

סמסטר א' תשס"ז מועד א
תאריך: 16.2.07
שעה: 9:00
משך הבחינה: 4 שעות
כל חומר עזר אסור בשימוש



בחינה בקורס: מבוא למדעי המחשב

- יש לענות על כל 4 השאלות.
- שימרו על כללי תכנות מודולארי. הגדירו קבועים היכן שצריך וכתבו פונקציות עזר.
- בכל השאלות במבחן יש לכתוב פונקציות יעילות ככל האפשר.
- 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.
- 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 13 עמודים, כולל עמוד זה.

בהצלחה!

שאלה 1 (20 נקודות):

כתבו את הפונקציה:

```
void split(char *str);
```

שמקבלת כפרמטר קלט/פלט מחרוזת המכילה רק אותיות אנגליות גדולות ואותיות אנגליות קטנות. על הפונקציה להחזיר את המחרוזת `str` מסודרת כך שכל האותיות הגדולות יקדימו את כל האותיות הקטנות (כלומר בתחילת המחרוזת יופיעו כל האותיות הגדולות ואחריהן כל האותיות הקטנות). אין חשיבות לסדר פנימי של האותיות הגדולות וכמו כן אין חשיבות לסדר הפנימי של הקטנות.

דוגמה:

פלטים אפשריים עבור הקלט "exFamOplRe" הם בין השאר "FORexample" ו-"ORFemplaxe"

שאלה 2 (20 נקודות):

בניח שמוגדר:

```
const int N = 100;
```

כתבו את הפונקציה:

```
void sort(int events[][N], int size);
```

הפונקציה מקבלת כפרמטר קלט/פלט מערך דו-מימדי ריבועי events בגודל size×size (למערך הדו-מימדי N עמודות פיסיות). איברי המערך events מייצגים קודים של אירועים. כל קוד של אירוע הוא מספר שלם בין 1 ל-10.

על הפונקציה להחזיר את המערך events ממוין בסדר עולה לאורך השורות וכלפי מטה.

דוגמה:

$$\text{events} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 10 & 9 \\ 1 & 3 & 4 \end{bmatrix} \quad \text{קלט: } \text{size} = 3$$

$$\text{events} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 4 & 9 & 10 \end{bmatrix} \quad \text{פלט:}$$

הערות:

1. הניחו ש- $\text{size} \leq N$.

2. בבדיקת התשובה יושם דגש מיוחד על יעילות האלגוריתם.

מספר מחברת: _____

ת"ז: _____

שאלה 3 א (10 נקודות):

כתבו מימוש רקורסיבי של הפונקציה:

```
int countAppearances(char* str, char ch);
```

הפונקציה מקבלת כקלט מחרוזות `str` ותו `ch`.
 על הפונקציה להחזיר את מספר המופעים של `ch` ב-`str`.

דוגמאות:

עבור:

```
str = "this is an example, is it?"  
ch = 'a'
```

הפונקציה תחזיר 2

עבור:

```
str = "this is an example, is it?"  
ch = 'z'
```

הפונקציה תחזיר 0

שאלה 3 (10 נקודות):

כתבו מימוש רקורסיבי של הפונקציה:

```
int appearsMost(char* text, char* letters, char& ch);
```

הפונקציה מקבלת כקלט שתי מחרוזות לא ריקות text ו- letters. נתון שבמחרוזת text מותרות חזרות של תווים, אך במחרוזת letters אף תו לא מופיע יותר מפעם אחת.

הפונקציה תעדכן בפרמטר הפלט ch את התו מ- letters אשר מספר הופעותיו ב-text מקסימאלי. אם יותר מאשר תו אחד מתווי letters מופיע ב-text במספר מקסימאלי של הופעות, יש לעדכן ב- ch את האחרון שבהם. הערך המוחזר של הפונקציה יהיה מספר ההופעות של התו שעודכן ב- ch.

דוגמאות:

עבור

```
text = "this is an example, is it?"
letters = "atbem"
```

יוחזר 2, וב- ch יופיע 'e' (מספר המופעים של 'a', 't' ו- 'e' ב- text הוא 2, אך 'e' מופיע אחרי 'a' ו- 't').

עבור

```
text = "this is an example, is it?"
letters = "atbim"
```

יוחזר 4, ו- ch יכיל את 'i'.

עבור

```
text = "this is an example, is it?"
letters = "bcd"
```

יוחזר 0, ו- ch יכיל את 'd'.

שאלה א4 (20 נקודות):

כתבו את הפונקציה:

```
char* intToStr(int n);
```

הפונקציה מקבלת כקלט שלם חיובי n ומחזירה מחרוזת שמכילה את ספרות המספר (בבסיס 10) לפי סדרן משמאל לימין (הספרה השמאלית ביותר של n תיכתב בתא 0 של המחרוזת).

הערה:

בפתרון סעיף זה אין להשתמש בפונקציות ספרייה.

דוגמה:

עבור $n = 722$

יוחזר:

'7'	'2'	'2'	'\0'
-----	-----	-----	------

שאלה 4 ב (20 נקודות):

המבנה הבא משמש לייצוג כתובת:

```
struct address{
    char* street;
    int house_num;
};
```

כתבו את הפונקציה

```
char* addToStr(struct address add);
```

המקבלת מבנה מסוג כתובת ומחזירה מחרוזת שמכילה את שם הרחוב ואחריו מספר הבית (בין השם למספר יפריד רווח אחד).

לכתיבת הפונקציה המבוקשת **עליכם להשתמש** בפונקציה מסעיף א' (גם אם לא הצלחתם לכתוב אותה) ובפונקציות הספרייה `string.h` הבאות:

(i) `strcat` (צימוד מחרוזות) שה- `prototype` שלה הוא

```
void strcat(char* str1, char* str2);
```

פונקציה זו מעתיקה את המחרוזת `str2` (כולל ה-`'\0'`) אל סופה של המחרוזת `str1` (התו הראשון של `str2` נכתב על ה-`'\0'` של `str1`). הפונקציה מניחה שגודל ההקצאה של `str1` מספיק לצורך הצימוד. היא משנה את `str1` אך לא משנה את `str2`.

(ii) `strlen` (חישוב אורך מחרוזת) שה- `prototype` שלה הוא

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

פונקציה זו מחזירה את מספר התווים במחרוזת הקלט `str` (לא כולל ה-`'\0'`).

