ת"ז: _____ מספר מחברת:_____

סמסטר א' תשע"ב מועד א
תאריך: 10.2.2012
שעה: 9:00
משך הבחינה: 4 שעות
כל חומר עזר אסור בשימוש

בחינה בקורס: מבוא למדעי המחשב

- **יש לענות על כל 5 השאלות.**
- **בכל השאלות במבחן יש לכתוב פונקציות יעילות ככל האפשר גם בזמן וגם בזיכרון.**
- **שימרו על סגנון התכנות. הגדירו קבועים היכן שצריך וכתבו פונקציות עזר.**
- **יש לשחרר את כל שטחי הזיכרון שהוקצו דינאמית כאשר אין בהם צורך יותר.**
- **אין צורך לבדוק הצלחה של הקצאות דינאמיות.**
- **בכל השאלות במבחן ניתן להניח שהקלט תקין.**
- **אלא אם נאמר אחרת, ניתן להשתמש (מבלי לממש) בכל הפונקציות הנתונות בעמוד 2 של מבחן זה.**
- **אין צורך לתעד את הפונקציות.**

- **יש לכתוב ת"ז ומספר מחברת בראש כל עמוד. עמוד ללא ת"ז בראשו לא ייבדק.**
- **יש לענות על כל שאלה במקום שמיועד לה בגוף השאלון. המחברות הן טיוטה בלבד ולא תיבדקנה.**
- **שימו לב כי הבחינה כוללת 16 עמודים, כולל עמוד זה.**

בהצלחה!

ת"ז: _____ מספר מחברת:_____

רשימת פונקציות ספריה מותרות לשימוש:

פונקציה	ספריה
`int strlen(char str[])`	String.h
`void strcpy(char dest[], char src[])`	String.h
`void strcat(char dest[], char src[])`	String.h
`int strcmp(char s1[], char s2[])`	String.h
`int __max(int a, int b)`	stdlib.h
`int __min(int a, int b)`	stdlib.h
`double pow(double base, double exp)`	math.h
`double sqrt(double x)`	math.h
`double fabs(double x)`	math.h
`cout`	iostream
`cin`	iostream
`char cin.get(char& outCh)`	iostream
`void cin.getline(char outStr[], int max, char delimiter)`	iostream

ת"ז: _____ מספר מחברת:_____

<u>שאלה 1 (20 נקודות)</u>:

כתבו את הפונקציה

```
char* octalToBin(char octalNum[]);
```

הפונקציה מקבלת כקלט מחרוזת octalNum, המכילה ספרות בין '0' ל-'7' המהוות הצגה אוקטלית (בבסיס 8) של מספר שלם וחיובי.
הפונקציה <u>מייצרת ומחזירה</u> מחרוזת <u>חדשה</u> המכילה את הספרות '0' ו-'1', המהוות את הייצוג הבינארי (בבסיס 2) של המספר המיוצג ב- octalNum.

<u>הערה</u>: במחרוזת המוחזרת ייתכנו **לכל היותר שני אפסים מובילים**.

למשל

octalToBin("721") תחזיר את המחרוזת "111010001".

octalToBin("7") תחזיר את המחרוזת "111".

octalToBin("12") תחזיר את המחרוזת "001010".

ת"ז: _____ מספר מחברת:_____

```
char* octalToBin(char octalNum[])
```

ת"ז: _____ מספר מחברת:_____

ת"ז: _____ מספר מחברת:_____

<u>שאלה 2 (20 נקודות):</u>

כתבו את הפונקציה:

```
int find0(int nums[], int n);
```

הפונקציה מקבלת כקלט מערך `nums` של מספרים שלמים, ואת גודלו `n` (שלם חיובי). ידוע שהמערך `nums` מקיים את הדרישות הבאות:

1. המערך מכיל את **כל** המספרים: $0,1,2\ldots,(n-1)$
2. כל המספרים: $1,2,\ldots,(n-1)$ ממויינים אחד ביחס לשני. כלומר, אם $1\leq i<j\leq(n-1)$, אז i מופיע במערך לפני j.

על הפונקציה להחזיר את אינדקס התא בו מופיע המספר 0 במערך `nums`.

למשל:

- `find0([1,2,3,4,0,5,6,7],8)` **תחזיר 4**.
- `find0([0,1,2,3,4,5,6,7],8)` **תחזיר** 0.
- `find0([1,0,2,3,4,5,6,7],8)` **תחזיר 1**.

ת"ז: _____ מספר מחברת:_____

```
int find0(int nums[], int n)
```

ת"ז: _____ מספר מחברת:_____

שאלה 3 (20 נקודות):

נתונים הקבועים הבאים:

```
const int MAX_NAME_LEN = 100;
const int MAX_NUM_OF_CHILDREN = 20;
```

נתונה ההגדרה הבאה לייצוג משפחה:

```
struct family{
     char name[MAX_NAME_LEN];          //שדה המכיל את שם המשפחה
     int numChildren;                  //שדה המכיל את מספר הילדים
     int ages[MAX_NUM_OF_CHILDREN];    //מערך של גילאי הילדים
};
typedef struct family Family;
```

כתבו את הפונקציה הבאה:

```
void findLargeFamilies(Family fams[], int& size,
                       int threshold);
```

הפונקציה מקבלת מערך fams של משפחות ואת גודלו size ובנוסף ערך סף חיובי threshold. המערך fams והפרמטר size הם פרמטרי קלט+פלט.

בקלט המערך fams מכיל size משפחות כלשהן ובפלט הוא יכיל רק את המשפחות בעלות יותר מ- threshold ילדים, עפ"י הסדר המקורי שהיה ביניהן.

הפונקציה תעדכן את size להיות גודלו הלוגי החדש.

ת"ז: _____ מספר מחברת:_____

```
void findLargeFamilies(Family fams[], int& size,
                       int threshold)
```

ת"ז: _____ מספר מחברת:_____

<u>שאלה 4 (20 נקודות):</u>

כתבו את הפונקציה

```
int* splitByKSize(int inData[], int size, int k);
```

הפונקציה מקבלת כקלט מערך `inData` של מספרים שלמים חיוביים, את גודלו `size` (שלם חיובי) וערך שלם חיובי k על פיו תבוצע החלוקה.

הפונקציה <u>מייצרת ומחזירה מערך חדש</u> ובו כל המספרים של `inData` מסודרים לפי החוקיות הבאה:

ראשית, כל מספרי `inData` שהם בתחום: {0,1,…,k-1}

אחריהם, כל המספרי `inData` שהם בתחום: {k,k+1,….,2*k-1}

אחריהם, כל המספרי `inData` שהם בתחום: {2*k,2*k+1,….,3*k-1}

…

אחריהם, כל המספרי `inData` שהם בתחום: {(k-1)*k,(k-1)*k+1,….,k*k-1}

ולבסוף כל המספרי `inData` שגדולים או שווים ל: k*k.

שימו לב:

- סדר המספרים בתוך כל תחום איננו חשוב.
- ייתכן בהחלט שחלק מהתחומים לא יכללו אף מספר מתוך `inData`.

<u>דוגמאות</u>

פלט אפשרי לקריאה

```
sortByKSize([13,100,12,24,2,31,5,19,79,20,30],11,5)
```

הוא המערך `[2,5,13,12,19,24,20,100,31,79,30]`.

ת"ז: _____ מספר מחברת:_____

```
int* sortByKSize(int inData[], int size, int k)
```

ת"ז: _____ מספר מחברת:_____

ת"ז: _____ מספר מחברת:_____

<u>שאלה 5א (10 נקודות):</u>

כתבו מימוש **רקורסיבי** של הפונקציה:

```
int sumPositive(int data[], int size);
```

הפונקציה מקבלת כקלט מערך `data` של מספרים שלמים, ואת גודלו `size` (שלם חיובי) ומחזירה את סכום המספרים החיוביים במערך.

<u>דוגמאות</u>:

הקריאה `sumPositive([2,-3,5,-7,8,-1],6)` **תחזיר** 15 (2+5+8).

הקריאה `sumPositive([0,-2,4],3)` **תחזיר** 4.

הקריאה `sumPositive([-2,-3],2)` **תחזיר** 0.

ת"ז: _____ מספר מחברת:_____

```
int sumPositive(int data[], int size)
```

ת"ז: _____ מספר מחברת:_____

<u>שאלה 5ב (10 נקודות):</u>

כתבו מימוש **רקורסיבי** של הפונקציה

```
bool subsetMul(int numbers[], int size, int mul);
```

הפונקציה מקבלת כקלט מערך `numbers` של מספרים שלמים, ואת גודלו `size` (שלם חיובי) ומספר נוסף, `mul`.

הפונקציה תחזיר true אם ורק אם קיימת איזושהי תת-קבוצה של מספרים מתוך המערך (לאו דווקא רצופה) שמכפלת המספרים בה שווה ל- `mul`.

<u>דוגמאות</u>

הקריאה `subsetMul([17,-3,2,-1,6],5,-18)` תחזיר true, כי קיימת תת-הקבוצה {-3,6,-1}.

הקריאה `subsetMul([17,-3,2,-1,6],5,17)` תחזיר true, כי קיימת תת-הקבוצה (שמכילה איבר בודד) {17}.

הקריאה `subsetMul([17,-3,2,-1,6],5,23)` תחזיר false, כי לא קיימת תת-הקבוצה מתאימה.

ת"ז: _____ מספר מחברת:_____

```
bool subsetMul(int numbers[], int size, int mul)
```