

מבוא למדעי המחשב מ' 234114

מבחן מועד א', סמסטר אביב תשס"ד 05.7.04

שם משפחה שם פרטי מס' סטודנט

--	--	--	--	--	--	--	--	--	--

שאלה	ערך	הישג
1	20	
2	15	
3	20	
4	20	
5	15	
6	15	
סכום	105	

משך המבחן: 3 שעות.
חומר עזר: אין להשתמש בחומר עזר.

הוראות לנבחנים ולנבחנות:

- 1 מלאו את הפרטים בראש דף זה (בעט).
- 2 בדקו שיש 14 עמודים (6 שאלות) כולל עמוד זה.
- 3 התשובות ייכתבו על טופס המבחן.
- 4 כתבו בכתב-יד נקי וברור (מומלץ להשתמש בעפרון ומחק).
- 5 אין להשתמש בפונקציות ספריה אלא אם כתוב אחרת בשאלה.
- 6 אין לכתוב הערות והסברים לתשובות (אם לא התבקשת).
- 7 בכל השאלות ניתן להניח שהקלט תקין.
- 8 ניתן לצבור עד 105 נקודות. תנסו לצבור את מירב הנקודות.

אביב תשס"ד
מרצים: ד"ר רועי פרידמן, צחי קרני.
מתרגלים: אולג רוכלנקו, רון רובינשטיין, עזרא אוהיון, ניר זפקוביץ, ויסאם קאדרי.

שאלה 1 (20 נקודות)

סעיף א. (5 נק')

1.

נכון או לא נכון? הטיפוסים של $*k$ ושל $**k$ תמיד שונים. _____

2.

נתון ש x , a הם כולם משתנים מסוג `integer` המוגדרים בתוכנית. ערך של x לפני ביצוע השורה הבאה הוא 5:

```
if ( (a<10)|| (a>9) || ++x ) x--;
```

מה יהיה ערך של x אחרי ביצוע של השורה הזאת?

א. 5

ב. 4

ג. 6

ד. 4 או 5 -- תלוי בערכים של המשתנים a ו b .

ה. לא ניתן לדעת.

3.

```
int a = 5;  
double *b = (double *) &a;
```

הערך של $*b$ לאחר סיום השורה השניה הוא:

א. 5

ב. 5.0

ג. לא ניתן לדעת

ד. התוכנית לא תתקמפל (שגיאת קומפילציה)

סעיף ב. (5 נק')

מה תדפיס התוכנית הבאה:

```
int main()
{
    struct word
    {
        unsigned int pos;
        unsigned int len;
    };
    typedef struct word Word;

    int w;
    int c = 1; // 1 for 234111, 2 for 234114 and 3 for 234117
    char *cname = "14723411IntroWetoLikeCS";
    Word words[NUMWORDS] = {{13,2},{17,4},{8,5},{15,2},{21,2},{3,5},{0,1}};

    words[NUMWORDS-1].pos = c;

    for (w = 0; w < NUMWORDS; w++)
    {
        char s[MAXLEN];
        strncpy(s,cname + words[w].pos, words[w].len);
        s[words[w].len] = '\0';
        printf("%s ",s);
    }

    return(0);
}
```

הערה:

פונקציה

char strncpy(char s1, const char *s2, size_t n)
מעתיקה n תווים מ-s2 ל-s1.

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

סעיף ג. (10 נק')

מהן סיבוכיות הזמן והמקום כפונקציה של n עבור הקריאה

`puzzle(n);`

כאשר נתונים מימושים הבאים של הפונקציה `puzzle()` ופונקצית עזר `helper()`:

1.

<pre>void puzzle (int n) { int i, mid = n; if (n <= 1) return; for (i = n; i >= 0; i--) if ((mid > i) && (mid > n-i)) mid = i; helper(mid); return; }</pre>	<pre>void helper (int m) { int cnt = m; for (; cnt > 0; cnt--) printf("?"); puzzle(m); return; }</pre>
---	--

סיבוכיות זמן: _____
סיבוכיות מקום: _____

2.

<pre>void puzzle (int n) { int i; if (n <= 1) return; helper(n/2); helper(n/2); return; }</pre>	<pre>void helper (int m) { int cnt = m; for (; cnt > 0; cnt--) printf("?"); puzzle(m); return; }</pre>
---	--

סיבוכיות זמן: _____
סיבוכיות מקום: _____

שאלה 2 (15 נקודות)

עבור מחרוזת s , תת מחרוזת של s היא רצף תווים שמופיע בשלמותו ב- s כאשר יש להתייחס לאות קטנה ואות גדולה כאותה אות ($a=A$) ולהתעלם מרווחים.
לדוגמה: **erf** היא תת מחרוזת של **Wonderful**.

עבור שתי מחזורות s_1, s_2 נגדיר את תת המחזורות משותפת הארוכה ביותר (Longest Common Substring) כמחזורות t , כאשר t היא תת-מחזורות של s_1 וגם של s_2 , ואין תת מחזורות משותפת ארוכה יותר.

מטרתכם בסעיף זה לממש את הפונקציה הבאה:

```
void longest_common_substring(char* s1, char* s2);
```

שמקבלת שתי מחרוזות s_1 ו- s_2 ומדפיסה את תת המחרוזת המשותפת הארוכה ביותר עבורם.

נסמן ב- n את סכום האורכים של s_1 ו- s_2 .

סיבוכיות זמן נדרשת: $O(n^3)$. סיבוכיות מקום: $O(1)$.

הערות:

1. יתכן שישנה יותר ממחרוזת אחת שהיא Longest Common Substring. במקרה כזה, עליכם להדפיס מחרוזת LCS אחת, לבחירתכם.
2. פונקציה שלא תעמוד בדרישות הסיבוכיות לא תקבל ניקוד.

```
void longest_common_substring (char* s1, char* s2)
{
```

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

}

שאלה 3 (20 נקודות)

בהינתן מערך דו מימדי של מספרים $N \times M$, נגדיר בחירה: רשימה המכילה עד איבר אחד מכל שורה של המערך. נייצג בחירה על ידי מערך בחירה בגודל N אשר מכיל אינדקסים בין 0 ל- M , לכל שורה במערך המקורי נשמור במערך הבחירה את האינדקס של האיבר המבוקש (הערך M יסמן כי אנחנו לא בוחרים איבר משורה זו).

דוגמא a הינו המערך המקורי ו-ia הינו מערך הבחירה:

```
a = 4 6 8 9 0
     3 2 1 5 21
     8 1 3 7 -7
     1 2 1 0 2
```

```
ia= 2 0 3 5
```

כלומר בחרנו את האיברים 7, 3, 8

$$a = \begin{pmatrix} 4 & 6 & 8 & 9 & 0 \\ 3 & 2 & 1 & 5 & 21 \\ 8 & 1 & 3 & 7 & -7 \\ 1 & 2 & 1 & 0 & 2 \end{pmatrix}$$

בכל הסעיפים יש להניח שהקלט תקין. הגדלים M, N מתקבלים כפרמטרים לפונקציה.

א. השלם את הפונקציה `print_selection` המקבלת מערך דו מימדי ומערך בחירה ומדפיסה את האיברים אותם בחרנו:

```
void print_selection(int a[][M], int ia[], int n, int m)
{
```

[illegible]

}

```
int find_min_index(int a[][M], int ia[], int n, int m)
{
```

}

ג. השלם את הפונקציה merge_n המקבלת מערך דו-מימדי בגודל $N \times M$ שכל שורה בו הינה ממוינת. ומערך חד-מימדי בגודל $N \times M$. בסיום הפונקציה המערך החד מימדי יכיל את כל איברי המערך המקורי ממוינים. סיבוכיות זמן $O(N^2M)$, סיבוכיות מקום $O(1)$.

דוגמא:

```
int a[N][M] = {{1,3,4,6},{0,2,3,8},{2,7,8,10}};
int c[N*M];
merge_n(a,N,M,c);
```

לאחר הפעלת merge_n: $c = [0,1,2,2,3,3,4,6,7,8,8,10]$

```
void merge_n(int a[][M], int n, int m, int c[])
{
    int* ia = _____
    int i, _____
    for(i = 0; i < n; i++) _____
    while(_____)
    {
        _____ = find_min_index(a, ia, n, m);
        _____
        c[ic++] = _____
        _____
    }
    free(____);
}
```

שאלה 4 (20 נקודות)

בשאלה זו נשאל על מטריצות (מערכים דו-ממדיים) משולשות עליונה (כלומר $A[i][j]$ לא מוגדר אם $i > j$) בלבד.

במטריצה משולשת עליונה נגדיר את העמודה ה- t בתור אוסף האיברים $A[0][t], \dots, A[t][t]$. נאמר שמטריצה משולשת עליונה היא ממוינת עמודות אם כל עמודה בה ממוינת מהקטן לגדול וגם כל איבר בעמודה k נמצא גם בעמודה $k+1$.

מכיוון שנשתמש במערכים דו ממדיים לייצוג מטריצות משולשות עליונה, נאחסן 1- באיברים לא מוגדרים.

לדוגמה, המערך הבא מייצג מטריצה משולשת ממוינת עמודות:

3	1	1	1
-1	3	3	2
-1	-1	5	3
-1	-1	-1	5

א. השלם את הפונקציה הבאה שבהינתן מטריצה משולשת עליונה ממוינת עמודות A ואיבר x ומחזיר את העמודה בעלת האינדקס הנמוך ביותר בה x מופיע.

תאר בקצרה את האלגוריתם. שימו לב שאם האלגוריתם המתואר אינו נכון, הקוד לא ייבדק!

```
int find_row(int A[N][N], int x)
{
```

$$\}$$

ב. השלם את הפונקציה הבאה שמקבלת מערך של מספרים $B[N]$, ובונה ממנו מטריצה משולשת עליונה ממוינת עמודות באופן הבא: בעמודה ה- i במטריצה יהיו האיברים $B[0], \dots, B[i-1]$.

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```

}

```

סיבוכיות מקום נוסף: $O(1)$, סיבוכיות זמן: $O(N^2)$.

הנחיות:

1. לכל אורך השאלה ניתן להניח שכל האיברים המוגדרים הם חיוביים.
2. ניתן להניח ש-N הוא קבוע שהוגדר ב-`#define`.
3. ניתן להשתמש בפונקציות הבאות:

```

void max_sort(int a[], int n)
void bubble_sort(int a[], int n)
void merge_sort(int a[], int n)
void quick_sort(int a[], int n)

```

שאלה 5 (15 נקודות)

מחשבון postfix הינו מחשבון אשר מקבל כקלט בתחילה את כל הארגומנטים לחישוב ובסופו את האופרטורים לביצוע.

לדוגמה: הביטוי

10 12 30 40 2 3 + - * +

יחושב

$(10 + (12 * (30 - (40 - (2 + 3))))) = -50$

השלימו את החסר במימוש הפונקציה הרקורסיבית postfix_calculator.

שימו לב, כי הביטוי לחישוב נתון במערך calc_expression וניתן להניח כי הקלט הינו תמיד חוקי (כל האברים במערך הם מספרים או אחד מהאופרטורים +, - או *). כמו כן ניתן להניח כי הקריאה הראשונה ל-postfix_calculator תמיד חוקית (len הינו מספר האיברים במערך calc_expression).

```
#include <stdio.h>
```

```
int postfix_calculator(int calc_expression[], int len);
```

```
int main()
```

```
{  
    int calc_expression[] = {10, 12, 30, 40, 2, 3, '+', '-', '-', '*', '+'};  
    int res = postfix_calculator(calc_expression, 11);  
    printf("The result is %d\n", res);  
    return(0);  
}
```

```

int postfix_calculator(int calc_expression[], int len)
{
    int arg1, arg2, res;

    if (_____)
        return (_____);

    arg1 = _____
    arg2 = _____

    switch (_____)
    {
        case '+':
            _____;
        break;
        case '-':
            _____;
        break;
        case '*':
            _____;
        break;
    }

    return (res);
}

```

שאלה 6 (15 נקודות)

כתבו פונקציה המקבלת כקלט מערך A בגודל 10, בו מופיע עבור כל ספרה כמה פעמים היא צריכה להיות מודפסת, ומדפיסה את כל הקומבינציות האפשריות של מספרים המורכבים מהספרות האלו.

לדוגמא, עבור:

$$A = \{ 2, 1, 0, 0, 0, 0, 0, 0, 0, 0 \}$$

יודפסו המספרים הבאים: 100,010,001.

```
void print_combinations(int A[])
{
```

[illegible]

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

}