

מבחן סדנה מתקדמת בתכנות - 61108

ד"ר מרק קורנבליט, ד"ר לאוניד קוגל, מר טוביה דב רוזנברג

סמסטר ב', מועד א', תשפ"ב

תאריך: 23.06.22

הוראות:

- משך המבחן 3 שעות.
- אין להשתמש בחומרי עזר, פרט לדף המצורף לשאלון.

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

שאלה 1 (40 נק')

כתבו פונקציה בשם **duplicateString** אשר מקבלת מחרוזת בסיס **base**, מחרוזת אמצע **middle** ומספר שלם **times**.

על הפונקציה ליצור ולהחזיר מחרוזת דינאמית חדשה המהווה שרשור של מחרוזת הבסיס עם מחרוזת האמצע לסירוגין.

מחרוזת הבסיס תופיע **times** פעמים כתת-מחרוזת במחרוזת החדשה. מחרוזת האמצע לא תהיה בתחילת ובסוף המחרוזת החדשה ותופיע כתת-מחרוזת רק בין מחרוזת הבסיס.

לדוגמא, עבור $times = 3$ המחרוזת החדשה תיראה כך:

base	middle	base	middle	base
------	--------	------	--------	------

ניתן להניח שיש בזיכרון מספיק מקום להקצאת המחרוזת.

```

char* duplicateString(char* base, char* middle, int times)
{
    // zeroed dynamic array
    char* str = calloc(times * strlen(base) + (times - 1) * strlen(middle) + 1,
sizeof(char));
    int i;

    for (i = 1; i <= times - 1; i++)
    {
        strcat(str, base);
        strcat(str, middle);
    }
    strcat(str, base);
    return str;
}
  
```

שאלה 2 (40 נק')

נתון מספר חיובי שלם M המיוצג באמצעות רשימה מקושרת L , כך שכל איבר ברשימה מכיל ספרה אחת של המספר (הספרה המשמעותית ביותר של $M -$ באיבר הראשון של L , ..., הספרה הכי פחות משמעותית של $M -$ באיבר האחרון של L).

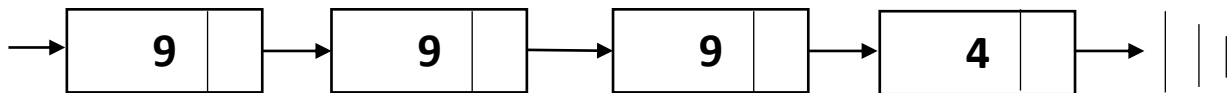
כתבו פונקציה void בשם **List_num_add** המקבלת מצביע לראש הרשימה L ומספר חיובי שלם N .

על הפונקציה להגדיל ב- N את המספר M המיוצג ע"י L (כלומר להפוך L לרשימה מקושרת שתיוצג את המספר $M+N$). במידת הצורך להאריך את הרשימה (במקרה זה ניתן להניח שיש בזיכרון מספיק מקום להקצאה).

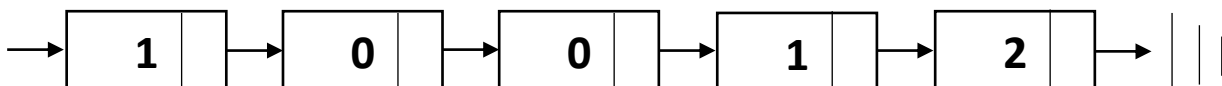
שימו לב. אין ליצור רשימה חדשה אלא רק לעדכן את הרשימה המקורית.

דוגמא:

המספר 9994 ירשם כמתואר להלן, כך שהספרה הראשונה מופיעה בראש הרשימה והאחרונה בזנבה:



לאחר הוספת הערך 18, הרשימה תראה כך:



יש להגדיר את הטיפוס של איברי הרשימה מעל הפונקציה.

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

/* Auxiliary function: estimation of the size of a number */
int countDigit(int number)
{
    int j = 0;
    while (number)
    {
        ++j;
        number /= 10;
    }
    return j;
}
```

גרסה 1

```
void List_num_add_1(Item* head, int N)
{
    int i, p, k, new_size, number = 0;
    Item* node = head;

    /* Creating a number from the list */
    while (node)
    {
        number = number * 10 + node->data;
        node = node->next;
    }

    number += N; //new number
    new_size = countDigit(number); //the size of a new number

    /* List transformation for a new number */
    p = (int)pow(10, new_size);
    node = head;
    for (i = 0; i < new_size; ++i)
    {
        p /= 10;
        k = number / p;
        node->data = k;
        number = number - k * p;
        if (!node->next && i < new_size - 1) //additional element
            node->next = (Item*)calloc(1, sizeof(Item));
        node = node->next;
    } //for
}
```

```
void List_num_add(Item* head, int N)
{
    int i, p, k, size = 0, new_size, number = 0, diff;
    Item* node = head;

    /* Creating a number from the list */
    while (node)
    {
        number = number * 10 + node->data;
        node = node->next;
        size++;
    }

    number += N; //new number
    new_size = countDigit(number); //the size of a new number
    diff = new_size - size;

    /* Addition of new elements for a new number */
    if (diff > 0)
    {
        node = head;
        while (node->next) //search of a last element
            node = node->next;
        for (i = 1; i <= diff; i++) //diff new elements
        {
            node->next = (Item*)malloc(sizeof(Item));
            node = node->next;
        }
        node->next = NULL;
    }

    /* List filling for a new number */
    p = (int)pow(10, new_size);
    node = head;
    while (node)
    {
        p /= 10;
        k = number / p % 10;
        node->data = k;
        node = node->next;
    }
}
```

שאלה 3 (40 נק')

נתונה הגדרת המבנה הבאה:

```
typedef struct {
    char address[30];
    int rooms;
    int price;
} apartment;
```

המתארת מידע על דירה למכירה במשרד תיווך, כאשר:

- address – הכתובת של הדירה למכירה,
- rooms – מספר החדרים בדירה,
- price – מחיר הדירה.

כתבו פונקציה בשם **appropriateApartment** אשר מקבלת מחרוזת **fileName** המהווה שם של **קובץ בינארי**. הקובץ מורכב מרשומות הנתונים שהפורמט של כל אחת מהן מתאים למבנה apartment. כמו כן, הפונקציה מקבלת כפרמטרים את מספר החדרים הרצוי לקונה (מספר שלם **room_number**) ואת המחיר המקסימאלי שהקונה מוכן לשלם (מספר שלם **max_price**).

על הפונקציה ליצור מערך דינאמי של מבני apartment. המערך יכיל את נתוני כל הדירות של הקובץ שמחירן אינו עולה על max_price ומספר חדריהן שווה ל-room_number. הפונקציה תחזיר את כתובת תחילת המערך ותעביר (by reference) את גודלו. אם אף דירה בקובץ לא מתאימה לתנאים לעיל או הקובץ לא נפתח, הפונקציה תחזיר NULL ותעביר 0.

דוגמא:

עבור הקובץ הבא:

address	Begin 7	Reznik 4	Ben-Gurion 27	Dubnov 13	Cohen 66
rooms	3	4	5	3	3
price	700,000	1,000,000	4,000,000	1,100,000	900,000

i- room_number=3, max_price=1,000,000

הפונקציה תיצור את המערך

Begin 7	Cohen 66
3	3
700,000	900,000

ותעביר 2.

ניתן להניח שיש בזיכרון מספיק מקום להקצאת המערך.

```
apartment* appropriateAppartment(char* fileName, int room_number, int max_price,
int* count)
{
    int i=0;
    FILE* fptr = fopen(fileName, "rb");
    apartment* approp_apart_units, unit;
    *count = 0;
    if (!fptr)
        return NULL;

    /* Estimation of array size */
    fread(&unit, sizeof(apartment), 1, fptr);
    while (!feof(fptr))
    {
        if (unit.rooms == room_number && unit.price <= max_price)
            (*count)++;
        fread(&unit, sizeof(apartment), 1, fptr);
    }

    if (*count == 0)
    {
        fclose(fptr);
        return NULL;
    }

    approp_apart_units = (apartment*)calloc(*count, sizeof(apartment)); //array
allocation

    /* Filling of array */
    rewind(fptr);
    fread(&unit, sizeof(apartment), 1, fptr);
    while (!feof(fptr))
    {
        if (unit.rooms == room_number && unit.price <= max_price)
            approp_apart_units[i++] = unit;
        fread(&unit, sizeof(apartment), 1, fptr);
    }

    fclose(fptr);
    return approp_apart_units;
}
```

חלק 2.

שאלה 4 (20 נק')

בחרו את התשובה הנכונה ונמקו בכל אחד מהסעיפים הבאים (חובה לרשום את הבחירה והנימוק במחברת):

א. נתון קובץ טקסט בעל התוכן הבא: **1 2 3 4 5** (משמאל לימין). אחרי כל אחד מחמישה המספרים (גם אחרי 5) יש רווח. מיד אחרי הפתיחה המוצלחת של הקובץ באמצעות מצביע fptr מופיע הקטע הבא בתוכנית (x הוא משתנה מסוג int):

```
while (!feof(fptr)) {  
    fscanf(fptr, "%d", &x);  
    printf("%d ", x);  
}
```

כמה פעמים מבוצע גוף הלולאה?

- (1) 5 פעמים
- (2) 4 פעמים
- (3) מספר אינסופי פעמים
- (4) פעם אחת

(5) אף תשובה אינה נכונה

הקוד מאורגן בצורה לא תקינה – ללא קלט לפני בדיקת סוף הקובץ. כתוצאה אחרי טיפול בחמישה המספרים השמורים ניכנס לתוך הלולאה גם בפעם השישי כאשר נהיה כבר בסוף הקובץ.

```
double (**q) ();
```

ב.

על מה מצביע פוינטר q?

- (1) על פוינטר ל- double
- (2) על פונקציה
- (3) על פוינטר לפונקציה
- (4) על פוינטר לפוינטר ל- double
- (5) q זה לא פוינטר אלא פונקציה
- (6) אף תשובה אינה נכונה

לפי תחביר שפת C, q זה פוינטר לפוינטר לפונקציה ללא פרמטרים אשר מחזירה double. ז"א, q מצביע על פוינטר לפונקציה.

ג. נתון פוינטר ל- int בשם p. בחרו מה יקרה אחרי ביצוע פעולת

```
p = (int*)malloc(0);
```

- (1) malloc תקצה ביית אחד בעל אינדקס 0
- (2) p יצביע על איזשהו מקום
- (3) p יהיה שווה ל- NULL
- (4) הפעולה תגרום לשגיאת ריצה
- (5) אף תשובה אינה נכונה

במידה והפרמטר של malloc שווה ל- 0, הפונקציה לא מקצה זיכרון אבל לא מחזירה NULL.



מכון טכנולוגי חולון Holon Institute of Technology

ד. בתוכנית נתונה הצהרה הבאה:
`char *str = "hello";`
 בחרו מה יקרה אחרי ביצוע פעולת
`str = "good-bye";`

(1) ישתנה ערך של מצביע `str`

(2) תוכן המערך `str` ישתנה ל- "good-bye"

(3) תוכן המערך `str` ישתנה ל- "good"

(4) תוכן המערך `str` ישתנה ל- "good-byehello"

(5) הפעולה תגרום לשגיאת ריצה

(6) אף תשובה אינה נכונה

לא מוגדר פה מערך ו- `str` הוא מצביע. עבור הגדרה זו של `str` המחשב מקצה שטח זיכרון ל- `str` ולמחרוזת "hello" ומציב ל- `str` את כתובת התו הראשון של המחרוזת.

חלק 3.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
    int z='z'-1;
    putchar(z-(int)pow((z^1)-z,z<<1));
}
```

יש לנמק את התשובה.

X

בשורה הראשונה משתנה `z` מקבל ערך שקטן ב-1 מקוד ASCII של 'z', ז"א קוד ASCII של 'y'. ייצוג בינארי של 1 מורכב מאפסים וסיבית 1 בסוף. לפי כללי פעולת $(XOR)^a = a^0$, $a^1 = a$. לכן z^1 שונה מ- `z` רק בסיבית ימנית ולכן הפרש ביניהם הוא 1 או -1. אחרי הזזת שמאלה סיביות של `z`, הסיבית הימנית של `z` תהיה 0 ולכן זה מספר זוגי. לכן z^1 בחזקת $z << 1$ שווה ל- 1. `z` שווה לקוד ASCII של 'y' ואז `z-1` שווה לקוד ASCII של 'x'. לכן הפלט הוא x.

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