

סמסטר א' תש"ע מועד א
תאריך: 29.1.10
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 בבחן יש לכתוב פונקציות ייעילות ככל האפשר גם בזמן וגם בזיכרון.
- שימרו על כללי תכונות מודולריי. הגדרו קבועים היכן שציריך וכתבו פונקציות עזר.
- יש לשחרר את כל שטחי הזיכרון שהוקazo דינאמית כאשר אין בהם צורך יותר.
- אין צורך לבדוק הצלחה של הקצאות דינמיות.
- בכל השאלות בבחן ניתן להניח שהקלט תקין.
- אין צורך לטעד את הפונקציות.
- יש לכתוב ת"ז ומספר מחברת בראש כל עמוד. עמוד ללא ת"ז בראשו לא יבדק.
- יש לענות על כל שאלה במקום שמיועד לה בגוף השאלה. המחברות הן טיווחה בלבד ולא תידקנה.
- שימוש לב Ci הבחינה כוללת 14 עמודים, כולל עמוד זה.

בצלחה!

שאלה 1 (15 נקודות):

"סדרת הפרשים העוקבים" מוגדרת באופן הבא:

האיבר הראשון בה הוא 1

האיבר השני בה גדול ב- 1 מהאיבר הראשון, כלומר האיבר השני הוא 2.

האיבר השלישי בה גדול ב- 2 מהאיבר השני, כלומר האיבר השלישי הוא 4.

האיבר הרביעי בה גדול ב- 3 מהאיבר השלישי, כלומר האיבר הרביעי הוא 7.

וכך הלאה.

כabbo תוכנית אשר קולעת מהמשתמש מספר שלם חיובי ח, ומדפיסה את ח האיברים הראשונים (עם רוח ביןיהם) בסדרת הפרשים העוקבים.

למשל אם המשמש מכניס 8, התוכנית תדפיס: 29 22 16 11 7 4 2 1.

עמוד 3 מתוך 14

מספר מחברת:

ה'ג:

```
#include <iostream>
using namespace std;
void main()
```

שאלה 2 (20 נקודות):

הנি�חו כי מוגדר הקבוע הבא:

```
const int MAX_DIGITS_IN_INT = 10;
```

כתבו את הפונקציה

```
void decimalLengths(int nums[], int size, int counts[]);
```

הפונקציה מקבלת כקלט מערך `nums` של מספרים שלמים חיוביים ואת גודלו `size` (שלם חיובי). כמו כן הפונקציה מקבלת מערך נוסף `counts` שגודלו הפיסי הוא: `1 + INT_MAX_IN_INT`. בערך זה הפונקציה אמורה להחזיר את הפלט שלה.

הפונקציה תעדכן את המערך `counts` כך ש- `[i]`将在 `counts` יכיל את מספר המספרים ב- `nums` שהם בני `i` ספרות עשרוניות.

כלומר `[1]`将在 `counts` יכיל את מספר המספרים ב- `nums` שהם בני ספרה אחת.

`[2]`将在 `counts` יכיל את מספר המספרים ב- `nums` שהם בני שתי ספרות.

`[3]`将在 `counts` יכיל את מספר המספרים ב- `nums` שהם בני שלוש ספרות.

וכך הלאה.

הערות:

1. ב- `[0]`将在 `counts` יש לשימם תמיד את הערך 0.
2. במנית ספרות של מספר אין לכלול אפסים מובילים.
3. שימו לב, מספר העשרות המקסימלי של שלם חיובי מתייחס לתוך בסביבה שלנו היא 10, ולכן גודלו הפיסי של המערך `counts` מספיק.

דוגמה:

בקריאה (decimalLengths([1,100,9,9,12,20009,78],7,counts) הפונקציה תעדכן את המערך `counts` כך: `[0,3,2,1,0,1,0,0,0,0,0,0]`.

עמוד 5 מתוך 14

מספר מחברת;

ח'ז:

```
int decimalLengths(int nums[], int size, int counts[])
```

שאלה 3 א (10 נקודות):

כתבו מימוש רקורסיבי של הפונקציה:

```
int countLower(char* str);
```

הfonקציה מקבלת כקלט מחרוזת `str`, ומחזירה את מספר האותיות lower case ("האותיות הקטנות",
במחרוזת `a-z`.

דוגמאות:

הקריאה ("Good day" countLower 6

הקריאה ("GOOD DAY" countLower 0

והקריאה ("53 & 4 + 3 Ay Q!" 1 countLower

עמוד 7 מתוך 14

מספר מחברת:

ת"ז:

```
int countLower(char *str)
```

שאלה 3 ב (15 נקודות):

כתבו מימוש רקורסיבי של הפונקציה

```
void getMinToStart(int numbers[], int n);
```

הfonקציה מקבלת מערך numbers של מספרים שלמים ואות גודלו n (שלם חיובי). הפונקציה תנסה את סדר איברי המערך numbers כך שבסוף ריצתה, numbers[0] יהיה את המספר הקטן ביותר מבין המספרים ב- numbers. הסדר של שאר המספרים במערך אינו משנהו.

דוגמאות

עבור המערך [-7,-3,5,-2,1,3,8,9,3], numbers=[-3,5,-2,1,-7,8,9,3]

עבור המערך [0,7,9,2,0,0,1], numbers=[7,9,2,0,0,1]

עמוד 9 מתוך 14

מספר מחברת:

ה'ז

```
void getMinToStart(int numbers[], int n)
```

שאלה 4 (20 נקודות)

כתבו את הפונקציה:

```
int setDiffSize(int A[], int n1, int B[], int n2);
```

הfonקציה מקבלת כקלט מערכיים A ו-B של מספרים שלמים ואת גודלם n1 ו-n2 (שלמים חיוביים) בהתאם. נתון כי הנתונים בכל אחד מהמערכות **מוינו**inus בסדר עליה ממש.

על הפונקציה להחזיר את מספר האיברים אשר נמצאים ב- A ואינם נמצאים ב- B.

הערה: היה שמערכי הקלט ממוינוinus בסדר עליה ממש, מספר לא יכול להופיע יותר מפעם אחת.

דוגמאות

הקריאה: (4, 2, 1, 3, 7, 8), 5, [-6, -5, 1, 2, 7] 5, [1, 3, 7, 8], 3, כיוון שאיברים 5, 6, 1 נמצאים ב- A אך אינם נמצאים ב- B.

והקריאה: (7, 5, 1, 2, 7), 5, [-1, 3, 4, 8, 14, 18, 25], 7, תחזיר 5.

עמוד 11 מתוך 14

מספר מחברת:

ת"ז:

```
int setDiffSize(int A[], int n1, int B[], int n2)
```

שאלה 5 (20 נקודות):

המבנה הבא מייצג נקודה במרחב:

```
typedef struct point{
    double x;           // רכיב ה- x
    double y;           // רכיב ה- y
} Point;
```

נקודה נמצאת בربיע הראשון אם שני הרכיבים שלה חיוביים (גדולים ממש מאפס). למשל (1.4,2.3) ו- (0.9,18.0) נמצאות בربיע הראשון אבל (-3.4,4.0) ו- (0,0.9) (12.1,-12.1) לא נמצאות בربיע הראשון.

כתבו את הפונקציה הבאה:

```
int* pointsInQuad1(Point* pointsArr, int size, int& resSize);
```

הfonקציה מקבלת מערך `pointsArr` של נקודות, ואת גודלו `size` (שלם חיובי).
הfonקציה מייצרת ומוחירה מערך המכיל את האינדקסים של כל הנקודות ב- `pointsArr`, הנמצאות בربיע הראשון, לפי סדר הופען ב- `pointsArr`. כמו כן הfonקציה מעדכנת בפרמטר הפלט `resSize` את גודלו של המערך המוחזר.

הערה: אם כל הנקודות ב- `pointsArr` אינן בربיע הראשון תחזיר `NULL` ותעדכו בפרמטר הפלט `resSize` את הערך 0.

לדוגמה, הקריאה:

```
pointsInQuad1([(0.9,18.0), (0.0,0.9), (-3.4,4.0), (1.4,2.3), (12.1,-12.1)], 5, resSize)
תחזיר את המערך: [0, 3] ותעדכו ב- resSize את הערך 2
```

עמוד 13 מתוך 14

מספר מחברת:

ת"ז:

```
int* pointsInQuad1(Point* pointsArr, int size, int& resSize)
```

עמוד 14 מתוך 14

מספר מחברת:

ת"ז: