



מכון טכנולוגי חולון
Holon Institute of Technology

מבחן סדנה מתקדמת בתכנות-61108

ד"ר מרק קורנבליט, מר חיים שפיר, גב' אסתר אמיתי

סמסטר קיץ, מועד א', תשע"ט

תאריך: 23.10.19

הוראות:

- משך המבחן 3 שעות.

- אין להשתמש בחומרי עזר, פרט לדף המצורף לשאלון

שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת מחרוזת **fileName**.
תוכן של קובץ טקסט ששמו נכלל ב- **fileName** הוא טקסט מוצפן שנוצר מהטקסט המקורי באופן הבא:
אחרי תו אחרון של כל שורה בטקסט המקורי מוסיפים רווח (ניתן להניח שאין רווחים בטקסט המקורי);
אחרי הרווח מוסיפים מספר ביקורת וצמוד למספר מוסיפים תו ביקורת;
מספר הביקורת יהיה מספר ששווה לסכום קודי ASCII של כל תווי השורה;
תו הביקורת יהיה תו שקוד ASCII שלו הוא שארית החלוקה של מספר הביקורת ב-128;
תו של סוף השורה אמור להופיע גם בסוף של השורה האחרונה של הקובץ.

דוגמא:

תוכן הקובץ

abc 294&
3+2+1=6 351_
A 65A

טקסט מקורי

abc
3+2+1=6
A

הסבר:

שורה ראשונה. קודי ASCII של 'c', 'b', 'a' הם 97, 98, 99 בהתאמה. $294 = 97 + 98 + 99$,
 $38 = 294 \% 128$. 38 הוא קוד ה-ASCII של '&'.
שורה שנייה. סכום קודי ASCII של כל תווי השורה (ספרות ותווים לפעולות אריתמטיות) הוא 351.
 $95 = 351 \% 128$. 95 הוא קוד ה-ASCII של '_'.
שורה שלישית. קוד ה-ASCII של 'A' שהוא התו היחיד בשורה שווה ל-65. $65 = 65 \% 128$.

על הפונקציה לבדוק נכונות של מספר ביקורת ותו ביקורת בכל שורה.
אם המספר והתו נכונים בכל שורה הפונקציה תחזיר 1. אם לפחות בשורה אחת המספר או התו לא נכונים, היא תחזיר 0. אם הקובץ לא נפתח יש להחזיר -1.

גולומב 52, ת.ד. 305, חולון 5810201
טלפקס: 03-5026528

52 Golomb St., Holon 5810201 Israel

www.hit.ac.il Tel/Fax: 972-3-502-6528

הפקולטה למדעים

המחלקה למדעי המחשב

Faculty of Sciences

Department of Computer Science

```
int check_controls_in_file_rows (char *fileName)
{
    int sum=0, control_num;
    char c, control_char;
    FILE *fptr = fopen(fileName, "r");
    if( !fptr )
        return -1;
    while ( (c = fgetc(fptr)) != EOF )
    {
        if (c != ' ')
            sum += c;
        else
        {
            fscanf(fptr, "%d%c", &control_num, &control_char);
            if (control_char == '\n')
                //real control char is digit that was interpreted as last digit of
                control number
            {
                control_char = control_num%10 + '0';
                control_num /= 10;
            }
            else
                fgetc(fptr); //input '/n'
            if (control_num != sum || control_char != sum%128)
            {
                fclose(fptr);
                return 0;
            }
            else
                sum = 0;
        } //else
    } //while
    fclose(fptr);
    return 1;
}
```

```
int check_controls_in_file_rows_2 (char *fileName)
{
    int sum=0, control_sum;
    char c, last_char;
    FILE *fptr = fopen(fileName, "r");
    if( !fptr )
        return -1;
    while ( (c = fgetc(fptr)) != EOF )
    {
        if (c != ' ')
            sum += c;
        else
        {
            control_sum = fgetc(fptr) - '0'; //first digit
            last_char = fgetc(fptr);
            while ( (c = fgetc(fptr)) != '\n' )
            {
                //last_char is current digit of control_sum
                control_sum = control_sum*10 + last_char - '0';
                last_char = c;
            }
            if (control_sum != sum || last_char != sum%128)
            {
                fclose(fptr);
                return 0;
            }
            else
                sum = 0;
        }
    }
    fclose(fptr);
    return 1;
}
```

שאלה 2 (28 נק')

נתונות הגדרה הבאה של הטיפוס:

```
typedef struct item  
{  
    char *string;  
    struct item *next;  
} Item;
```

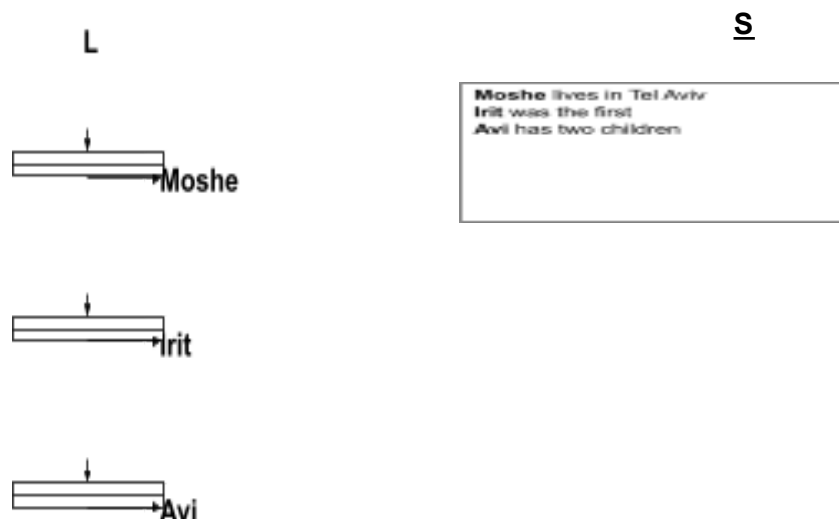
כתוב פונקציה אשר מקבלת מערך מחרוזות **S** עם גודלו ובונה רשימה מקושרת **L** של מחרוזות המורכבת מאיברים מסוג **Item**.

שדה ה-**string** של האיבר (חוליה) ה-**i** ברשימה **L** (מתחילתה) יצביע למחרוזת דינאמית החדשה המכילה את המילה הראשונה במחרוזת **S[i]** (מילה היא רצף תווים שאינם רווח המופרדים ממילים אחרות ע"י רווחים). אם **S[i]** היא מחרוזת ריקה אז המחרוזת המתאימה באיבר ה-**i** של **L** גם תהיה ריקה.

נתון כי אף מחרוזות ב-**S** אינן מתחילות ברווח. תתכן מחרוזת המורכבת ממילה יחידה בלבד.

הפונקציה תחזיר את כתובת ראש הרשימה **L**.

דוגמא:



ניתן להניח שיש בזיכרון מספיק מקום להקצאות.

```
Item *array_of_strings_to_list (char **S, int n)
{
    Item *lst, *curr_ptr;
    int i, j;
    lst = (Item *)malloc(sizeof(Item)); //dummy
    curr_ptr = lst;
    for (j=0; j<n; j++)
    {
        curr_ptr->next = (Item *)malloc(sizeof(Item));
        curr_ptr = curr_ptr->next;
        //initial allocation - for all the string with '\0'
        curr_ptr->string = (char *)malloc(strlen(S[j])+1);
        for (i=0; S[j][i]!='\0' && S[j][i]!=' '; i++)
            curr_ptr->string[i] = S[j][i];
        curr_ptr->string[i] = '\0';
        //reallocation for the first word in a string with '\0'
        curr_ptr->string = (char *)realloc(curr_ptr->string, i+1);
    }
    curr_ptr->next = NULL;
    /* deletion of dummy */
    curr_ptr = lst;
    lst = lst->next;
    free (curr_ptr);
    return lst;
}
```

שאלה 3 (28 נק')

כתוב פונקציה המקבלת שני מערכים דו-ממדיים דינאמיים **A** ו-**B** בעלי אותה כמות שורות **n**. יחד עם **n**, הפונקציה מקבלת מערכים חד-ממדיים **SA** ו-**SB** של גדלי השורות ב-**A** וב-**B** בהתאמה כך ש-**SA[i]** יהיה שווה לגודל שורה **A[i]** ו-**SB[i]** יהיה שווה לגודל שורה **B[i]**.

על הפונקציה ליצור מערך דו-ממדי דינאמי **C** בעל **n** שורות. גודל שורה **C[i]** יהיה שווה למינימום בין גדלי השורות **A[i]** ו-**B[i]** וכל איבר **C[i][j]** יהיה שווה ל-**B[i][j]+A[i][j]**.

הפונקציה תחזיר את הכתובת של מערך **C** ותעביר במצביעים (by reference) מערך דינאמי **SC** המורכב מגדלי השורות של **C** כך ש-**SC[i]** יהיה שווה לגודל שורה **C[i]**.

ניתן להניח שיש בזיכרון מספיק מקום להקצאות.

<u>SA</u>	<u>A</u>	<u>SB</u>	<u>B</u>
2	13 33	3	20 19 56
4	15 17 19 21	3	44 11 39
6	41 43 53 41 17 15	4	55 50 27 14

<u>SC</u>	<u>C</u>
2	33 52
3	59 28 58
4	96 93 80 55

```

int **sum_of_2_dim_arrays(int **A, int **SC,
int n)
{
    int i, j, **C;
    C = (int **)calloc(n, sizeof(int *));
    *SC = (int *)calloc(n, sizeof(int));
    for (i=0; i<n; i++) {
        (*SC)[i] = (SA[i]<SB[i])? SA[i] : SB[i];
        C[i] = (int *)calloc((*SC)[i], sizeof(int));
        for (j=0; j<(*SC)[i]; j++)
            C[i][j] = A[i][j]+B[i][j];
    }
}
  
```

```
}  
return C;  
}
```

שאלה 4 (16 נק')

בחר את התשובה הנכונה ונמק בכל אחד מהסעיפים הבאים (חובה לרשום את הבחירה והנימוק במחברת):

1. איזה פעולה לא מוגדרת למבנים?
 - (1) השמה
 - (2) לקיחת כתובת
 - (3) גישה לאיבר
 - (4) השוואה
 - (5) sizeof
 - (6) כל הפעולות לעיל מוגדרות

(4) – לפי כללי שפת C, כדי להימנע מהשוואת תכני המרווחים בין איברי המבנה

2. מהי הטענה הנכונה?
 - (1) אחרי ביצוע פעולת הזזה שמאלה במספר, מצד ימין המספר מתמלא ב-0-ים
 - (2) אחרי ביצוע פעולת הזזה שמאלה במספר, מצד ימין המספר מתמלא ב-1-ים
 - (3) אחרי ביצוע פעולת הזזה שמאלה במספר, מצד ימין המספר מתמלא בסיביות שהלכו לאיבוד מצד שמאל
 - (4) אחרי ביצוע פעולת הזזה שמאלה במספר, שיטת מילוי המספר מצד ימין תלויה במערכת
 - (5) אף טענה אינה נכונה

(1) – לפי כללי שפת C

3.

```
int x, *ptr1;  
const int y = 5, *ptr2;
```

איזה פעולה תהיה בלתי חוקית?

- (1) ptr1 = &x;
- (2) ptr2 = &y;
- (3) ptr1 = &y;
- (4) ptr2 = &x;

(5) כל הפעולות לעיל חוקיות

(3) – כי גישה לקבוע דרך מצביע רגיל תאפשר לשנות את הקבוע

4. איזו מילה היא לא חלק של הנחית קדם-מעבד?

- (1) define
- (2) ifdef
- (3) undef
- (4) ifndef
- (5) typedef



מכון טכנולוגי חולון
Holon Institute of Technology

(6) כל המילים לעיל הן חלקים של הנחיות קדם-מעבד

(5) – משתמשים ב- typedef להגדרת טיפוס חדש

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
    char *s="123";
    puts( (char *) ( ((long long)s*(long long)s+2*(long long)s+1)/(long long)(s+1) ) );
}
```

יש לנמק את התשובה.

23

הסבר:

Casting ל- long long ממיר כתובת (ערך s) למספר שלם.
מקבלים

$$(s^2+2s+1)/(s+1)=(s+1)^2/(s+1)=s+1$$

Casting ל- char * קובע ערך s+1 ככתובת של תו מס' 1 במחרוזת.
פלט מתו מס' 1 עד סוף המחרוזת:

23