

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

סמסטר א', מועד ב', יום שלישי י"ב אדר תשע"ח, Tuesday, Feb 27, 2018

**מחלקה:** מדעי המחשב

**מרצים:** חרות סטרמן, ערן קאופמן, יוסי זגורי

**שם הקורס:** מבוא למחשבים ושפת C

**מס' הקורס:** 7028510

**משך הבחינה:** שעתיים וחצי.

**אופן הבחינה:** בחינה במחברות.

### הנחיות כלליות:

- המבחן בנוי משלושה חלקים וכולל בחירה (סה"כ 104 נק').
  - חלק א' (40 נק'): תחביר ומושגים, יש לענות על 8 מתוך 10 שאלות.
  - חלק ב' (40 נק'): שאלות פתוחות, יש לענות על 2 מתוך 3 שאלות.
  - חלק ג' (24 נק'): ניהול זיכרון, יש לענות על השאלה – ללא בחירה.
- המבחן במתכונת סגורה - אין להשתמש בכל חומר עזר כתוב או אלקטרוני.
- יש לענות על כל השאלות במחברת הבחינה בלבד.
- השתמשו בדפים האי זוגיים כדפי טיוטה במידת הצורך וציינו "טיוטה" בראש הדף.
- ניתן לקבל 20% מערך שאלה פתוחה במידה ונכתבה התשובה "לא יודע/ת" ולא נכתבה תשובה אחרת.
- נא להקפיד על כתב נקי ומסודר, שמות משמעותיים והזחות נכונות. קוד שאינו קריא לא ייבדק.
- הניסוחים בלשון זכר מטעמי קריאות.

**בהצלחה!**

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### חלק א': תחביר ומושגים (40 נק')

נא לענות על 8 מתוך 10 השאלות. יש לבחור את התשובה המתאימה ביותר. משקל כל שאלה 5 נק'.

#### שאלה 1

בפונקציה הבאה מה יהיה הפלט?

```
double avr(int a[],int size){
    int sum=0;
    for(int i =0 ; i < size; i++)
        sum += a[i];
    return sum/size;
}

int main(){
    int arr[]={1,3,4};
    printf("%.2f\n",avr(arr,3));
}
```

- א) 2.00
- ב) 2.6666
- ג) 2.66
- ד) 2

#### שאלה 2

מי מהמשפטים הבאים נכון?

- א) בביטוי אריתמטי תמיד מתבצעת המרה ( Explicit or Implicit Casting ).
- ב) המרה משתמעת (Implicit) נדרשת כאשר עושים Up Casting.
- ג) משתנים המוגדרים בתוך בלוק ( { } ) יוגדרו תמיד כמשתנים לוקאליים.
- ד)  $\text{Sizeof}(\text{void}^*) < \text{Sizeof}(\text{char}^*) < \text{Sizeof}(\text{int}^*)$

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### שאלה 3

מה גודל המבנה Person בבתים (bytes):

```
union id {
    int id;
    int name;
};

struct Person {
    union id me;
    int isStudent:1;
    int catagory:2;
    unsigned int err;
};
```

- א) 8
- ב) 12
- ג) 16
- ד) 20

### שאלה 4

int i=3, j=5, \*p = &i, \*q = &j;  
למה שווה הביטוי:  $7 * * p / * q + 7$  ?

- א) 21
- ב) 11
- ג) 18
- ד) 10

### שאלה 5

מי מהתחבירים הבאים אינו שקול לאחרים בהתייחסות לתא ספציפי בזיכרון?

- א)  $C[i][j]$
- ב)  $*(C[i]+j)$
- ג)  $** (C+i+j)$
- ד)  $*(*(C+i)+j)$

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### שאלה 6

בהינתן הפונקציה הבאה ומערך המוגדר כ- `int arr[10]` ממנו נבחר ערכים, מי מהקריאות הבאות יגרום לפונקציה למלא ייעודה באופן נכון?

```
void swap(int *a, int *b)
{
    int t;
    t = *b;
    *b = *a;
    *a = t;
}
```

- א) `swap(arr, (&*(arr+4)))`
- ב) `swap(&arr, &(arr+1))`
- ג) `swap(&arr, (arr+1))`
- ד) `swap(arr[1], arr[2])`

### שאלה 7

```
void doSomething(char *str)
{
    while(*str) {
        if((*str>64) && (*str<91)){
            *str+=32;
            str++;
        }
    }
}
```

א) מה אמור לעשות הקוד הנ"ל?

---

---

---

ב) מדוע וכיצד יש לשפר קוד זה?

---

---

---

---

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### שאלה 8

האם הקוד הבא יתקמפל?

```
int f(const int *x){
    return *x++;
}
```

- (א) הקוד יתקמפל וגם לו היה רשום במקום `return ++*x;` הוא היה מתקמפל.
- (ב) הקוד לא יתקמפל וגם לו היה רשום במקום `return ++*x;` הוא לא היה מתקמפל.
- (ג) הקוד יתקמפל אך לו היה רשום `return ++*x;` במקום הוא לא יתקמפל.
- (ד) הקוד לא יתקמפל אך לו היה רשום `return ++*x;` הוא כן יתקמפל.

### שאלה 9

מה נכון לומר על התוכנית הבאה:

```
#include<stdio.h>
int main()
{
    int x = 20, y = 100%80, i;
    for(i=0; i<10; i++);
        if(x - y)
            printf("x = %d y = %d\n", x, y);
    return 0;
}
```

- (א) הפונקציה `printf()` נקראת 10 פעמים.
- (ב) פלט התוכנית הוא `x = 20 y=20`.
- (ג) לתוכנית זו אין פלט כלל.
- (ד) הביטוי `if(x-y)` גורר שגיאת קומפילציה.

### שאלה 10

מה מייחד את אופציית הגדרת Bit Fields במסגרת Struct ?

- (א) מאפשרת להפעיל אופרטורים המיועדים למניפולציית ביטים.
- (ב) מאפשרת חלוקת הזיכרון שמקבל ה-Struct ללא אילוצי סוגי המשתנים המוגדרים בשפה.
- (ג) מאפשרת הגדרת סוג משתנה חדש (`typedef`) בכל גודל.
- (ד) ניתן לפנות לכתובת כל ביט (Bit) בזיכרון ישירות.

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### חלק ב': שאלות פתוחות (40 נק')

נא לענות על 2 מתוך 3 השאלות במחברת הבחינה. משקל כל שאלה 20 נק'.

#### שאלה 11: בעיית הצריחים

ברצוננו למלא לוח שחמט בצריחים כך שאף צריח אינו באותה השורה או עמודה כמו הצריח השני. לדוגמא בלוח שהוא שלוש על שלוש:

נשים צריח אחד בקצה השמאלי העליון כתוצאה מכך השורה והעמודה הראשונים תפוסים.

1	x	x
x		
x		

לאחר מכן נשים צריח בנקודה [2,2] וכל העמודה והשורה השניים תפוסים.

1	x	x
x	2	x
x	x	

ולבסוף נשים את הצריח השלישי בנקודה [3,3] וזה יחשב לנו כאפשרות מספר אחת.

1	x	x
x	2	x
x	x	3

אפשרות נוספת היא האופציה הבאה:

	1	
2		
		3

נכתוב פונקציה לספירת האפשרויות להצבת צריחים בלוח. אם ברצוננו לממש פונקציה זו כפונקציה רקורסיבית:

```
#define N 8
```

```
int countRooks(int row);
```

כאשר ה main נראה כך :

```
int main(){
    printf("%d\n", countRooks(0));
}
```

(א) (5 נקודות) איזה מהמשתנים כדאי לנו להגדיר כסטטיים ואלו כמשתנים אוטומטיים על המחשנית ומדוע?

(ב) (15 נקודות) ממשו את הפונקציה countRooks.

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

### שאלה 12

אנו רוצים לממש תבנית עיצוב (design pattern) מסוג ספק-צרכן (Producer-consumer). כלומר צד אחד (הספק) יכניס לרשימה מקושרת מצביע לפונקציה callback אותה הוא ירצה שהצרכן יריץ לאחר שיקבל אותה. ספקים שונים כמובן ירצו להריץ פונקציות שונות. הפונקציה consume מקבלת את ראש הרשימה (head), קוראת למצביע לפונקציה callback(), משחררת את ההודעה מהרשימה, ומעבירה את head להצביע על ההודעה הבאה ברשימה. (א) (5 נקודות) עליך להגדיר את מבני הנתונים הדרושים למימוש תבנית העיצוב ואת חתימת הפונקציה consume. (ב) (15 נקודות) עליך לממש את הפונקציה consume.

### שאלה 13

ברצוננו לממש מבנה נתונים מסוג max heap. מבנה הנתונים ממומש מעל מערך בגודל מקסימלי של 100, לכל איבר בערימה ישנם שני בנים באינדקסים  $2 * index + 1$ ,  $2 * index + 2$ . (א) (5 נקודות) הגדר את מבנה הנתונים של ה heap הנדרש. (ב) (15 נקודות) עליך לממש את הפונקציה

```
void addHeap(heap* mHeap, int elem)
```

אשר מוסיפה אלמנט חדש ל heap.

## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

**חלק ג': ניהול זיכרון (24 נק')**

יש לענות על השאלה – ללא בחירה.

### **שאלה 14**

בהתייחס לקטע הקוד הבא:

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define N 4
4.
5. typedef struct a
6. {
7.     int* arr;
8.     int size;
9. } A;
10.
11. int arr1[] = {1, 2, 3, 4};
12.
13. int main()
14. {
15.     A a1;
16.     a1.arr = arr1;
17.     a1.size = sizeof(arr1)/sizeof(arr1[0]);
18.     A* a2;
19.     a2 = malloc(sizeof(A));
20.     int arr2[N];
21.     for (int i=0; i<N; ++i)
22.     {
23.         arr2[i] = i*N;
24.     }
25.     a2->arr = arr2;
26.     a2->size = N;
27.
28.     printf("First array: ");
29.     for (int i=0; i<a1.size; ++i)
30.         printf("%d, ", a1.arr[i]);
31.     printf("\n");
32.
33.     printf("Second array: ");
34.     for (int i=0; i<a2->size; ++i)
35.         printf("%d, ", a2->arr[i]);
36.     printf("\n");
37.
38.     return 0;
39. }
```



## הפקולטה למדעי הטבע - המחלקה למדעי המחשב

(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, אזור הקוד, לא מוגדר. לדוגמא הכתובת &p1 לאחר שורה 26 היא כתובת במחסנית. הניחו שכל ההקצאות הדינמיות מצליחות. כאשר כתוב &p הכוונה היא היכן מאוחסן המצביע p. כאשר כתוב p הכוונה היא לאיזה כתובת p מצביע. כאשר כתוב \*p הכוונה היא לאיזה כתובת המוצע של p מצביע.

**יש לענות על 8 שורות בטבלה מתוך 10.**

מיקום	כתובת	מספר כתובת
	arr1 (11)	1
	&(a1) (15)	2
	a1.arr (16)	3
	&(a1.size) (17)	4
	&a2 (18)	5
	a2 (18)	6
	a2 (19)	7
	arr2 (20)	8
	a2->arr (25)	9
	&(a2->size) (26)	10

(ב) (4 נק') הוסיפו קוד לשחרור כל הזיכרון בתכנית.  
(ג) (4 נק') מה הפלט של התכנית?

🐞 בהצלחה!