

מבחן סדנה מתקדמת בתכנות-61108

ד"ר מרק קורנבליט וד"ר נעמה קופלמן

סמסטר ב', מועד ב', תשע"ח

תאריך: 25.07.18

הוראות:

- משך המבחן 3 שעות.

- אין להשתמש בחומרי עזר, פרט לדף המצורף לשאלון

שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת מערך דו-ממדי סטטי (לא דינאמי) A של מספרים שלמים בעל $rows$ שורות ו- $cols$ עמודות, וגם מחרוזת $fileName$. המספר הפיסי של העמודות יינתן ע"י קבוע בשם $COLS$. יש להגדיר אותו לפני הפונקציה. ידוע שכל שורה ב- A היא מערך ממוין בסדר עולה.

על הפונקציה ליצור קובץ טקסט ששמו נכלל ב- $fileName$ לשמירת מערך A בדרך חסכונית וללא איברים חוזרים בשורות באופן הבא:

שורה מס' i בקובץ תתאים לשורה מס' i של A (השורות בקובץ ממוספרות מ-0). בתחילת כל שורה בקובץ יופיע מספר השורה עם נקודה ורווח אחריו. כל סדרת איברים זהים בשורה מס' i של A תיוצג בקובץ כזוג מספרים: (1) ערך האיבר החוזר ו-(2) מספר החזרות. הזוגות בכל שורה של הקובץ יהיו ממוינים בסדר עולה על פי המספר הראשון בזוג (שהוא גם המספר שהופיע במקור במערך). המספרים בכל שורה יהיו מופרדים ברווחים.

דוגמא:

תוכן הקובץ	A
0. 2 1 3 2 6 4 9 1	2 3 3 6 6 6 6 9
1. 0 4 1 2 7 2	0 0 0 0 1 1 7 7
2. 5 8	5 5 5 5 5 5 5 5
3. 3 3 4 5	3 3 3 4 4 4 4 4



מכון טכנולוגי חולון
Holon Institute of Technology

```
#define COLS 8 //example

void compress_matrix_to_file (int A[][COLS], int rows, int cols, char
*fileName)
{
    int i, j, ser_size;
    FILE *fptr = fopen(fileName, "w");
    if (!fptr)
        exit(1);
    for (i=0; i<rows; i++)
    {
        ser_size = 1; //number of equal elements in the current
series
        fprintf(fptr, "%d.", i);
        for (j=1; j<cols; j++)
            if (A[i][j] == A[i][j-1]) //the same series
                ser_size++;
            else //new series
            {
                fprintf(fptr, " %d %d", A[i][j-1], ser_size);
                ser_size = 1;
            }
        /* last series in row i */
        fprintf(fptr, " %d %d\n", A[i][j-1], ser_size);
    }
    fclose(fptr);
}
```

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

שאלה 2 (28 נק')

נתונות ההגדרות הבאות של טיפוסים:

```
typedef struct data_item {
    int data;
    struct data_item *next;
} Dataltem;
```

```
typedef struct ptr_item {
    Dataltem *ptr;
    struct ptr_item *next;
} PtrItem;
```

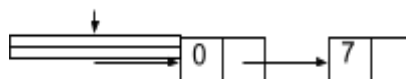
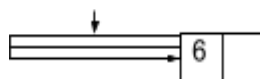
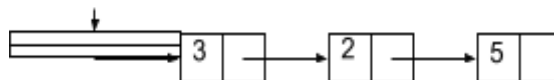
כתוב פונקציה המקבלת **רשימה מקושרת L** של **רשימות מקושרות** אשר מורכבת מאיברים מסוג **PtrItem**. השדה **ptr** של איבר מס' i ב- **L** יצביע לראש **רשימת המספרים** מס' i ושדה ה- **next** שלו יצביע לאיבר הבא ב- **L**. רשימות מספרים יהיו מורכבות מאיברים מסוג **Dataltem**.

על הפונקציה לבנות רשימה מקושרת חדשה **R** אשר מורכבת מאיברים מסוג **Dataltem** ומכילה בתוכה את כל האיברים של הרשימות של **L** (מהרשימה הראשונה ועד לרשימה האחרונה, ומהאיבר הראשון בכל רשימה עד לאיבר האחרון בתוכה).

הפונקציה תחזיר את כתובת האיבר הראשון של רשימת **R**.

L

דוגמא:



ניתן להניח שיש בזיכרון מספיק מקום להקצאות.

```

DataItem *list_of_lists_to_list (PtrItem *L)
{
    DataItem *data_item_list, *curr_ptr,
        *lst = (DataItem *)malloc(sizeof(DataItem)); //dummy
    PtrItem *temp = L;

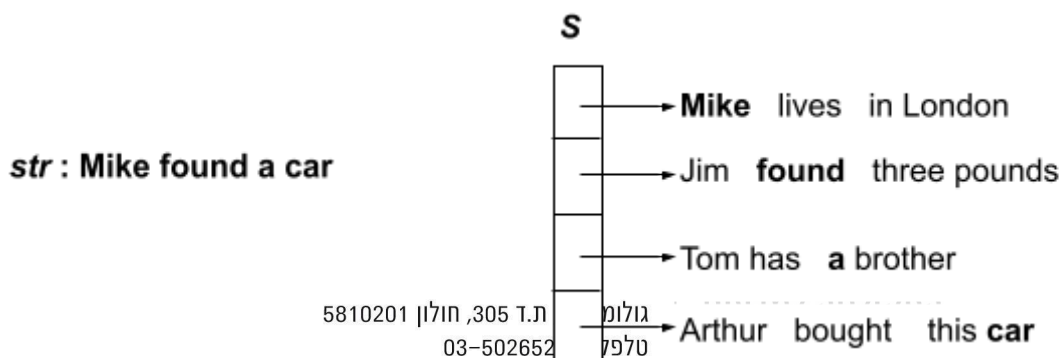
    curr_ptr = lst;
    while (temp) // scan list of lists
    {
        data_item_list = temp->ptr;
        while (data_item_list) // scan internal list
        {
            curr_ptr->next = (DataItem *)malloc(sizeof(DataItem));
            curr_ptr = curr_ptr->next;
            curr_ptr->data = data_item_list->data;
            data_item_list = data_item_list->next;
        }
        temp = temp->next;
    }
    curr_ptr->next = NULL;
    /* deletion of dummy */
    curr_ptr = lst;
    lst = lst->next;
    free (curr_ptr);
    return lst;
}
  
```

שאלה 3 (28 נק')

כתוב פונקציה שמקבלת מערך מחרוזות **S** (כמערך מצביעים לתווים) וגודלו n . כל אחת מהמחרוזות מורכבת מ- n מילים (מילה היא רצף התווים שאינם רווח המופרד ממילים אחרות ע"י רווח אחד או יותר) ולא מתחילה ברווח.

על הפונקציה ליצור מחרוזת חדשה **str** דינאמית המורכבת מהמילה הראשונה של המחרוזת **S[0]**, המילה השנייה של **S[1]**, ..., מילה n -ית של **S[n-1]** (מתחילת **str** עד סופה בהתאמה), עם רווח אחד בין המילים הסמוכות ב-**str**.

דוגמא:



הפונקציה תחזיר את כתובת המחרוזת החדשה.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
/* returns index of beginning of i-th word in sentence */
int i_th_word (char *sentence, int i)
{
    int pos=1, count=0;
    if (i==0) return 0;
    while (count<i)
    {
        pos++;
        if (sentence[pos-1]==' ' && sentence[pos]!=' ')
            count++;
    }
    return pos;
}

char *diagonal_strings (char **S, int n)
{
    int i, j, k=0, size=0, pos;
    char *str;

    for (i=0; i<n; i++)
        size += strlen(S[i]);
    str = (char *)malloc(size+1);

    /* copy i-th word in S[i] to str */
    for (i=0; i<n; i++)
    {
        pos = i_th_word(S[i], i);
        for (j=pos; S[i][j] != ' ' && S[i][j] != '\0'; j++)
            str[k++] = S[i][j];
        str[k++] = ' ';
    }

    str[k-1] = '\0';
    str = (char *)realloc(str, k);
    return str;
}
```

```
#include <stdio.h>

unsigned char fun1(unsigned char, unsigned);
unsigned char fun2(unsigned char, unsigned);

void main()
{
    unsigned char
        (*fp[2])(unsigned char, unsigned)={fun1,fun2}; // 1
    unsigned char val=123; // 2
    val = fp[0](val,8);
    val = fp[1](val,4);
    printf("%u\n",val);
}

unsigned char fun1(unsigned char value, unsigned n)
{
    value = ~value;
    value <= n;
    value = ~value;
    return value;
}

unsigned char fun2(unsigned char value, unsigned n)
{
    value = ~value;
    value >= n;
    value = ~value;
    return value;
}
```

1. מהו פלט התוכנית? נמק.

fp הוא מערך של שני מצביעים לפונקציות. fp[0] מצביע ל-fun1 ו-fp[1] מצביע ל-fun2. הפונקציה fun1 מבצעת הזזת סיביות של מספר value שמאלה ב-n סיביות. מצד ימין המספר מתמלא ב-1-ים. הפונקציה fun2 מבצעת הזזת סיביות של מספר value ימינה ב-n סיביות. מצד שמאל המספר מתמלא ב-1-ים. לכן קודם fun1 מזיזה כל 8 סיביות של val שמאלה (הן כולן הולכות לאיבוד) ומחליפה אותן בשמונה 1-ים. אחר כך fun2 מזיזה 4 סיביות של val ימינה כך שמצד שמאל המספר מתמלא בארבע 1-ים. ז"א val לא משתנה אחרי קריאה ל-fun2. הייצוג העשרוני של 11111111 הוא 255.



מכון טכנולוגי חולון
Holon Institute of Technology

לכן הפלט יהיה 255.

2. האם ישתנה הפלט בעקבות החלפת 123 ב-135 בשורה המסומנת בהערה כ-2 (//2)? נמק תשובתך. אם כן, הסבר כיצד ישתנה הפלט.

לא ישתנה כי בכל מקרה כל 8 סיביות של המספר מוחלפות ב-1-ים.

3. האם ישתנה הפלט בעקבות החלפת {fun1, fun2} ב- {fun2, fun1} בשורה המסומנת בהערה כ-1 (//1)? אם כן, הסבר כיצד ישתנה הפלט.

בעכבות ההחלפה [fp[0 יצביע ל-fun2 ו- [fp[1 יצביע ל-fun1.
לכן קודם fun2 תזיז כל 8 סיביות של val ימינה (הן כולן ילכו לאיבוד) ותחליף אותן בשמונה 1-ים. ולאחר מכן fun1 תזיז 4 סיביות של val שמאלה כך שמצד ימין המספר יתמלא בארבע 1-ים.
אז הייצוג הבינארי של המספר שוב יהיה 11111111 ולכן הפלט לא ישתנה.

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
    printf("-"); puts("printf"- "puts"? "printf-printf": "puts-puts");
}
```

יש לנמק את התשובה.

-printf-printf

הסבר:

קודם printf מדפיסה -. מחרוזות "printf" ו-"puts" נמצאות בכתובות שונות ולכן הפרש הכתובות שלהן שונה מ-0. פירוש של כל מה ששונה מ-0 זה אמת. ז"א התנאי מתקיים ו-puts תדפיס עוד את המתרחות printf-printf.