

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: 24 פברואר 2019

סמסטר: א' מועד: ב'

משך הבחינה: שעותיים

חומר עזר: פתוח.

הוראות כלליות:

- הניסוח הוא בלשון זכר מטעמי נוחות ומתייחס לכולם/!
- יש לרשום את התשובות בצורה מרוכזת ולפי הסדר 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.
- יש לענות על כל השאלות. אין בחירה במבחן.
- ניתן להסתמך על כל סעיף במבחן גם אם לא פתרתם אותו על מנת לפתור סעיף אחר.
- 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.

שאלה 1 (20 נק')

```
1  #include <stdio.h>
2  typedef struct _float_struct {
3      unsigned int mantissa      :23;
4      unsigned int exponent      :8;
5      unsigned int sign          :1;
6  } fs_t;
7  typedef union _float_union {
8      fs_t float_st;
9      float native_float;
10     unsigned int native_int;
11 } fu_t;
12 int main()
13 {
14     fu_t fu[3];
15     fu[0].float_st.sign = 1;
16     fu[0].float_st.exponent = 127;
17     fu[0].float_st.mantissa = 1 << 22;
18     fu[1].native_int = 0x40E00000;
19     fu[2].native_float = fu[0].native_float + fu[1].native_float;
20     printf("%lu\n", sizeof(fu)); // %lu - unsinged long
21     printf("%f\n", fu[0].native_float);
22     printf("%f\n", fu[1].native_float);
23     printf("%f\n", fu[2].native_float);
24     return 0;
25 }
```

הערות:

- הקוד מבוסס על Little-Endian, דהיינו הביטים בכתובות הנמוכות הם least significant ובכתובות הגבוהות הם most significant. לדוגמה: המספר 15.0 מיוצג ע"י sign=0, exponent=0x82, mantissa=0x700000. (mantissa היא השם של חלק השבר).
- בנספח למבחן בסוף מופיע השקף מהמצגות שמתאר מבנה של floating point.

על בסיס הקוד לעיל ענו על השאלות הבאות: (5 נק' לכל סעיף)

- א. מה הפלט של שורה 20? _____
- ב. מה הפלט של שורה 21? _____
- ג. מה הפלט של שורה 22? _____
- ד. מה הפלט של שורה 23? _____

שאלה 2 (20 נק')

```
1  #include <stdio.h>
2  typedef struct _struct {
3      int arr[20];
4      int *ptr;
5  } st_t;
6  void print_st(st_t *_st, int i) { printf("%d\n", _st->arr[i]); }
7  void func1(st_t _st1) {
8      for (int i=0; i<20; ++i)
9          _st1.arr[i] = 20-i;
10     print_st(&_st1, 5);
11 }
12 void func2(st_t _st1) {
13     for (int i=0; i<20; ++i)
14         _st1.ptr[i] = 20-i;
15     print_st(&_st1, 10);
16 }
17 int main()
18 {
19     st_t st1;
20     st1.ptr = st1.arr;
21     for (int i=0; i<20; ++i)
22         st1.arr[i] = i+1;
```

```

23  print_st(&st1, 0);
24  func1(st1);
25  print_st(&st1, 1);
26  func2(st1);
27  print_st(&st1, 2);
28  return 0;
29  }

```

על בסיס הקוד לעיל ענו על השאלות הבאות: (4 נק' לכל סעיף)

- א. מה הפלט של שורה 23? _____
- ב. מה הפלט של שורה 24? _____
- ג. מה הפלט של שורה 25? _____
- ד. מה הפלט של שורה 26? _____
- ה. מה הפלט של שורה 27? _____

שאלה 3 (20 נק')

```

1  #include <stdio.h>
2  #define macro(x,y) {  x=x^y; \
3                        y=x^y; \
4                        x=x^y; }
5  int main()
6  {
7      int a=1, b=2;
8      macro(a,b);
9      printf("a: %d, b: %d\n", a, b);
10 }

```

בהתבסס על הקוד לעיל, ענו על השאלות הבאות: (5 נק' לכל סעיף)

א. מה הערכים המודפסים בשורה 9:

a: _____, b: _____

ב. מחליפים את שורות 5-10 (פונקציית main) בקוד הבא:

```

5 int main()
6 {
7     int a=73, b=37;
8     macro(73,37);
9     printf("a: %d, b: %d\n", a, b);
10 }

```

i. האם הקוד יתקמפל וירוע בצורה תקינה? כן \ לא (1 נק')

ii. אם כן, מה תוצאת ההדפסה בשורה 9: _____ (4 נק')

iii. אם לא, באיזה שלב ובאיזו שורה תתגלה הטעות?

a. שלב: Preprocessing / Compiling / Linking / Runtime (2 נק')

b. שורה: _____ (2 נק')

ג. מעדכנים את macro ואת main לפי הקוד הבא:

```

1 #include <stdio.h>
2 #define macro(x,y,z) { x=(z)((long)x^(long)y);\
3                       y=(z)((long)x^(long)y);\
4                       x=(z)((long)x^(long)y); }
5 int main()
6 {
7     char *str1="hello", *str2="world";
8     macro(str1,str2,char*);
9     printf("%s %s\n", str1, str2);
10    return 0;
11 }

```

i. האם הקוד יתקמפל וירוע בצורה תקינה? כן \ לא (1 נק')

ii. אם כן, מה תוצאת ההדפסה בשורה 9: _____ (4 נק')

iii. אם לא, באיזה שלב ובאיזו שורה תתגלה הטעות?

a. שלב: Preprocessing / Compiling / Linking / Runtime (2 נק')

b. שורה: _____ (2 נק')

ד. בהתייחס לסעיף ג', משנים את שורה 8 לשורה הבאה:

macro(str1[0],str2[0],char);

i. האם הקוד יתקמפל וירוע בצורה תקינה? כן \ לא (1 נק')

ii. אם כן, מה תוצאת ההדפסה בשורה 9: _____ (4 נק')

iii. אם לא, באיזה שלב ובאיזו שורה תתגלה הטעות?

a. שלב: Preprocessing / Compiling / Linking / Runtime (2 נק')

b. שורה: _____ (2 נק')

שאלה 4 (20 נק')

file.h

```
1  #pragma once
2  static int id = 0;
3  extern int sum;
4  static int incID();
5  static int decID();
6  extern void push(int n);
7  int pop();
```

file.c

```
8  #include <stdlib.h>
9  #include <math.h>
10 #include "file.h"
11 int sum = 0;
12 static void *arr = NULL;
13 static int incID() { return ++id; }
14 static int decID() { return --id; }
15 extern void push(int n) {
16     arr = realloc(arr, incID()*sizeof(int));
17     sum += (((int*)arr)[id-1]=n);
18 }
19 int pop() {
20     int ret = ((int*)arr)[id-1];
21     arr = realloc(arr, decID()*sizeof(int));
22     sum -= ret;
23     return ret;
```

24 }

main.c

```
25 #include <stdio.h>
26 #include <stdlib.h>
27 #include "file.h"
28 int main()
29 {
30     int a;
31     id = 5;
32     for (int i=0; i<10; ++i)
33         push(10-i);
34     for (int i=0; i<5; ++i)
35         printf("%d\n", pop());
36     printf("Sum: %d\n", sum);
37     printf("ID: %d\n", id);
38     return 0;
39 }
```

הקוד לעיל מחולק לשלושה קבצים, file.h, file.c ו-main.c. על בסיס הקוד לעיל, ענו על השאלות הבאות:

- א. מה יהיה הפלט בשורה 36: Sum: _____ (4 נק')
- ב. מה יהיה הפלט בשורה 37: ID: _____ (4 נק')
- ג. לגבי כל ביטוי כתבו מה תחום ההכרה שלו (כל התוכנית, הקובץ בו מוגדר, שורות X-Y), ובאיזה אזור בזיכרון הוא מוגדר (code, stack, heap, global, data, לא מוגדר): (12 נק')

קובץ	שורה	ביטוי	תחום ההכרה	אזור בזיכרון
main.c	31	id		
file.c	13	id		
file.c	14	declID()		
file.c	11	sum		
file.c	12	&arr		
file.c	17	*arr		

שאלה 5 (20 נק')

בדומה לפונקציות push ו-pop משאלה מס' 4, כתבו שתי פונקציות push ו-pop המקבלות מחרוזות בגודל משתנה ומכניסות\מוציאות אותן למערך רציף אחד בזיכרון. ניתן להוסיף משתנים בתוך הפונקציות אך אין להוסיף או לשנות קוד מחוץ לשתי הפונקציות האמורות. אפשר להשתמש בכל הקוד שקיים בשאלה 4. ניתן להשתמש במערך arr ולבנות בתוכו מבנה נתונים מתאים. (כל פונקציה 10 נק')

ה-prototype של הפונקציות הוא כדלהלן:

```
void push(const char *d, const int len);
```

d הוא מצביע המכיל את המחרוזת לשמירה במערך, ו-len מכיל את אורך המחרוזת.

```
void pop(char **d, int *len);
```

d הוא מצביע למצביע שבתוכו תשמר המחרוזת האחרונה שהוכנסה למערך, ו-len יכיל את אורך המחרוזת שנשמרה. יש להקצות את המקום הנדרש בתוך משתנה היעד.

קריאה לדוגמה תראה כך:

```
push("hello world", sizeof("hello world"));

push("goodbye", sizeof("goodbye"));

char *ptr;

int len;

pop(&ptr, &len);

printf("%d: %s\n", len, ptr); // should print "8: goodbye"

free(ptr);

pop(&ptr, &len);

printf("%d: %s\n", len, ptr); // should print "12: hello world"

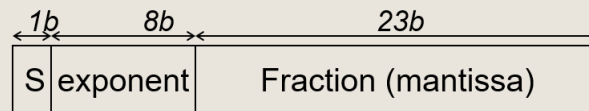
free(ptr);
```


Floating-Point

$$F \times 2^E$$

Need to represent F (*fraction*), E (*exponent*), and sign.

IEEE 754 Floating-Point Standard (32-bits):



$$N = (-1)^S \times 1.\text{fraction} \times 2^{\text{exponent} - 127}, \quad 1 \leq \text{exponent} \leq 254$$

$$N = (-1)^S \times 0.\text{fraction} \times 2^{-126}, \quad \text{exponent} = 0$$

exponent = 255 used for special values:

If fraction is non-zero, NaN (not a number).

If fraction is zero and sign is 0, positive infinity.

If fraction is zero and sign is 1, negative infinity.