

תכנות מערכות בשפת C ושפת סף

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני חלקים: תכנות מערכות בשפת C ושפת סף ובהם שמונה שאלות. עליך לענות על שש שאלות, על-פי ההנחיות שבכל פרק. בסך-הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: כל חומר עזר כתוב בכתב-יד או מודפס על נייר.

ד. לנוחותך, לשאלון זה מצורף מילון מונחים בשפות עברית, אנגלית, רוסית וערבית. תוכל להיעזר בו בעת הצורך.

בשאלון זה 35 עמודים ו-3 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות הן לנבחנות והן לנבחנים.

בהצלחה!

◀ המשך מעבר לדף

השאלות

חלק א': תכנות מערכות בשפת C (50 נקודות)

פרק ראשון (35 נקודות)

ענה על שתי השאלות 1-2.

שאלה 1 – שאלת חובה (20 נקודות)

סטודנטים נעזרים בחבריהם ברשת האינטרנט כדי לקבל עצות בנושאים שונים הנלמדים באוניברסיטה. ואולם, בכל תחום לימוד ישנם כמה סטודנטים הנותנים עצות שגויות ברשת כדי להטעות את חבריהם. סטודנטים אלו מכונים "טרולים".

הוחלט להקים מערכת ממוחשבת שתשמור מידע על הטרולים הפועלים ברשת, וכן על הפוסטים שהם מפרסמים, ועל-ידי כך תוכל לעקוב אחריהם.

המערכת מסדרת את הטרולים בקבוצות נפרדות, על פי תחום הלימוד שבו עוסק הפוסט. טרול שייך לקבוצה אחת לכל היותר, וחייב להשתייך לקבוצה כלשהי.

בעבור כל פוסט יישמר, בין היתר, מספר סימוני ה-LIKE שקיבל, כאשר מספר סימוני ה-LIKE הוא כמות הסטודנטים שהתייחסו לפוסט.

לפניך תיאור של **מבנה הנתונים**, אשר תומך במימוש הפעולות הנדרשות מן המערכת הממוחשבת (הפעולות יפורטו בהמשך).

נחזיק מבנה (רשומה), שעליו מצביע DS, ואת המבנה נכנה בשם "**המבנה הראשי**", המכיל את השדות האלה:

שדה 1: trollsArr – מערך דינמי של טרולים

כל תא במערך זה מייצג טרול, ומכיל, בין היתר, מצביע לראש רשימה מקושרת, המייצגת את הפוסטים שפרסם טרול זה בלבד.

מערך הטרולים ממוין בסדר עולה, על פי מספר הטרול.

שדה 2: postsArr – מערך דינמי של פוסטים

כל תא במערך זה מייצג פוסט, ומכיל, בין היתר, את מספר הלייקים שקיבל פוסט זה.

מערך הפוסטים ממוין בסדר עולה, על פי מספר הפוסט.

שדה 3: groupsArr – מערך דינמי של קבוצות

כל תא במערך זה מייצג קבוצה של טרולים, ומכיל, בין היתר, מצביע לראש רשימה מקושרת, המייצגת את הטרולים שבקבוצה זו בלבד.
התא ה־i במערך זה מייצג את הקבוצה שמספרה i.

בנספח א' מוצג התיאור הסכמתי של "המבנה הראשי", המאגד את כל מבני הנתונים שבהם מאחסנים את נתוני המערכת הממוחשבת. היעזר בו.

להלן הגדרת המבנה הראשי בשפת C :

```
typedef struct headType    // טיפוס המבנה הראשי
{
    trollPtr trollsArr;      // מערך דינמי של טרולים
    postPtr postsArr;        // מערך דינמי של פוסטים
    groupPtr groupsArr;      // מערך דינמי של קבוצות

} header, *headPtr;
```

עתה נפרט את מבנה הנתונים בעבור **שדה 2** של "**המבנה הראשי**" – **מערך דינמי של פוסטים**.

להלן מבנה של תא **במערך הדינמי של הפוסטים** בשפת C :

```
typedef struct postType    // טיפוס פוסט
{
    int postID;             // מספר הפוסט
    long trollID;           // מספר הטרול שפרסם את הפוסט
    int likes;              // מספר הלייקים שקיבל פוסט זה
} postRec, *postPtr;
```

עתה נפרט את מבנה הנתונים "**צומת ברשימה מקושרת של פוסטים**", כאשר צומת מייצג פוסט של טרול כלשהו.

שים לב: לכל טרול יש רשימה מקושרת נפרדת עבור הפוסטים שפרסם.

```
typedef struct postNodeType
{
    int postID;             // מספר הפוסט
    struct postNodeType *next; // שדה קישור
} postNodeRec, *postNodePtr;
```

עתה נפרט את מבנה הנתונים בעבור **שדה 1** של "**המבנה הראשי**" – **מערך דינמי של טרולים**.
להלן מבנה של תא **במערך הדינמי של הטרולים** בשפת C:

```
typedef struct trollType      // טיפוס טרול
{
    long trollID;              // מספר הטרול
    char name[20];             // שם הטרול
    int groupNo;               // מספר הקבוצה שהטרול שייך לה
    long strength;            // עוצמת הטרול
    postNodePtr lnp;          // מצביע לראש רשימת הפוסטים של טרול זה
} trollRec, *trollPtr;
```

עתה נפרט את מבנה הנתונים "**צומת ברשימה מקושרת של טרולים**", כאשר צומת מייצג טרול השייך לקבוצה כלשהי.

שים לב: לכל קבוצה יש רשימה מקושרת נפרדת של הטרולים השייכים לה.

```
typedef struct trollNodeType
{
    long ID;                   // מספר הטרול
    struct trollNodeType *next; // שדה קישור
} trollNodeRec, *trollNodePtr;
```

עתה נפרט את מבנה הנתונים בעבור **שדה 3** של "**המבנה הראשי**" – **מערך דינמי של קבוצות**.
להלן מבנה של תא **במערך הדינמי של הקבוצות** בשפת C:

```
typedef struct groupType      // טיפוס קבוצה
{
    int trollNum;              // מספר הטרולים השייכים לקבוצה זו
    trollNodePtr gp;           // מצביע לראש רשימת הטרולים השייכים לקבוצה זו
} groupRec, *groupPtr;
```

להלן הגדרות התקפות לכל הסעיפים שיבואו בהמשך:

```
typedef enum {FAILURE, SUCCESS, INVALID_INPUT, ALLOCATION_ERROR}  
statusType;  
  
typedef enum {FALSE, TRUE} boolean;
```

נתונה ספריית פונקציות המכילה, בין היתר, את הפונקציות האלה:

פונקצייה זו מחפשת במערך הטרולים את הטרול שמספרו ID . אם הטרול שמספרו ID נמצא במערך זה, אזי הפונקצייה מחזירה את מיקומו במערך; אחרת - היא מחזירה את הערך (-1) .	int findtroll(long ID)
פונקצייה זו מחפשת במערך הפוסטים את הפוסט שמספרו ID . אם הפוסט שמספרו ID נמצא במערך זה - אזי הפונקצייה מחזירה את מיקומו במערך, אחרת - היא מחזירה את הערך (-1) .	int findpost(long ID)

הנח שהפונקציות האלה כתובות, וניתן להשתמש בהן בכל הסעיפים הבאים בלי לכתוב אותן מחדש. כמו כן, בעבור כל סעיף ניתן להשתמש בכל פונקצייה שמומשה בסעיפים שלפניו.

להלן הגדרות של משתנים גלובליים:

```
header head;  
  
headPtr DS = &head;  
  
int numOfGroups;    // מספר הקבוצות הקיימות בכל רגע במערכת הממוחשבת  
int numOfTrolls;    // מספר הטרולים הקיימים בכל רגע במערכת הממוחשבת  
int numOfPosts;     // מספר הפוסטים הקיימים בכל רגע במערכת הממוחשבת
```

ענה על הסעיפים שלהלן:

א. לפניך פונקצייה שכותרתה:

```
statusType addTroll(long ID, char name[], int group, long strength)
```

פונקצייה זו מקבלת את הפרמטרים האלה בעבור טרול חדש:

ID – מספר הזהות של הטרול,

name – שם הטרול,

group – מספר הקבוצה שהטרול שייך לה,

strength – עוצמת הטרול (מידת חוזקו).

הפונקצייה אמורה להוסיף טרול חדש זה למערכת הממוחשבת.

נוסף על כך הפונקצייה:

- מעדכנת את מספר הטרולים הרשומים במערכת.
 - ממיינת את מערך הטרולים.
 - מעדכנת את מספר הטרולים השייכים לקבוצה group.
- הנח כי הטרול שמספרו ID לא קיים במערכת הממוחשבת.

הפונקצייה מחזירה ערך מטיפוס statusType כמפורט בטבלה שלהלן:

ALLOCATION_ERROR	אם קיימת בעיה בהקצאת זיכרון
INVALID_INPUT	אם הקבוצה group לא קיימת במערכת הממוחשבת
SUCCESS	אם הטרול שמספרו ID נוסף בהצלחה למערכת הממוחשבת

בפונקציות שלהלן חסרים יחד **שבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (7), בסדר עולה, וכתוב ליד כל
מספר את הביטוי החסר שהוא מייצג.

```
statusType addTroll(long ID, char name[], int group, long strength)
{
    statusType status = SUCCESS;

    trollNodePtr tr;

    int n;

    if(group < 0 || group >= numOfGroups) return INVALID_INPUT;

    n = numOfTrolls;

    if(n == 0)

        DS->trollsArr = _____(1)_____;

    else

    {

        DS->trollsArr = _____(2)_____;

    }

    if(DS->trollsArr == NULL) return ALLOCATION_ERROR;

    DS->trollsArr[n].trollID = ID;

    _____(3)_____;

    DS->trollsArr[n].groupNo = group;

    DS->trollsArr[n].strength = strength;

    DS->trollsArr[n].lnp = NULL;

    numOfTrolls++;
}
```



```
qsort(DS->trollsArr, numOfTrolls, ____ (4) ____, ____ (5) ____);  
// שיוך טרול לקבוצה  
tr = malloc(sizeof(trollNodeRec));  
if(tr == NULL) return ALLOCATION_ERROR;  
tr->ID = ID;  
tr->next = ____ (6) ____;  
____ (6) ____ = tr;  
Ds->groupsArr[group].trollNum++;  
return status;  
}
```

כאשר:

```
int intcmp(const void *v1, const void *v2)  
{  
    trollPtr p = v1;  
    trollPtr q = v2;  
    return (____ (7) ____);  
}
```

ב. לפניך פונקצייה שכותרתה:

```
statusType groupInfo(int group)
```

פונקצייה זו מקבלת את הפרמטר `group`, שהוא מספר קבוצה, ומציגה את מספר הטרולים השייכים לקבוצה זו.

נוסף על כך, בעבור כל טרול השייך לקבוצה `group` הפונקצייה מציגה את שמו ואת עוצמתו.

הפונקצייה מחזירה ערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

INVALID_INPUT	אם הקבוצה <code>group</code> לא קיימת במערכת הממוחשבת
SUCCESS	אם הקבוצה <code>group</code> קיימת במערכת הממוחשבת

בפונקצייה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
statusType groupInfo(int group)
```

```
{  
    statusType status = SUCCESS;  
    groupRec p;  
    trollNodePtr s;  
    int index;  
    if(group < 0 || group >= numOfGroups) return INVALID_INPUT;  
    p = ____ (1) ____;  
    printf("\n Numbber of trolls %d\n", p.trollNum);  
    s = p.gp;  
    while(s)
```

```
{  
    index = ____ (2) ____;  
    printf("\n\t%s \t%d \n", ____ (3) ____ , ____ (4) ____ );  
    s = s->next;  
}  
return status;  
}
```

ג. לפניך פונקצייה שכותרתה:

```
statusType removeTroll(long ID)
```

פונקצייה זו אמורה להסיר את הטרול שמספרו ID מן המערכת הממוחשבת.

נוסף על כך, הפונקצייה:

- מעדכנת את מספר הפוסטים הקיימים במערכת.
- ממיינת את מערך הפוסטים.
- מעדכנת את מספר הטרולים הקיימים במערכת.
- ממיינת את מערך הטרולים.
- מעדכנת את מספר הטרולים בקבוצה שהטרול שייך לה.

הפונקצייה מחזירה את הערך מטיפוס statusType כמפורט בטבלה שלהלן:

ALLOCATION_ERROR	אם קיימת בעיה בהקצאת זיכרון
INVALID_INPUT	אם הטרול שמספרו ID לא קיים במערכת הממוחשבת
SUCCESS	אם הטרול שמספרו ID הוסר בהצלחה מן המערכת הממוחשבת

בפונקציית חסרים **שלושה-עשר** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (13), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
statusType removeTroll(long ID)
{
    statusType status = SUCCESS;
    trollNodePtr p, q;
    postNodePtr r, t;
    int index, index1, group;
    index = ____ (1) ____;
    if(index == -1) return INVALID_INPUT;
    // הסרת הטרוול מן הקבוצה שהוא שייך לה
    group = ____ (2) ____;
    p = ____ (3) ____;
    if(p && p->ID == DS->trollsArr[index].trollID)
    {
        ____ (3) ____ = p->next;
        free(p);
    }
    else
    {
        while(p)
        {
            q = p;
            p = p->next;
            if(p->ID == DS->trollsArr[index].trollID)
            {
                ____ (4) ____;
                free(p);
                break;
            }
        }
    }
}
```

```
DS->groupsArr[group].trollNum--;  
// הסרת כל הפוסטים של הטורל  
r = _____(5) _____;  
while(r)  
{  
    index1 = _____(6) _____;  
    DS->postsArr[index1].postID = DS->postsArr[numOfPosts-1].postID;  
    DS->postsArr[index1].trollID = DS->postsArr[numOfPosts-1].trollID;  
    DS->postsArr[index1].likes = DS->postsArr[numOfPosts-1].likes;  
    _____(7) _____;  
    if(numOfPosts == 0) DS->postsArr = NULL;  
    else  
        DS->postsArr = _____(8) _____;  
    t = r;  
    r = r->next;  
    free(t);  
}  
if(DS->postsArr)  
    _____(9) _____;  
    _____(10) _____ = NULL;  
// הסרת הטורל ממערך הטורלים  
DS->trollsArr[index].trollID = DS->trollsArr[numOfTrolls-1].trollID;  
strcpy(DS->trollsArr[index].name, DS->trollsArr[numOfTrolls-1].name);  
DS->trollsArr[index].groupNo = DS->trollsArr[numOfTrolls-1].groupNo;  
DS->trollsArr[index].strength = DS->trollsArr[numOfTrolls-1].strength;  
DS->trollsArr[index].lnp = DS->trollsArr[numOfTrolls-1].lnp;  
    _____(11) _____;  
if(numOfTrolls == 0) DS->trollsArr = NULL;  
else  
    DS->trollsArr = _____(12) _____;  
if(DS->trollsArr)  
    _____(13) _____;  
return status;  
}
```

המשך השאלה בעמוד הבא.

```
int intcmp(const void *v1, const void *v2)
{
    trollPtr p = v1;
    trollPtr q = v2;
    return (p->trollID - q->trollID);
}
```

```
int intcmp1(const void *v1, const void *v2)
{
    postPtr p = v1;
    postPtr q = v2;
    return (p->postID - q->postID);
}
```

ד. לפניך פונקצייה שכותרתה:

```
statusType transferTroll(long ID, int group)
```

פונקצייה זו מקבלת את הפרמטרים האלה:

ID – מספר הזהות של טרול,

group – מספר הקבוצה שיש להעביר אליה את הטרול.

פונקצייה זו מעבירה את הטרול שמספרו ID אל הקבוצה שמספרה group .

הפונקצייה מחזירה את הערך מטיפוס statusType כמפורט בטבלה שלהלן:

ALLOCATION_ERROR	אם קיימת בעיה בהקצאת זיכרון
INVALID_INPUT	אם הקבוצה group לא קיימת במערכת הממוחשבת, או אם הטרול שמספרו ID לא קיים במערכת הממוחשבת
SUCCESS	אם הטרול שמספרו ID הועבר בהצלחה אל הקבוצה group

בפונקצייה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (5), בסדר עולה, וכתוב ליד
כל מספר את הביטוי החסר שהוא מייצג.

```
statusType transferTroll(long ID, int group)
{
    int index;
    statusType status;
    char name[20];
    long strength;
    index = ____ (1) ____;
    if(index == -1) return INVALID_INPUT;
    ____ (2) ____;
    strength = ____ (3) ____;
    status = ____ (4) ____;
    if(status == SUCCESS)
        status = ____ (5) ____;
    return status;
}
```

שאלה 2 - שאלת חובה (15 נקודות)

הסעיפים א'–ב' מתייחסים לתיאור שלהלן:

בסעיפים א'–ב' מתייחסים לאותיות האנגליות הקטנות 'z' – 'a' כאל ספרות, כאשר ערכה של האות 'a' יהיה 0, ערכה של האות 'b' יהיה 1, ערכה של האות 'c' יהיה 2, וכן הלאה. מכאן – ערכה של האות 'z' יהיה 25.

אותיות אלו ישמשו לייצוג מספרים ממשיים חיוביים (גדולים או שווים ל-0), בבסיס 26.

מספר ייוצג כמחרוזת בפורמט הזה: $a_{k-1}a_{k-2}\dots a_1a_0 \bullet a_{-1}a_{-2}\dots a_{-p}$, כאשר a_i מייצגת אות אנגלית קטנה. המספר מכיל k "ספרות" לפני הנקודה ועוד p "ספרות" אחרי הנקודה, ולצורך השאלה ניתן להניח כי $k \geq 1$ וגם $p \geq 1$.

ערכו של המספר בייצוג זה נתון על ידי הביטוי:

$$a_{k-1} \cdot 26^{k-1} + a_{k-2} \cdot 26^{k-2} + \dots + a_0 \cdot 26^0 + a_{-1} \cdot 26^{-1} + \dots + a_{-p} \cdot 26^{-p}$$

דוגמאות:

ערכו של המספר המיוצג על ידי המחרוזת "a.a" הוא $0 \cdot 26^0 + 0 \cdot 26^{-1} = 0.0$.

ערכו של המספר המיוצג על ידי המחרוזת "ba.c" הוא $1 \cdot 26^1 + 0 \cdot 26^0 + 2 \cdot 26^{-1} = 26.077$.

א. לפניך פונקצייה שמקבלת מחרוזת s, המייצגת מספר חיובי בבסיס 26, ומחזירה את ערכו של המספר הזה כ-double.

הנח כי:

- המחרוזת s תקינה.
- הפונקצייה סורקת את המחרוזת s פעם אחת בלבד.

בפונקצייה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4), בסדר עולה, וכתוב ליד
כל מספר את הביטוי החסר שהוא מייצג.

```
double strval(char *s)
{
    double base = 1, result = 0;
    while(_____(1)_____)
    {
        result *= 26;
        result += (_____(2)_____) ;
        ++s;
    }
    while(_____(3)_____ != 0)
    {
        _____(4)_____ ;
        result += (_____(2)_____) *base;
    }
    return result;
}
```

ב. נתונים שני מערכים, a ו- b , שכל אחד מהם הוא מערך של מחרוזות, וכל מחרוזת מייצגת מספר בבסיס 26, כמתואר בסעיף א'.

שני המערכים ממוינים בסדר עולה, על פי ערכן המספרי של המחרוזות שהם מכילים. לפיכך פונקציית המקבלת את שני המערכים **הממוינים** a ו- b , שגודלם n ו- m בהתאמה, וכן מספר ממשי x .

הפונקצייה מחזירה 1 אם במערכים a ו- b קיימות שתי מחרוזות כלשהן $a[i]$ ו- $b[j]$, כך ש- $a[i] - b[j] == x$.

אם לא קיימות שתי מחרוזות כאלה – הפונקצייה תחזיר את הערך 0.

בפונקצייה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (5), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int finddiff(char* a[], int n, char* b[], int m, float x)
{
    int i = 0, j = 0;
    double diff;
    while(i < n && j < m)
    {
        diff = _____ (1) _____ - _____ (2) _____;
        if(_____ (3) _____) return 1;
        if(diff < x) _____ (4) _____;
        else _____ (5) _____;
    }
    return 0;
}
```

הסעיפים ג'-ד' מתייחסים לתיאור שלהלן:

לפניך תיאור של הדופק (קצב פעימות הלב) של שחקן במבחן הקבלה הראשון בחייו לתיאטרון הלאומי.

בדקה הראשונה למבחן, הדופק הוא 57 פעימות לדקה.

בדקה השנייה למבחן, הדופק הוא 60 פעימות לדקה.

בכל דקה שהיא כפולה של 3, הדופק עולה, והוא גדול ב-2 פעימות מן הדופק שבדקה הקודמת. **לדוגמה:** בדקה השלישית למבחן, הדופק יהיה 62 פעימות לדקה (2 + 60).

בכל דקה אחרת – הדופק הוא הממוצע של הדופק שנמדד בשתי הדקות הקודמות + 2. **לדוגמה:** בדקה הרביעית הדופק של הנבחן יהיה: $63 = 2 + \frac{60+62}{2}$.

ג. לפניך פונקצייה **רקורסיבית** אשר מקבלת כפרמטר מספר דקה - n, ומחזירה את הדופק (קצב פעימות הלב) של השחקן בדקה ה-n של מבחן הקבלה.

בפונקצייה חסרים **שלושה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (3), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
double heartbeat(int n)
{
    if(n == 1) return 57;
    if(n == 2) return 60;
    if(n % 3 == 0)
        return _____ (1) _____;
    return ( _____ (2) _____ + _____ (3) _____ ) / 2 + 2;
}
```

ד. ידוע ששחקן מאבד את הכרתו אם הדופק שלו מגיע ל- m ומעלה.

לפניך פונקצייה אשר מקבלת כפרמטר את הערך m (רמת הדופק שהחל ממנו יאבד השחקן את הכרתו).

הפונקצייה מחזירה את משך הזמן המרבי, בדקות, שבו שחקן מסוגל להיבחן בלי שיאבד את הכרתו.

שים לב:

- הדופק הוא סדרה עולה ממש, כלומר: $\text{heartbeat}(n) \geq \text{heartbeat}(n-1) + 1$
- הדופק בדקה ה- m גדול ממש מ- m , כלומר: $\text{heartbeat}(m) > m$, ולכן מספר הדקה המבוקשת הוא בהכרח קטן מ- m .

הערות:

1. בסעיף זה אתה רשאי לזמן את הפונקצייה heartbeat מסעיף ג'.
2. אם בדקה n כלשהי הדופק של השחקן הוא m בדיוק, אזי השחקן מסוגל להיבחן רק $n-1$ דקות (בלי שיאבד את הכרתו).

בפונקצייה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום
במחברת הבחינה את מספרי הביטויים החסרים (1) – (4), בסדר עולה, וכתוב ליד כל מספר
את הביטוי החסר שהוא מייצג.

```
int maxlen(int m)
{
    int min = 1, max = m, mid;
    if(m <= 57) return 0;
    while(max > min + 1)
    {
        mid = (min + max) / 2;
        hb = _____ (1) _____
        if(hb == m) _____ (2) _____;
        if(hb < m) _____ (3) _____;
        if(hb > m) _____ (4) _____;
    }
    return min;
}
```

פרק שני (15 נקודות)

ענה על אחת מבין השאלות 3-4 (לכל שאלה - 15 נקודות).

שאלה 3

לפניך פונקצייה שכותרתה:

```
void compressString(char *str, char *pair)
```

פונקצייה זו מקבלת מחרוזת str המכילה אותיות אנגליות קטנות בלבד (a – z), ודוחסת את המחרוזת באופן הזה:

הפונקצייה תמצא במחרוזת זוג אותיות **שונות**, המופיעות ברציפות מספר הפעמים הרב ביותר, ותחליף אותם בתו "!".

נוסף על כך, הפונקצייה תחזיר במשתנה pair את זוג האותיות הזה.

אם ישנם כמה זוגות שמופיעים מספר זהה (מקסימלי) של פעמים, אז הפונקצייה תדחוס רק **אחד** מן הזוגות.

לדוגמה:

במחרוזת "abababcd", הזוג "ab" מופיע שלוש פעמים ברציפות, הזוג "ba" מופיע פעמיים ברציפות, ואילו שאר הזוגות מופיעים פעם אחת בלבד. אי לכך, יש לדחוס את הזוג "ab", ובסיום הפונקצייה נקבל כי: pair = "ab", str = "!!cd".

במחרוזת "ababa", הזוג "ab" והזוג "ba" מופיעים מספר זהה (מקסימלי) של פעמים. אי לכך, יש לדחוס את אחד מהם, ובסיום הפונקצייה נקבל: pair = "ab", str = "!!a".
או: pair = "ba", str = "a!!".

הערות:

- הפונקצייה דוחסת זוג אותיות שונות **בלבד**, ולכן המחרוזת "aaaaaa", למשל, לא תידחס כלל.
- הזוג "ab" נחשב לשונה מן הזוג "ba".

הפונקצייה compressString נעזרת בפונקציות tonum, frequentPair ו־ compressPair המופיעות בהמשך.

להלן הפונקצייה tonum :

```
int tonum(char c)
{
    return c - 'a';
}
```

להלן הגדרות בשפת C :

```
#define LETTERS 26

typedef enum {FALSE,TRUE} boolean;
```

בפונקציות חסרים אחד-עשר ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (11), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
boolean frequentPair(char *str, char *pair)
{
    int hist[LETTERS][LETTERS] = {{0}};
    int max = 0, max_i = 0, max_j = 0;
    int i, j;
    while(str[0] != '\0' && str[1] != '\0')
    {
        if(str[0] != str[1])
        {
            _____(1)_____;
        }
        _____(2)_____;
    }
    for(i = 0; i < LETTERS; ++i)
        for(j = 0; j < LETTERS; ++j)
        {
            if(max < hist[i][j])
            {
                max = hist[i][j];
                max_i = i;
                max_j = j;
            }
        }
    if(_____(3)_____) return FALSE;
    pair[0] = _____(4)_____;
    pair[1] = _____(5)_____;
    pair[2] = _____(6)_____;
    return TRUE;
}
```



```
void compressPair(char *str, char *pair)
{
    char *out = str;
    while(str[0] != '\0' && str[1] != '\0')
    {
        if(str[0] == pair[0] && str[1] == pair[1])
        {
            _____(7)_____;
            str = _____(8)_____;
        }
        else
        {
            _____(9)_____;
            str++;
        }
        out++;
    }
    out[0] = _____(10)_____;
    _____(11)_____ out[1] = '\0';
}
```

```
void compressString(char *str, char *pair)
{
    if(frequentPair(str, pair))
        compressPair(str, pair);
}
```

שאלה 4

להלן הכרזות:

```
#define N 3  
  
typedef enum {FALSE,TRUE} boolean;
```

א. לפניך פונקצייה שכותרתה:

```
int factors(int num)
```

פונקצייה זו מקבלת את `num`, שהוא מספר שלם חיובי וגדול מ-0, ומחזירה את כמות הגורמים (המחלקים) השונים שיש לו.

דוגמה:

עבור `num = 5`, הפונקצייה תחזיר 2, מכיוון שלמספר 5 קיימים 2 מחלקים בלבד, שהם: 1 ו-5.
עבור `num = 4`, הפונקצייה תחזיר 3, מכיוון שלמספר 4 קיימים 3 מחלקים בלבד, שהם: 1, 2 ו-4.
עבור `num = 32`, הפונקצייה תחזיר 6, מכיוון שלמספר 32 קיימים 6 מחלקים בלבד, שהם: 1, 2, 4, 8, 16 ו-32.

בפונקצייה חסרים **שני** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (2), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int factors(int num)  
{  
    int cnt = 0;  
    int i;  
    for(i = 1; i*i < num; i++) if(num % i == 0) _____ (1) _____;  
    if(_____ (2) _____) cnt++;  
    return cnt;  
}
```



ג. להלן שלוש פונקציות:

```
boolean findKFactors(int arr[], int K)
```

```
{  
    int used[N] = {0};  
    return findKFactors_aux(arr, used, 0, K);  
}
```

```
boolean check(int arr[], int used[], int K)
```

```
{  
    int sum = 0, i;  
    for(i = 0; i < N; i++) sum += arr[i]*used[i];  
    return (factors(sum) == K);  
}
```

```
boolean findKFactors_aux(int arr[], int used[], int index, int K)
```

```
{  
    boolean res;  
    if(check(arr, used, K)) return TRUE;  
    if(index == N )return FALSE;  
    used[index] = TRUE;  
    res = findKFactors_aux(arr, used, index+1, K);  
    if(res) return TRUE;  
    used[index] = FALSE;  
    return findKFactors_aux(arr, sed, index+1, K);  
}
```

המשך השאלה בעמוד הבא.

נתון המערך arr הזה:

13	1	8
----	---	---

1. רשום במחברתך את הערך המוחזר על-ידי הזימון `findKFactors(arr, 2)`.
 2. רשום במחברתך את הערך המוחזר על-ידי הזימון `findKFactors(arr, 3)`.
 3. רשום במחברתך את הערך המוחזר על-ידי הזימון `findKFactors(arr, 5)`.
 4. להלן ארבעה היגדים שאחד מהם מתאר את הפעולה שמבצעת הפונקצייה `findKFactors`. רשום במחברתך את מספרו של ההיגד הנכון.
- **היגד 1:** הפונקצייה מחפשת במערך תת-קבוצה של איברים, כך שסכום כל האיברים בתת-הקבוצה הוא בדיוק K . אם קיימת תת-קבוצה כזו, אזי הפונקצייה מחזירה את הערך `TRUE`, אחרת – היא מחזירה את הערך `FALSE`.
 - **היגד 2:** הפונקצייה מחפשת במערך רצף איברים הנמצאים זה ליד זה, וסכום כל האיברים ברצף הוא מספר בעל K מחלקים. אם קיים רצף כזה, אזי הפונקצייה מחזירה את הערך `TRUE`, אחרת – היא מחזירה את הערך `FALSE`.
 - **היגד 3:** הפונקצייה מחפשת במערך תת-קבוצה של איברים, כך שסכום כל האיברים בתת-הקבוצה הוא מספר בעל K מחלקים. אם קיימת תת-קבוצה כזו, אזי הפונקצייה מחזירה את הערך `TRUE`, אחרת – היא מחזירה את הערך `FALSE`.
 - **היגד 4:** הפונקצייה מחפשת במערך תת-קבוצה של K איברים, כך שסכומם הוא מספר בעל K מחלקים. אם קיימים K מספרים כאלה, אזי הפונקצייה מחזירה את הערך `TRUE`, אחרת – היא מחזירה את הערך `FALSE`.

חלק ב': שפת סף (50 נקודות)

פרק שלישי (20 נקודות)

ענה על שאלה 5 – שאלת חובה.

שאלה 5

לפניך תוכנית בשפת אסמבלי, הכוללת שגרה **רקורסיבית** בשם REC. השגרה מקבלת באמצעות מחסנית את הארגומנט NUM, שהוא מספר עשרוני שלם חיובי הגדול מ-0. אם ספרותיו של המספר (החל מספרת האחדות) מהוות סדרה עולה, שגרה זו מחזירה באוגר CL את הערך 1, אחרת – השגרה מחזירה באוגר CL את הערך 0.

לדוגמה: בעבור המספר 85541 השגרה תחזיר 1, ובעבור המספר 8154 השגרה תחזיר 0. הנחת יסוד: ערכו של המספר שמשוכן במשתנה NUM הוא 65535 לכל היותר.

א. בשגרה חסרים **שישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (6), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
SSEG SEGMENT STACK 'STACK'

DB 100 DUP()

SSEG ENDS

DSEG SEGMENT

NUM DW 54321

ZERO DW 0

DSEG ENDS

CODE SEGMENT

    ASSUME CS:CODE, DS:DSEG
```

המשך השאלה בעמוד הבא.

תכנות מערכות בשפת C ושפת סף,
אביב תשע"ח, סמל 714001

```
START:  MOV     AX, DSEG
        MOV     DS, AX
        PUSH    NUM
        PUSH    ZERO
        CLC
        CALL    REC
        ADD     SP, 2      ; (**)
        MOV     AH, 4CH
        INT     21H
REC:     PUSH    BP
        MOV     BP, SP
        JC      NEXT
        MOV     AX, _____ (1) _____
        MOV     BX, 10
        XOR     CL, CL
NEXT:    XOR     DX, DX
        _____ (2) _____
        CMP     DL, _____ (3) _____
        _____ (4) _____
        OR      AX, AX
        JNE     NEX
        _____ (5) _____
        JMP     EXIT
NEX:     _____ (6) _____
        STC
        CALL    REC
EXIT:    POP     BP
        RET     2
CODE ENDS
END START
```

ב. אם נשמיט מן התוכנית הנתונה את השורה המסומנת ב-**(**)**, האם שינוי זה ישפיע על ביצועי התוכנית? ענה "כן" או "לא".

ג. אם נחליף בתוכנית הנתונה את השורה **OR AX, AX** בשורה **CMP AX, 0**, האם שינוי זה ישפיע על הערך שתחזיר השגרה? ענה "כן" או "לא".

פרק רביעי (30 נקודות)

ענה על שתיים מבין השאלות 6-8 (לכל שאלה – 15 נקודות).

שאלה 6

לפניך תוכנית בשפת אסמבלי, הכוללת שגרה בשם SUM. השגרה מקבלת באמצעות מחסנית את הארגומנט NUM, שהוא מספר עשרוני שלם חיובי הגדול מ-0.

התוכנית מציבה במשתנה RESULT את סכום ספרותיו של המספר NUM (החל מספרת האחדות) בסימנים מתחלפים.

דוגמאות:

- בעבור המספר 187, התוכנית תציב במשתנה RESULT את הערך 0, מכיוון ש- $7 - 8 + 1 = 0$.
- בעבור המספר 295625, התוכנית תציב במשתנה RESULT את הערך 11, מכיוון ש- $5 - 2 + 6 - 5 + 9 - 2 = 11$.
- בעבור המספר 71919, התוכנית תציב במשתנה RESULT את הערך 23, מכיוון ש- $9 - 1 + 9 - 1 + 7 = 23$.

בתוכנית חסרים **שישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (6), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
SSEG SEGMENT STACK 'STACK'
DB 100 DUP()
SSEG ENDS
DSEG SEGMENT
NUM          DW  54321
RESULT       DW  ?
DSEG ENDS
CODE SEGMENT
                ASSUME CS:CODE,DS:DSEG
```

המשך השאלה בעמוד הבא.

```

START:  MOV     AX, DSEG
        MOV     DS, AX
        PUSH    NUM
        CALL    SUM
        _____ (1) _____
        MOV     AH, 4CH
        INT     21H
SUM:     PUSH    BP
        MOV     BP, SP
        MOV     AX, [BP+4]
        MOV     WORD PTR [BP+4], 0
        MOV     BX, 10
        XOR     DX, DX
AG:      OR      AX, AX
        _____ (2) _____
        _____ (3) _____
        ADD     [BP+4], _____ (6) _____
        OR      AX, AX
        _____ (2) _____
        _____ (4) _____
        CWD
        _____ (5) _____
        ADD     [BP+4], _____ (6) _____
        NEG     AX
        CWD
        JMP     AG
EXIT:    POP     BP
        RET

CODE ENDS

END START

```


שאלה 7

להלן סגמנט נתונים:

```
DATA SEGMENT
STR1  DB  'GOOD'
NUM1  DB  15
NUM2  DB  '15'
NUMS  DW  100H, 100
ADDR  DW  NUM1
ADDR1 DD  NUM2
DATA  ENDS
```

נתון כי האוגר DS מצביע על סגמנט הנתונים הזה, ותוכנו הוא: IFBCH .

- א. מהו הגודל של סגמנט הנתונים בבתים (bytes) ? רשום את תשובתך בבסיס עשרוני.
- ב. האם תוכנו של הבית (byte) שכתובתו NUM1 זהה לתוכנו של הבית (byte) שכתובתו NUM2 ? ענה "כן" או "לא".
- ג. מה תוכנו של הבית (בבסיס 10) שכתובתו ADDR1 ?
- ד. מה תוכנו של הבית (בבסיס 16) שכתובתו ADDR1+1 ?
- ה. מה תוכנו של הבית (בבסיס 16) שכתובתו ADDR1+2 ?
- ו. מה תוכנו של הבית (בבסיס 16) שכתובתו ADDR1+3 ?

שאלה 8

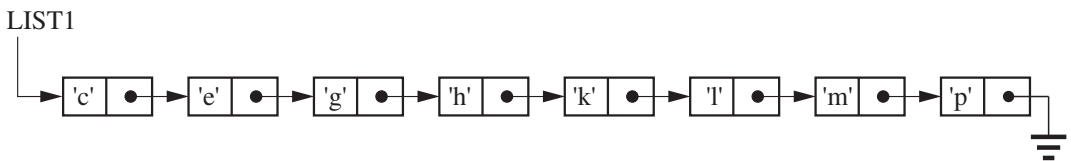
נתונות שתי רשימות מקושרות חד-כיווניות לא ריקות הבנויות מצמתים, ושתיהן ממוינות בסדר עולה.

כל צומת ברשימות הללו מכיל את שני השדות האלה:

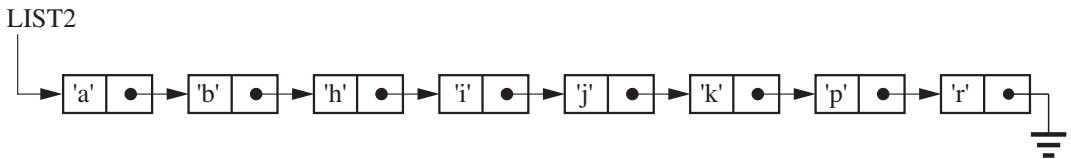
info – שדה מידע (אינפורמציה), שגודלו 8 סיביות, המכיל תו (אות קטנה מהאלף-בית האנגלי).

next – מצביע על הצומת הבא ברשימה, שגודלו מילה אחת (16 סיביות). המצביע next בצומת האחרון הוא 0.

נוסף על כך, נתון כי המשתנה LIST1 הוא מצביע לצומת הראשון ברשימה הראשונה, כלומר LIST1 מכיל את הכתובת של הצומת הראשון ברשימה הראשונה (ראה איור א'), וכן נתון כי המשתנה LIST2 הוא מצביע לצומת הראשון ברשימה השנייה, כלומר LIST2 מכיל את הכתובת של הצומת הראשון ברשימה השנייה (ראה איור ב').



איור א' לשאלה 8



איור ב' לשאלה 8

לפניך קטע תוכנית בשפת אסמבלי, אשר סורק את שתי הרשימות הנתונות (פעם אחת בלבד), מאתר את התווים המופיעים גם ברשימה הראשונה וגם ברשימה השנייה, ומכניס את התווים האלו למחסנית.

לדוגמה: עבור הרשימות שבאיור א' ובאיור ב', התווים שיוכנסו למחסנית הם: 'h', 'k', 'p'.

בשגרה חסרים **שמונה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (8), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```

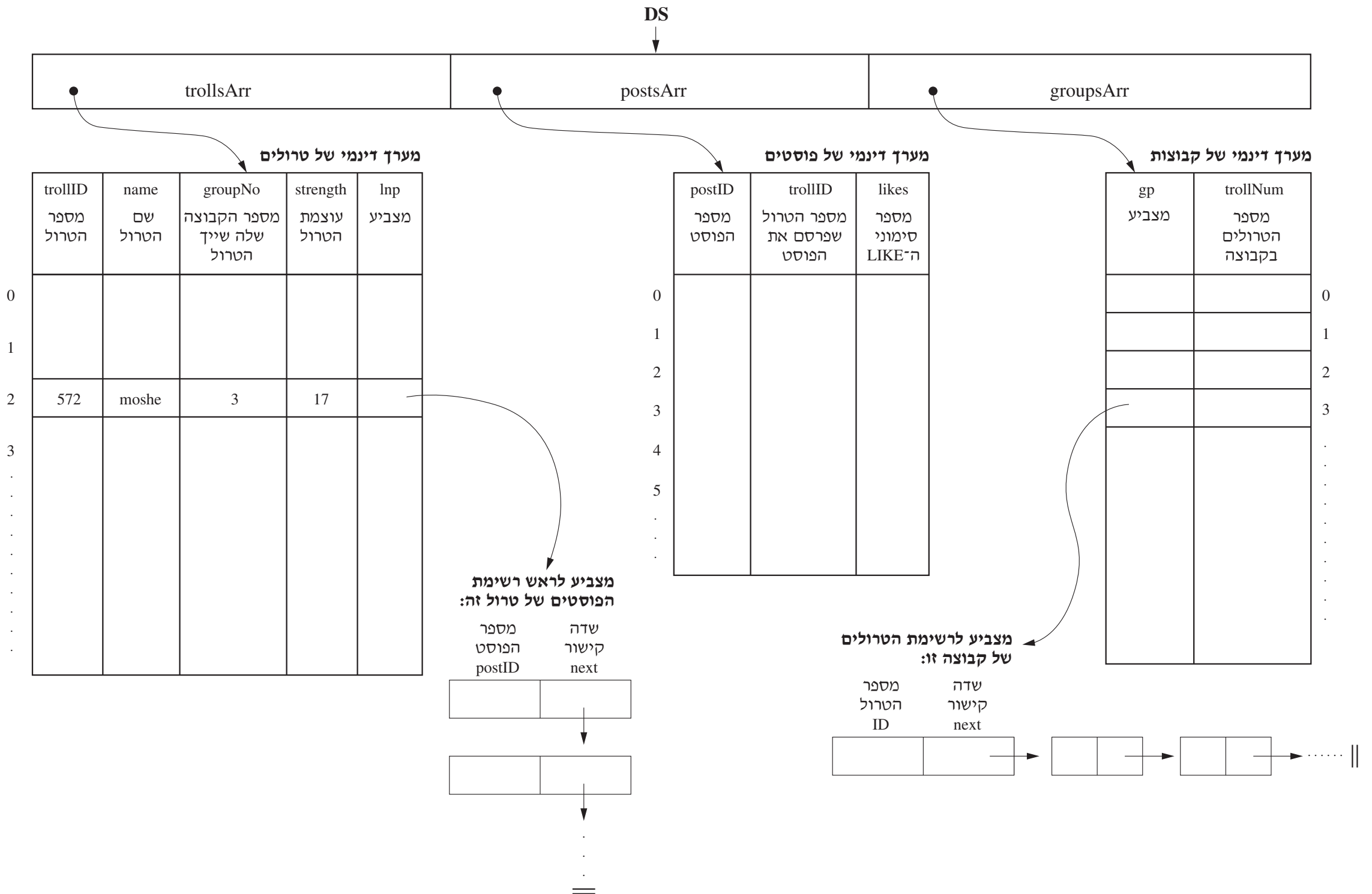
START:  MOV    AX, DATA
        MOV    DS, AX
        LEA    DI, LIST1
        LEA    SI, LIST2
        XOR    AX, AX
AGAIN:  MOV    AL, [DI]
        CMP    AL, [SI]
        JNE    NOT_EQU
        _____ (1) _____
        CMP    WORD PTR [DI+1], 0
        JE     EXIT
        _____ (2) _____
        CMP    WORD PTR [SI+1], 0
        JE     EXIT
        _____ (3) _____
        _____ (4) _____
NOT_EQU: CMP    AL, [SI]
        JNB    NEXT
        _____ (5) _____
        JE     EXIT
        MOV    _____ (6) _____
        JMP    _____ (7) _____
NEXT:   CMP    WORD PTR [SI+1], 0
        JE     EXIT
        MOV    _____ (8) _____
        JMP    _____ (7) _____
EXIT:   MOV    AH, 4CH
        INT    21H

```

בהצלחה!

נספח א' – לשאלה 1: תיאור סכמתי של "המבנה הראשי" עבור המערכת הממוחשבת

לשאלון 714001, אביב תשע"ח



תרגום המונח			המונח
אנגלית	רוסית	ערבית	
חלק א' – תכנות מערכות בשפת C :			
initialization	Инициализация	ابتداء	אתחול
composition	Включение в себя, содержание в себе	ضمّ	הכלה
memory allocation	выделение памяти	تخصيص ذاكرة	הקצאת זיכרון
flag	Обозначение конца, конечный элемент	عَلَم	זקיף
one-direction	однонаправленный	ذات اتّجاه واحد	חד-כיווני
type	Тип	نوع	טיפוס
data structure	структура данных	بُنية البيانات	מבנה נתונים
dynamic array	Динамический массив	مصفوفة غير ثابتة	מערך דינמי
pointer	Указатель	مُؤشّر	מצביע
global variable	Глобальная переменная	متغيّر عامّ	משתנה גלובאלי
bits series	Последовательность битов	سلسلة أرقام ثنائية / سلسلة بتات	סדרת סיביות
parameter	Параметр	متغيّر (بارامتر)	פרמטר
node	Узел, вершина	مَفْرَق	צומת
fixed	Константа	ثابت	קבוע
binary file	Двоичный файл	مِلَفّ ثنائيّ	קובץ בינארי
linked list	связный список	قائمة مرتبطة	רשימה מקושרת
field	поле	حَقْل	שדה
concatenation	Конкатенация	تَرَاوُط	שרשור
cell	Ячейка	خلية	תא

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
חלק ב' – שפת סף:			
register	Регистр	מخزن (ريغستر)	אוגר
main diagonal	Главная диагональ	مائل رئيسي	אלכסון ראשי
hexadecimal base	Шестнадцатеричная система счисления, основание 16	قاعدة الست عشرية	בסיס הקסאדצימלי
stack	Стек	باغة	מחסנית
decimal base number	десятичное число	عدد عشري	מספר בבסיס עשרוני
marked number	обозначенное число, число со знаком	عدد مُعَلَّم	מספר מסומן
bits	биты	أرقام ثنائية (بتات)	סיביות
palindrome	Палиндром	سِيَّاقٌ مُتَنَاطِر (بولينדרوم)	פלינדרום
compilation	компиляция	ترجمة الأوامر	קומפילציה (הידור)
routine	Функция, рутина	رتابة / روتين	שגרה
recursive routine	Рекурсивная функция	إجراء تراجعي	שגרה רקורסיבית