל סעיף ענו על השאלות (2 נק' לכל תת-סעיף, מקסימום 20 נק'):

א. שורת הקומפילציה: `gcc macro.c`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

ב. שורת הקומפילציה: `gcc macro.c -D b=-1`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

ג. שורת הקומפילציה: `gcc macro.c -D b=+2`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

ד. שורת הקומפילציה: `gcc macro.c -D b=+3`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

ה. שורת הקומפילציה: `gcc macro.c -D b=hello`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

ו. שורת הקומפילציה: `gcc macro.c -D 'xor(a,b)=a^b'`

- i. האם נכשל: כן \ לא.
- ii. אם נכשל, באיזה שלב: `pre-processing / compiling / linking`, טעות בשורה: \_\_\_\_\_
- iii. אם הצליח, מה מודפס למסך בעת הרצת התוכנית: \_\_\_\_\_

### שאלה 3 (20 נקודות)

נתון הקוד הבא

```
1  #include <stdio.h>
2  int plus(int a, int b) { printf("plus(%d,%d): ", a,b); return a + b; }
3  int minus(int a, int b) { printf("minus(%d,%d): ", a,b); return a - b; }
4  int mul(int a, int b) { printf("mul(%d,%d): ", a,b); return a * b; }
5  int div(int a, int b) { printf("div(%d,%d): ", a,b); return a / b; }
6  int and(int a, int b) { printf("and(%d,%d): ", a,b); return a & b; }
7  int or(int a, int b) { printf("or(%d,%d): ", a,b); return a | b; }
8  int xor(int a, int b) { printf("xor(%d,%d): ", a,b); return a ^ b; }
9  int main()
10 {
11     int i;
12     int (*fp[])(int,int) = {plus,minus,mul,div,and,or,xor};
13     for (i=0; i<sizeof(fp)/sizeof(*fp);++i)
14         printf("%d\n", fp[i](i+1,i-1));
15     return 0;
16 }
```

א. מה הפלט של התוכנית (14 נק'):

ב. מהו טיפוס (type) המשתנה fp (3 נק')?

ג. מה ה- sizeof(fp) (3 נק')?

#### שאלה 4 (20 נקודות)

בהתייחס לקוד הבא, ענו על הסעיפים המופיעים לאחרי:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  unsigned long getid() {
4      static unsigned long id=0;
5      return ++id;
6  }
7  typedef struct _node {
8      unsigned long id;
9      char *name;
10     struct _node *next;
11 } node_t;
12 void init(node_t *node, char *name, node_t *next) {
13     if(node) {
14         node->id=getid();
15         node->name=name;
16         node->next=next;
17     }
18 }
19 void alloc(node_t **node, char *name, node_t *next) {
20     if (node) {
21         *node = malloc(sizeof(node_t));
22         init(*node, name, next);
23     }
24 }
25 node_t n1 = {0,"",NULL};
26 node_t *n2 = NULL;
27 int main()
28 {
29     char buff[] = "";
30     node_t n3 = {0,buff,NULL};
31     node_t *n4;
32     alloc(&n4,n1.name,NULL);
33     init(&n3,buff,n4);
34     alloc(&n2,"hello",&n3);
35     init(&n1,(char*)init,n2);
36     for (n4=&n1; n4; n4=n4->next) {
37         printf("%p\t%p\t%p\n", &n4->id, &n4->name, n4->name);
38     }
39 }
```

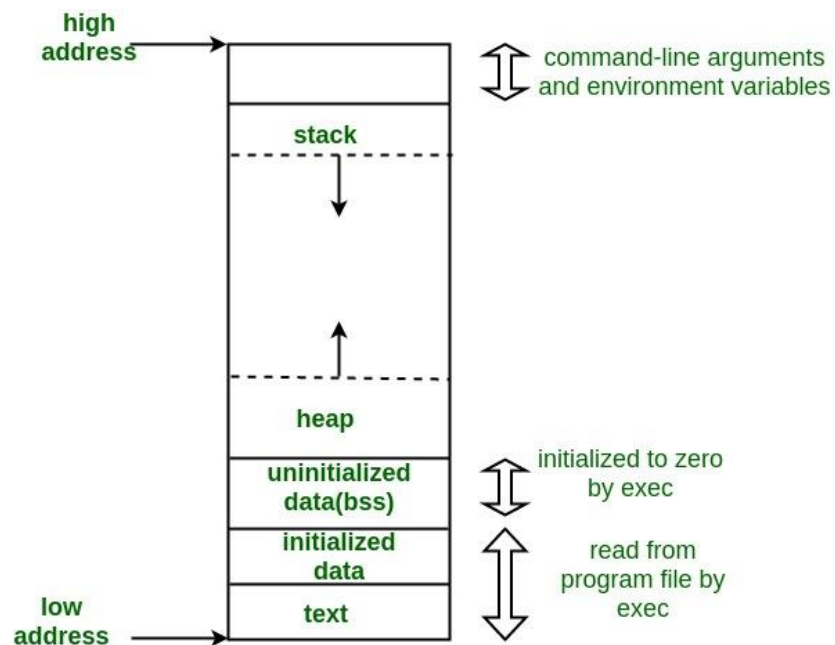
א. כתבו לגבי כל ביטוי בטבלה הבאה, באיזה אזור בזיכרון הוא נמצא (text/code segment, data/global, read-only data, heap, stack או לא מוגדר. (8 נק')

Expression	Line number	Memory area
&id	4	
&node	12	
&name	12	
&next	12	
&n1	32	
&n1.name	32	
n1.name	32	
n3.name	33	
&n2	34	

ב. שורה 37 מדפיסה שלשות של כתובות. לכל שורת הדפסה מלאו את שם האזור בזיכרון בו נמצאת הכתובת (text, data, read-only data, heap, stack) או לא מוגדר. (6 נק')

&n4->id	&n4->name	n4->name

ג. בהנחה שהכתובת &n4=0x100, מה תהיה הכתובת (&n3.name) (2 נק'): \_\_\_\_\_



- ד. מה הערך sizeof(buff) בשורה 29 (2 נק'): \_\_\_\_\_
- ה. מה הערך sizeof(n3.name) בשורה 30 (2 נק'): \_\_\_\_\_

### שאלה 5 (20 נקודות)

הקוד בשאלה 4 לא מטפל כראוי בשחרור הקצאות דינאמיות. עליכם להציע שינויים בקוד, כך שניתן יהיה לשחרר את ההקצאות הדינאמיות כראוי, לפי ההנחיות הבאות.

עליכם להתבסס על הפונקציה release, שהמבנה שלה מתואר להלן. הפונקציה מקבלת כפרמטר צומת שורש לרשימה המקושרת (root node) ועליה לשחרר את ההקצאות הדינאמיות שבתוך הרשימה. עליכם לממש את הפונקציה עצמה, בהתבסס על החלקים הקיימים. במידת הצורך, ניתן להוסיף משתנים מקומיים, גלובאליים ואחרים. אם צריך לבצע שינויים בפונקציות אחרות, יש לתאר בדיוק את שורת השינוי ואת תוכן השינוי. אין להוסיף פונקציות חדשות לקוד. ניתן להוסיף מאקרו לטובת קריאות הקוד.

ניתן להניח כי כל הקצאות הדינאמיות, מבוצעות באמצעות הפונקציה alloc:

```
void release(node_t *root) {
```

```
while (root) {
```

```
}
```

```
}
```



## משתנים נוספים

[illegible]

## שינויים לפונקציות

[illegible]

## assert.h

```
// procedure that actually prints error message
void __assert(char* file,int line,char* test);

#ifdef NDEBUG
    #define assert(e)      ((void)0)
#else
    #define assert(e)      \
        ((e) ? (void)0 :   \
         __assert(__FILE__, __LINE__, #e))
#endif
```