

(3)

סמסטר קיץ תשס"ד
מועד ב': 14/11/2004
משך הבחינה: 3 שעות
חומר עזר: שני דפי עזר

בחינה בקורס: תוכנה 1
פרופ' דן כהן-אור ואפי פוגל

הנחיות כלליות לבחינה:

- המבחן מורכב משלוש שאלות תכנות ושתי שאלות "אמריקאיות". יש להשיב על כל השאלות.
- חובה לתעד כל פעולה לא טריוויאלית שנעשית.
- יש לכתוב קוד קריא ויעיל ככל האפשר.
- נא לכתוב בכתב קריא.

בהצלחה !

ת.ז.: 0400 7289

טבלת תשובות לחלק האמריקאי

a	b	c	d	e	
			X		שאלה 4
				X	שאלה 5

הערות לתשובות בחלק האמריקאי

אמנם רק תשובה נכונה תזכה בניקוד עבור כל שאלה, אולם ניתן לצרף לכל תשובה אמריקאית הסבר קצר. ההסבר לא ייבדק במסגרת הבדיקה הרגילה, אך ניתן יהיה להסתמך עליו במסגרת ערעור, אם יידרש. מומלץ לצרף הסבר לתשובה אמריקאית במיוחד במקרים קיצוניים בהם **נראה לך** שתשובתך נכונה, אך **נראה לך** שהיא איננה התשובה שאליה התכוון המרצה. מובן שהסבר שגוי או בלתי סביר לא יועיל בכל מקרה (אך גם לא יזיק, אם ממילא סימנת את התשובה הנכונה).

שאלה 4:

שאלה 5:

שאלה מס' 1 (30 נקודות)

נתון טיפוס נתונים PRODUCT שמוגדר באופן הבא:

```
typedef struct product {
    int num;           /* catalog number */
    float price;       /* product price */
    unsigned int length; /* name length */
    char * name[30]    /* product name */
} PRODUCT;
```

נתונים שני קבצים בינאריים in_file1 ו-in_file2 שמכילים struct-ים מטיפוס PRODUCT המסודרים בסדר עולה של מספר קטלוגי (שמוצג ע"י שדה num). המוצרים בכל קובץ שונים זה מזה, אך אותו מוצר יכול להופיע בשניהם.

כתוב תוכנית יעילה שקוראת את שני הקבצים הנ"ל ויוצרת קובץ בינארי נוסף, out_file2, ובו מיזוג תוכנם של שני הקבצים שנקראו. לקובץ החדש אותו פורמט כמו לשני קבצי הקלט, כלומר הוא מכיל את המוצרים בסדר עולה של מספר קטלוגי וכל המוצרים בו שונים זה מזה.

ניתן להשתמש בפונקציה:

```
size_t fread(void * a_ptr, size_t elem_size, size_t n, FILE * fp);
```

פונקציה זו קוראת $n * \text{elem_size}$ בתים (bytes) מתוך הקובץ המקושר ל-fp לתוך המערך ש-a_ptr מצביע אליו. הפונקציה מחזירה את מספר האלמנטים שנקראו בהצלחה.

שאלה מס' 2 (30 נקודות)

כתוב פונקציה בעלת המפרט (prototype) הבא:

```
int ** sort_matrix(int ** matrix, int n, int m);
```

המקבלת מערך דו-מימדי המוצבע על ידי המשתנה matrix. מספר העמודות במערך הוא m ומספר השורות במערך הוא n. בשלב ראשון, הפונקציה ממיינת כל שורה במערך בנפרד, לפי סדר עולה של הערכים. בשלב שני, הפונקציה ממיינת את השורות בסדר לקסיקוגרפי המוגדר כדלקמן. שורה i במטריצה A קטנה משורה j במטריצה A אם ורק אם קיים d ($0 \leq d \leq m-1$) כך שלכל k , $0 \leq k < d$ מתקיים: $A[i][k] = A[j][k]$ וגם $A[i][d] < A[j][d]$. במקרה של הצלחה, הפונקציה מחזירה מצביע למטריצה הממוינת. במקרה של כשלון, הפונקציה מחזירה NULL. מותר להקצות זיכרון נוסף בגודל $O(n)$ לכל היותר. לדוגמא, עבור קלט שמתאר את המטריצה הבאה (שורות משמאל לימין):

0	10	20	30
30	20	10	0
0	10	0	10
10	20	10	20

הפונקציה מחזירה מצביע למערך שמתאר את המטריצה הבאה:

0	0	10	10
0	10	20	30
0	10	20	30
10	10	20	20

שאלה מס' 3 (25 נקודות)

כתוב פונקציה בעלת המפרט (prototype) הבא:

```
char * strtok(const char * str, const char * delimiters);
```

התוים המופיעים במחרוזת delimiters הם התוים התוחמים. התו `\0` אינו תו תוחם.
 הפונקציה משתמשת בתוים תוחמים על מנת לחתוך את מחרוזת הקלט `str` לתתי-מחרוזות הנקראות `token`.
 אם מחרוזת הקלט `str` שונה מ-`NULL`, הפונקציה סורקת את מחרוזת הקלט מתחילתה. אחרת, הפונקציה סורקת את המחרוזת שנסרקה בקריאה הקודמת לאותה פונקציה, החל מהמקום שבו הסתיימה הסריקה בקריאה הקודמת. בכל קריאה, הפונקציה מחפשת את ה-`token` הבא ומחזירה מצביע ל-`token` שנמצא. אם לא נמצא `token` נוסף, יוחזר `NULL`. הסריקה מתבצע באופן הבא:

- הפונקציה מחפשת את התו הראשון במחרוזת שאינו אחד מבין התוים התוחמים.
- אם הסריקה הגיע לסוף מחרוזת הקלט ולא מצאה תו שאינו תו תוחם, הפונקציה מחזירה `NULL`.
- אם בסריקה נמצא תו שאינו תו תוחם, תו זה הוא התחלה של `token` חדש. הפונקציה ממשיכה לסרוק את מחרוזת הקלט עד שמגיעה לתו התוחם הראשון (או לסוף מחרוזת הקלט). הפונקציה מחליפה תו זה ב-`\0` בתוך מחרוזת הקלט ומחזירה את המצביע להתחלה של ה-`token`.

אם הסריקה הסתיימה בסוף מחרוזת הקלט ובפעם הבאה הפונקציה נקראת עם `str` שהוא `NULL`, אז התנהגות הפונקציה בקריאה הבאה אינה מוגדרת. שימו לב: קבוצת התוים התוחמים יכולה להשתנות מקריאה לקריאה. במימוש של פונקציה זו אין להשתמש בפונקציות נוספות. לדוגמא, הפקודות הבאות:

```
char * t = strdup("You-know-it's---sad,but--true.");
t = strtok(t, "--");
while (t != NULL) {
    printf("%s\n", t);
    t = strtok(NULL, "-.,");
}
```

מדפיסות על המסך:

```
You
know
it's
sad
but
true
```

שאלה מס' 4 (10 נקודות)
נתון הביטוי הרגולרי הבא:

`(m/^(.+)\{1,2\}$ /)`

המחרוזות הבאות:

- 1. ab
- 2. abab
- 3. ababab
- 4. abababab
- 5. ababababab

אילו מחרוזות מתאימות לביטוי הרגולרי?

- (a) 2, 3 ו-1
- (b) 1, 2, 3, 4 ו-5
- (c) 1, 2, 3 ו-4
- (d) 2, 3 ו-4
- (e) 1, 2, 3 ו-4

שאלה מס' 5 (10 נקודות)

מה תדפיס התכנית הבאה?

```
int x = 5;
void g(int *x) {
    (*x)--;
    printf("%d ", *x);
}
void f(int y) {
    printf("%d ", x);
    if (x < 6) {
        x = 3000;
        g(&y);
    }
    x = x - y;
    printf("%d ", x);
}
int main() {
    int x = 6;
    f(x);
    printf("%d ", x);
    return 0;
}
```

- a) 6 0 6
- b) 6 5 2995 5
- c) 5 5 0 6
- d) 5 5 2999 6
- e) 5 5 2995 6