

מבחן סדנה מתקדמת בתכנות – 61108

**ד"ר מרק קורנבליט, ד"ר לאוניד קוגל,
מר מוטי רוסו, מר מרדכי חגיז, מר חיים שפיר**

סמסטר ב', מועד ב', תש"ף

תאריך: 2.08.20

הוראות:

- **משך המבחן: 3 שעות.**
- **חל איסור על שימוש בחומרי עזר, פרט, לדף המצורף לשאלון. חל איסור העזרות באחרים.**
- **על סטודנט להיות מחובר ל-ZOOM עם מצלמה דלוקה במהלך כל המבחן.**
- **ניתן להשתמש רק במכשיר אלקטרוני אחד שדרכו עושים את המבחן. אסור להשתמש במכשיר נוסף (מחשב, טאבלט, טלפון נייד, מחשבון וכו').**
- **מותר להשתמש בדפים חלקים לכתובת טיוטה. אסור להשתמש בדפים לא חלקים שהוכנו מראש.**

המבחן מורכב משלושה חלקים:

- **חלק 1 (80 נק') – כתיבת קוד. יש לענות על שתי שאלות מתוך שלוש.**
- **חלק 2 (20 נק') – ארבע שאלות אמריקאיות.**
- **חלק 3 (10 נק') – שאלת בונוס.**

חלק 1. בחלק זה יש לענות על שתי שאלות מתוך שלוש.

שאלה 1 (40 נק')

נגדיר מחרוזת כ-"ממוינת בקבוצות" אם האותיות הקטנות שבתוכה, האותיות הגדולות שבתוכה, וגם הספרות שבתוכה ממוינות בסדר עולה (לא ממש) מתחילת המחרוזת עד לסופה, לפי קודי ASCII שלהן. מדובר רק על סדר התווים בתוך כל אחת מקבוצות התווים. לא משנה כיצד התווים מהקבוצות השונות ממוקמים יחסית זה לזה. מיקום התווים שאינם שייכים לאף אחת מהקבוצות לא משנה.

לדוגמא,

המחרוזת "B4\$Ca#%ds5P8P&9zY" (משמאל לימין) היא ממוינת בקבוצות;
המחרוזת "B4\$Ca#%ds5P8P&9rY" (משמאל לימין) היא אינה ממוינת בקבוצות (קוד 's' הוא גדול מקוד 'r').

כתוב פונקציה אשר מקבלת מחרוזת ומחזירה 1 אם המחרוזת ממוינת בקבוצות ו-0 אחרת.

```
int group_sort(char *str)
{
    char last_capital, last_small, last_digit;
    int i;
    last_capital = last_small = last_digit = '\0';
    for (i = 0; str[i] != '\0'; i++)
        if (isupper(str[i]))
            if (str[i] < last_capital)
                return 0;
            else
                last_capital = str[i];
        else
            if (islower(str[i]))
                if (str[i] < last_small)
                    return 0;
                else
                    last_small = str[i];
            else
                if (isdigit(str[i]))
                    if (str[i] < last_digit)
                        return 0;
                    else
                        last_digit = str[i];

    return 1;
}
```

שאלה 2 (40 נק')

נתונה הגדרה הבאה של הטיפוס:

```
typedef struct item
{
  int data;
  struct item *next;
} Item;
```

נאמר שרשימה מקושרת L_1 בעלת n_1 איברים מסוג **Item** היא **תחילית (prefix)** של רשימה מקושרת L_2 בעלת n_2 איברים מסוג **Item** אם $n_1 \leq n_2$ ושדה `data` של איבר שנמצא במקום i מתחילת L_1 ($0 \leq i < n_1$) שווה לשדה `data` של איבר שנמצא במקום i מתחילת L_2 . רשימה ריקה היא תחילית של כל רשימה.

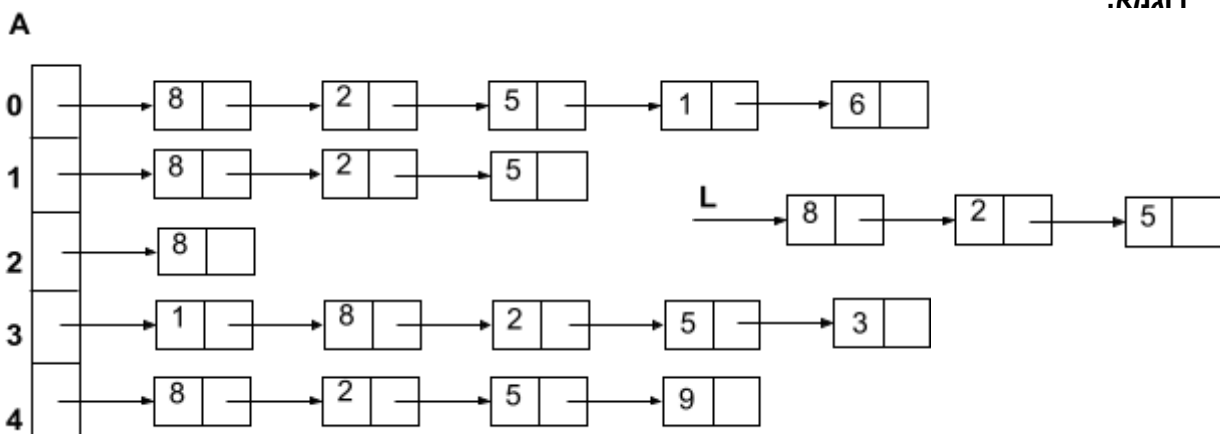
כתוב פונקציה אשר מקבלת מערך **A** של רשימות מקושרות עם גודלו וגם רשימה מקושרת נפרדת **L**. כל הרשימות מורכבות מאיברים מסוג **Item**.

על הפונקציה לגלות את כל רשימות במערך **A** ש-**L** היא תחילית שלהן. יש ליצור רשימה חדשה (אשר גם היא תהיה מורכבת מאיברים מסוג **Item**) שתכיל את האינדקסים במערך **A** של כל רשימות במערך בהם **L** היא תחילית. אין חשיבות לסדר האיברים ברשימה החדשה.

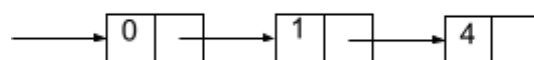
הפונקציה תחזיר את כתובת האיבר הראשון של הרשימה החדשה. אם הרשימה החדשה תהיה ריקה, הפונקציה תחזיר NULL.

ניתן להניח שיש בזיכרון מספיק מקום להקצאת איברי הרשימה החדשה.

דוגמא:



רשימה חדשה



```
//Auxiliary function - checks whether L1 is prefix of L2
int is_prefix(Item *L1, Item *L2)
{
    while (L1 && L2)
    {
        if (L1->data != L2->data)
            return 0;
        L1 = L1->next;
        L2 = L2->next;
    }
    if (L1)
        return 0;
    return 1;
}

//Auxiliary function
void insert_in_head(int x, Item **p1)
{
    Item *temp = (Item *)malloc(sizeof(Item));
    temp->data = x;
    temp->next = *p1;
    *p1 = temp;
}

Item *arr_of_lists_to_list_of_prefixes(Item **A, int n, Item *L)
{
    int i;
    Item *pref_list = NULL;
    for (i = n - 1; i >= 0; i--)
        if (is_prefix(L, A[i]))
            insert_in_head(i, &pref_list);
    return pref_list;
}
```

שאלה 3 (40 נק')

מטריצה דלילה הינה מטריצה שמרבית איבריה בעלי ערך אפס.

כתוב פונקציה (void) 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.

מחרוזת **fileName** תהווה שם של **קובץ טקסט** ששומר מטריצה דלילה בעלת **rows** שורות ו- **cols** עמודות בדרך חסכונית באמצעות אחסון רק איברי המטריצה שאינם בעלי ערך אפס. הקובץ מכיל **rows** שורות. שורה מס' **i** בקובץ מתאימה לשורה מס' **i** של המטריצה (השורות ממוספרות מ-0). בתחילת כל שורה נמצא מספרה. כל איבר של המטריצה ששונה מאפס מוצג בקובץ כזוג המספרים: ערך של האיבר ומס' עמודה בה הוא נמצא במטריצה. המספרים הסמוכים בכל שורה מופרדים ברווח אחד. התו של סוף השורה מופיע מיד אחרי המספר האחרון בכל שורה, כולל השורה האחרונה.

על הפונקציה לשחזר את המטריצה כמערך **A**. יש למלא את **A** בהתאם.

דוגמא:



תוכן הקובץ

$$\begin{bmatrix} 0 & 0 & 4 & 0 & 7 \\ 3 & 0 & 0 & 0 & 0 \\ 9 & 1 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```

0 4 2 7 4
1 3 0
2 9 0 1 1 6 3
3
  
```

```
#define COLS 5 //example

void file_to_sparse_matrix(char *filename, int A[][COLS], int
rows, int cols)
{
    int i, j, num;
    char ch;
    FILE *fptr = fopen(filename, "r");
    if (!fptr)
        exit(1);
    for (i = 0; i < rows; i++)
        for (j = 0; j < cols; j++)
            A[i][j] = 0;
    for (i = 0; i < rows; i++)
    {
        fscanf(fptr, "%d", &num); //number of row
        while ((ch = fgetc(fptr)) != '\n')
        {
            fscanf(fptr, "%d%d", &num, &j);
            A[i][j] = num;
        }
    }
    fclose(fptr);
}
```

חלק 2.

שאלות אמריקאיות (20 נק')

בחר את התשובה הנכונה ונמק בכל אחד מהסעיפים הבאים.
התשובה הנכונה ללא נימוק או עם נימוק לא נכון תזכה רק בנקודה אחת.
לפי כללי שפת C הינו גם יכול להיות נימוק מקובל.

1. מה לא מאותחל כברירת המחדל?

- (1) משתנה מסוג static
- (2) משתנה גלובלי
- (3) האיבר האחרון של מערך סטטי בו האיברים הראשונים מאותחלים באופן מפורש בהצהרה
- (4) איברי מערך דינאמי שהוקצה באמצעות פונקציית calloc
- (5) התו האחרון במערך תווים סטטי
- (6) סמן המיקום הנוכחי בקובץ ברגע פתיחתו
- (7) כל הנתונים שניתנו כאן מאותחלים כברירת המחדל

כברירת המחדל: משתנים static וגלובלי מאותחלים ב-0, איברים אחרונים במערך סטטי מאותחלים ב-0 אם איברים ראשונים מאותחלים במפורש, איברי מערך דינאמי שהוקצה באמצעות calloc מאותחלים ב-0, המיקום הנוכחי בקובץ הוא תחילת הקובץ בפתיחה לקריאה או לכתיבה וסוף הקובץ בפתיחה להוספה.
במקרה הכללי אף איבר של מערך סטטי לא מאותחל כברירת המחדל.

2. מה קורה בצד שמאל של מספר unsigned לאחר ביצוע פעולת הזזת סיביות ימינה בתוכו?

- (1) מצד שמאל המספר מתמלא ב-0-ים
- (2) מצד שמאל המספר מתמלא ב-1-ים
- (3) מצד שמאל המספר מתמלא בסיביות שהלכו לאיבוד מצד ימין
- (4) שיטת מילוי המספר מצד שמאל תלויה במערכת
- (5) אף תשובה אינה נכונה

לפי כללי שפת C.

3. מהו "abcd" - "xyz" ?

- (1) מחרוזת
- (2) כתובת
- (3) מספר שלם
- (4) הפעולה היא בלתי חוקית מבחינת המהדר
- (5) אף תשובה אינה נכונה

מחרוזת כבועה במשפט היא כתובת של התו הראשון שלה. לפי כללי אריתמטיקת מצביעים הפרש הכתובות זה מספר שלם.

4. נניח שבן-אדם פותח קובץ בינארי באמצעות עורך טקסט. מהי הטענה הנכונה לגבי נתונים הנשמרים בקובץ?

* **"לא ניתן לקריאה על ידי בני אדם"** פירושו: אדם יראה ג'יבריש ("קשקוש").

* **"ניתן לקריאה על ידי בני אדם"** פירושו: אדם יוכל לראות את הערך שבו מדובר ולא ג'יבריש.

(1) מספרים שלמים ניתנים לקריאה ע"י בן-אדם כאשר תווים לא ניתנים לקריאה ע"י בן-אדם

(2) **מספרים שלמים לא ניתנים לקריאה ע"י בן-אדם כאשר תווים ניתנים לקריאה ע"י בן-אדם**

(3) גם מספרים שלמים וגם תווים לא ניתנים לקריאה ע"י בן-אדם

(4) גם מספרים שלמים וגם תווים ניתנים לקריאה ע"י בן-אדם אם רק מספרים שלמים או רק תווים נשמרים בקובץ אולם הם לא ניתנים לקריאה אם מופיעים בקובץ ביחד

(5) גם מספרים שלמים וגם תווים תמיד ניתנים לקריאה ע"י בן-אדם

קובץ בינארי הוא רצף בתים (כמו בזיכרון פנימי). מצד שני תו זה בית אחד ולכן כל בית מופיע כתו. אזי בן-אדם יראה את התווים כמו שהם אבל מספרים אראה כקשקוש.

חלק 3.

שאלת בונוס (10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
    int n = strlen("\\\\\\"), m = strlen("\n\n0\n"), k =
    strlen("////");
    char *s = "Ah!", *x = s + m;
    puts( (char *) ( (long long) (pow((int)x, n) - k) / (int)(x + m) ) );
}
```

חובה לנמק את התשובה! התשובה ללא נימוק או עם נימוק לא נכון לא תזכה בנקודות.

Ah!

סדרת escape שהתו הראשון שלה הוא '\ מציגה לא את עצמה אלא איזה תו אחר. במיוחד 'ח' זה תו של סוף השורה, '0' זה תו של סוף המחרוזת, '\ זה תו ' עצמו. לכן מחרוזת "\\\" מורכבת מ-2



מכון טכנולוגי חולון
Holon Institute of Technology

תווים וארכה הוא 2. ב- "`\0\h\h`" יש שני תווים עד לסוף המחרוזת וארכה 2. מחרוזת "`////`" מורכבת מתווים רגילים וארכה 4. אזי $k=4$, $m=2$, $n=2$ ו- $x=s+2$.
Casting ל- `long long` ול- `int` ממיר כתובת (ערך x) למספר שלם.
 $(x^2-4)/(x+2)=(x-2)(x+2)/(x+2)=x-2=s+2-2=s$
Casting ל- `char` * קובע ערך s ככתובת של התו הראשון במחרוזת.
`puts` מקבלת מחרוזת s ומדפיסה את התוכן שלה "`!Ah`".