

תכנות מערכות בשפת C ושפת סף

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

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

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

בשאלון זה 40 עמודים.

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

בהצלחה!

השאלות

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

פרק ראשון (35 נקודות)

ענה על שתי השאלות 1–2 .

שאלה 1 – שאלת חובה (25 נקודות)

מעוניינים לבנות מערכת ממוחשבת שתנהל מידע על שחקנים וקבוצות כדורגל בליגה הלאומית. כדי לנהל את המידע הוחלט להחזיק שתי רשימות מקושרות חד-כיוונית: אחת בעבור קבוצות כדורגל, והשנייה בעבור שחקני כדורגל בליגה הלאומית.

הנחות יסוד:

1. שחקן יכול להיות שייך לקבוצה אחת לכל היותר.
2. שחקן שלא שייך לאף קבוצה ייקרא "שחקן חופשי".
3. קבוצה ללא שחקנים תיקרא "קבוצה לא פעילה".
4. בניהול המידע של המערכת הממוחשבת קבוצה עשויה לעבור למצב של "קבוצה לא פעילה".

להלן הגדרות והכרזות של טיפוסים הנתונים **במערכת הממוחשבת** בשפת C התקפות בכל סעיפי השאלה:

```
typedef enum {FAILURE, SUCCESS, INVALID_INPUT,
DUPLICATE_RECORD, MISSING_RECORD}    statusType;
```

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

```
typedef struct playerType    // טיפוס שחקן
```

```
{
    long plyrID;                // מספר הזהות
    int age;                    // גיל
    char firstName[15];        // שם פרטי
    char lastName[20];         // שם משפחה
    struct teamType *tmPtr;     // מצביע לצומת ברשימת הקבוצות, המייצג את הקבוצה שבה
                                // משחק השחקן
    struct playerType *next;    // שדה קישור, מצביע לצומת הבא המייצג שחקן

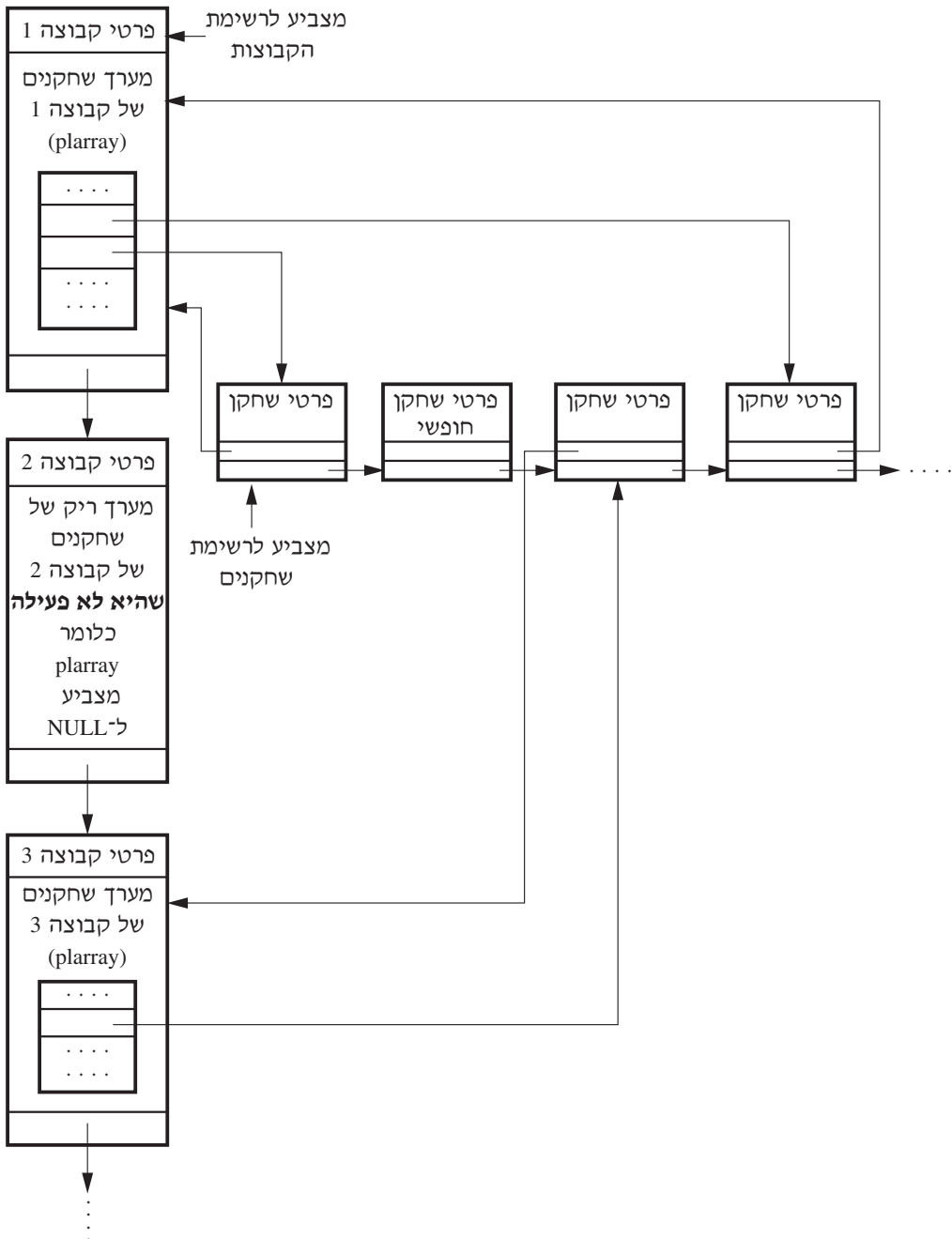
} playerRec , *playerPtr;
```

```
typedef struct teamType    // טיפוס הקבוצה
```

```
{
    char teamName[20];         // שם הקבוצה
    int num;                   // מספר השחקנים בקבוצה
    playerPtr *plarray;        // מערך דינאמי של שחקני קבוצה. כל תא במערך זה מצביע
                                // לצומת המייצג שחקן ברשימה מקושרת חד-כיוונית של השחקנים
    struct teamType *next;     // שדה קישור, מצביע לצומת הבא המייצג קבוצה

} teamRec, *teamPtr;
```

להלן איור של מבנה הנתונים לניהול מידע:



נתונה ספריית פונקציות. הספרייה מכילה, בין היתר, את הפונקציות האלה:

שם הפעולה	תיעוד
statusType init (void)	הפונקציה מאתחלת את המערכת הממוחשבת.
הוספת שחקן statusType insertPlayer (long playerId,char lastName[], char firstName[],int age)	הפונקציה מקבלת את פרטי השחקן. אם השחקן אינו מופיע במערכת הממוחשבת, אז הפונקציה מוסיפה אותו לראש הרשימה המקושרת של שחקנים ומחזירה את הערך SUCCESS ; ואם הוא כן מופיע – אזי היא מחזירה את הערך DUPLICATE_RECORD .
מחיקת שחקן statusType deletePlayer (long playerId)	הפונקציה מקבלת את מספר הזהות של שחקן. אם השחקן מופיע ברשימת השחקנים, אז הפונקציה מוחקת אותו מן המערכת הממוחשבת ומחזירה את הערך SUCCESS ; אם הוא אינו מופיע, אזי היא מחזירה את הערך MISSING_RECORD .
הוספת קבוצה statusType insertTeam (char name[])	הפונקציה מקבלת את שם הקבוצה. אם הקבוצה אינה מופיעה ברשימת הקבוצות, אז היא מוסיפה אותה לראש רשימת הקבוצות ומחזירה את הערך SUCCESS ; אם היא מופיעה, אזי היא מחזירה את הערך DUPLICATE_RECORD .
מחיקת קבוצה statusType deleteTeam (char teamName[])	הפונקציה מקבלת את שם הקבוצה. אם הקבוצה מופיעה ברשימת הקבוצות, אז היא הופכת את כל שחקניה ל"שחקנים חופשיים", מוחקת את הקבוצה מרשימת הקבוצות ומחזירה את הערך SUCCESS ; ואם היא אינה מופיעה, היא מחזירה את הערך MISSING_RECORD .
הוספת שחקן לקבוצה statusType joinPlayerToTeam (long PlayerID,char team[])	הפונקציה מקבלת את מספר הזהות של שחקן ואת שם הקבוצה. אם השחקן מופיע ברשימת השחקנים והנו שחקן חופשי והקבוצה team מופיעה ברשימת הקבוצות, אז הפונקציה מוסיפה את השחקן הזה לקבוצה ומחזירה את הערך SUCCESS ; אם לא – היא מחזירה את הערך MISSING_RECORD .

<p>מחיקת שחקן מקבוצה</p> <p>statusType deletePlayerFromTeam (long playerID)</p>	<p>הפונקציה מקבלת את מספר הזהות של שחקן. אם השחקן מופיע ברשימת השחקנים ושייך לאחת הקבוצות, אז היא מוחקת את השחקן מהקבוצה בה הוא משחק, כלומר הופכת אותו ל-"שחקן חופשי", ומחזירה את הערך SUCCESS, ואם הוא אינו מופיע, אזי היא מחזירה את הערך MISSING_RECORD.</p>
<p>הדפסת שחקנים בליגה</p> <p>void printPlayers (void)</p>	<p>הפונקציה מדפיסה את פרטי כל שחקני הליגה הלאומית.</p>
<p>הדפסת קבוצות בליגה</p> <p>void printTeams (void)</p>	<p>הפונקציה מדפיסה את פרטי כל הקבוצות של הליגה הלאומית.</p>
<p>הדפסת פרטי קבוצה</p> <p>statusType printTeamDetails (char team[])</p>	<p>הפונקציה מקבלת את שם הקבוצה. אם הקבוצה נמצאת ברשימת הקבוצות, אז היא מדפיסה את פרטי הקבוצה ואת פרטי כל שחקני הקבוצה ומחזירה את הערך SUCCESS, ואם הקבוצה אינה נמצאת ברשימה – היא מחזירה את הערך MISSING_RECORD.</p>
<p>teamPtr mySort (teamPtr head,int (*compare) (teamPtr p,teamPtr q))</p>	<p>הפונקציה מקבלת את head שהינו מצביע לראש רשימת קבוצות ואת פונקציות ההשוואה compare. הפונקציה ממיינת בסדר עולה את רשימת הקבוצות.</p>
<p>איתור שחקן</p> <p>playerPtr findPlayer (long playerID)</p>	<p>הפונקציה מקבלת את מספר הזהות של שחקן. אם השחקן מופיע ברשימת השחקנים, אז היא מחזירה מצביע לצומת המייצג את השחקן הזה; אם הוא אינו מופיע, אזי היא מחזירה את הערך NULL.</p>
<p>איתור קבוצה</p> <p>teamPtr findTeam (char name[])</p>	<p>הפונקציה מקבלת את שם הקבוצה. אם הקבוצה מופיעה ברשימת הקבוצות, אז היא מחזירה מצביע לצומת המייצג את הקבוצה הזו; אם הוא אינו מופיע, אזי היא מחזירה את הערך NULL.</p>

בכל סעיף הנך רשאי להשתמש בכל הפונקציות שבטבלה שלעיל.

להלן הגדרות בעבור משתנים גלובאליים:

```
playerPtr players; // מצביע לראש רשימה מקושרת חד-כיוונית של שחקנים בליגה הלאומית  
teamPtr teams;    // מצביע לראש רשימה מקושרת חד-כיוונית של קבוצות בליגה הלאומית
```

א. לפניך פונקציה שכוורתה:

```
statusType insertPlayer(long playerID, char lastName[],  
char firstName[], int age)
```

פונקציה זו מקבלת את פרטי השחקן: מספר הזהות של – playerID, שם המשפחה – lastName, השם הפרטי – firstName, והגיל – age. הפונקציה מוסיפה את השחקן הזה לראש רשימה מקושרת של השחקנים.

הערה: פונקציה זו אינה מצרפת את השחקן לקבוצה כלשהי, כלומר השחקן יהיה "שחקן חופשי".

הפונקציה מחזירה את הערך מטיפוס statusType, כמפורט בטבלה שלהלן:

אם שחקן בעל מספר זהות playerID קיים ברשימת השחקנים.	DUPLICATE_RECORD
אם שחקן בעל מספר זהות playerID נוסף לרשימת השחקנים.	SUCCESS

בפונקציה 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 insertPlayer(long playerID,char lastName[],
char firstName[],int age)
{
    statusType status =____(1)____;
    playerPtr p,q;
    q =____(2)____;
    if(q)
    {
        status = DUPLICATE_RECORD;
        return status;
    }
    p = malloc(sizeof(playerRec));
    p -> plyrID = playerID;
    strcpy(p->firstName,firstName);
    strcpy(p->lastName,lastName);
    p -> age = age;
    p -> tmptr =____(3)____;    //"שחקן חופשי"
    ____ (4) ____ = players;
    ____ (5) ____ = p;
    return status;
}

```


ב. לפניך פונקציה שכותרתה:

```
statusType deletePlayerFromTeam(long playerID)
```

פונקציה זו מקבלת את מספר הזהות של שחקן – playerID .
אם השחקן מופיע ברשימת השחקנים, אז היא מוחקת את השחקן מהקבוצה שבה הוא משחק, כלומר הופכת אותו ל-"שחקן חופשי", ומחזירה את הערך SUCCESS . אם לא – היא מחזירה את הערך MISSING_RECORD .

הערה: הפונקציה הזו **אינה** מוחקת את השחקן מרשימת השחקנים.

הפונקציה מחזירה את הערך מטיפוס statusType , כמפורט בטבלה שלהלן:

MISSING_RECORD	אם שחקן בעל מספר זהות playerID לא קיים ברשימת השחקנים או אם השחקן הוא "שחקן חופשי".
SUCCESS	אם שחקן בעל מספר זהות playerID אינו "שחקן חופשי".

להלן תהליך למחיקת שחקן מהקבוצה:

מאתרים את השחקן במערך שחקני קבוצתו ומציבים את מיקומו במערך הזה במשתנה k .
ערך שנמצא בתא שמיקומו i+1 יועתק לתא i לכל $k+1 \leq i < \text{num}$, כאשר num מציין את מספר השחקנים בקבוצה זו.

מעדכנים את מספר השחקנים בקבוצה זו.

מקצים מערך דינמי חדש של שחקני קבוצה זו כאשר גודל המערך החדש הוא כמספר השחקנים בקבוצה בשלב זה.

בפונקציה 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 deletePlayerFromTeam(long playerID)
{
    playerPtr p;
    teamPtr t;
    int i,k;
    statusType status = SUCCESS;
    p = ____ (1) ____;
    if ((p==NULL) || (p->tmptr==NULL)) status= MISSING_RECORD;
    else
    {
        t = p->tmptr;
        p->tmptr = NULL;
        for(i=0;i<____ (2) ____; i++)
        {
            if (____ (3) ____== playerID)
            {
                k = i;  break;
            }
        }
        for(i=k+1;i<____ (2) ____; i++)
        {
            t -> plarray[i-1] = t->plarray[i];
        }
        t->num = t->num - 1;
        t->plarray=realloc(t->plarray,____ (4) ____);
    }
    return status;
}

```

ג. לפניך פונקציה שכותרתה:

```
statusType deletePlayer(long playerID)
```

פונקציה זו מקבלת את מספר הזהות של שחקן playerID .

אם השחקן מופיע ברשימת השחקנים (כשחקן חופשי או כשחקן שאינו חופשי), אז הפונקציה מוחקת אותו מן המערכת הממוחשבת.

הפונקציה הזו מחזירה את הערך מטיפוס statusType כמפורט בטבלה שלהלן:

MISSING_RECORD	אם שחקן בעל מספר זהות playerID אינו קיים ברשימת שחקנים.
SUCCESS	אם שחקן בעל מספר זהות playerID קיים ברשימת השחקנים.

בפונקציה 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 deletePlayer(long playerID)
{
    playerPtr p,q;
    statusType status= ____ (1) ____ ;
    p = findPlayer(playerID);
    if(p==NULL) status = MISSING_RECORD;
    else
    {
        if (p->tmptr) status = ____ (2) ____;
        q = players;
        if ( ____ (3) ____ )
        {
            players = q->next;
            free (p);
        }
        else
        {
            while (q->next)
            {
                if (q->next == p)
                {
                    q->next = ____ (4) ____;
                    ____ (5) ____;
                    break;
                }
                q = q->next;
            }
        }
    }
    return status;
}

```

ד. לפניך פונקציה שכותרתה:

```
statusType deleteTeam(char teamName[])
```

פונקציה זו מקבלת את שם הקבוצה - teamName. אם הקבוצה נמצאת ברשימת הקבוצות, אז הפונקציה הופכת את כל שחקניה ל"שחקנים חופשיים" (ראה הנחת יסוד 2), ומוחקת את הקבוצה הזו מרשימת הקבוצות.

הפונקציה מחזירה את הערך מטיפוס statusType כמפורט בטבלה שלהלן:

אם teamName לא מופיע ברשימת הקבוצות.	MISSING_RECORD
אם teamName מופיע ברשימת הקבוצות.	SUCCESS

בפונקציה 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 deleteTeam(char teamName[])
{
    statusType status = SUCCESS;
    int i;
    teamPtr t,s;
    t = _____(1)_____ ;
    if (t==NULL) status = MISSING_RECORD;
    else
    {
        if(_____(2)_____)
        {
            for(i=0;i < t->num;i++)
            {
                _____(3)_____ = NULL;
            }
        }
        if (t == teams)      //ראש רשימה
        {
            teams = t->next;
        }
        else
        {
            s = teams;
            while(s->next != t)
            {
                s = s->next;
            }
            _____(4)_____ ;
        }
        _____(5)_____ ;
    }
    return status;
}

```

ה. לפניך פונקציה שכותרתה:

```
statusType joinPlayerToTeam(long playerID, char team[])
```

הפונקציה מקבלת את מספר הזהות של שחקן - playerID, ואת שם הקבוצה team .

אם השחקן מופיע ברשימת השחקנים והנו שחקן חופשי והקבוצה team מופיעה ברשימת הקבוצות, אזי הפונקציה מוסיפה את השחקן הזה לקבוצה ומחזירה את הערך SUCCESS .
אם לא - היא מחזירה את הערך MISSING_RECORD.

הפונקציה מחזירה את הערך מטיפוס statusType , כמפורט בטבלה שלהלן:

SUCCESS	במקרה של הצלחה.
MISSING_RECORD	בכל מקרה אחר.

בפונקציה חסרים **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 joinPlayerToTeam(long playerID, char team[])
{
    statusType status = SUCCESS;
    playerPtr p;
    teamPtr t;
    p = findPlayer(playerID);
    t = findTeam(team);
    if((p==NULL) || (t ==NULL)) status = MISSING_RECORD;
    else
        if (____(1)____) status = MISSING_RECORD;
        else
        {
            p->tmptr = ____ (2) ____;
            if(t->num == 0)
            {
                (t->plarray) = realloc(____(3)____, sizeof(playerPtr));
                ____ (4) ____ = p;
                t->num = 1;
            }
            else
            {
                t -> plarray = realloc
                    (____(5)____, (____ (6) ____) *sizeof(playerPtr));
                ____ (7) ____ = p;
                t -> num = t->num + 1;;
            }
        }
    return status;
}

```


1. מטרתנו היא לכתוב פונקציה רקורסיבית בשם:

```
teamPtr mySort (teamPtr head, int (*compare) (teamPtr p, teamPtr q))
```

הפונקציה מקבלת את head, שהנו מצביע לראש רשימת הקבוצות, ואת פונקציות ההשוואה compare. תפקידה של הפונקציה הוא למיין בסדר עולה את רשימת הקבוצות באופנים שונים. המיין יתבצע פעם לפי שם הקבוצה, פעם לפי מספר השחקנים בקבוצה ופעם לפי מפתח אחר. ניתן לזמן את הפונקציה הזו באופנים שונים, כדלהלן:

אפשרות 1: מיין קבוצות לפי שמות (מחרוזות). הזימון לפונקציה יראה כך:

```
mySort (teams, compareNames)
```

אפשרות 2: מיין קבוצות לפי מספר השחקנים בקבוצה (מספרים). הזימון לפונקציה ייראה כך:

```
mySort (teams, compareSizes)
```

להלן מימוש של פונקציות ההשוואה שבהן תשתמש הפונקציה **mySort**.

```
int compareNames (teamPtr p, teamPtr q)
{
    return strcmp(p -> teamName, q->teamName);
}

int compareSizes (teamPtr p, teamPtr q)
{
    if (p -> num == q -> num) return 0;
    else
        if (p -> num < q -> num) return -1;
        else return 1;
}
```

לפניך מימוש הפונקציה **הרקורסיבית mySort** :

בפונקציה 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.

```
teamPtr mySort(teamPtr head,int (*compare)(teamPtr p,teamPtr q))
{
    teamPtr t,s;
    t=head ;
    if ((t==NULL) || (t->next == NULL)) return head ;
    t = _____(1)_____;
    if (compare(head,t) < 0)
        _____(2)_____;
    else
    {
        s = head;
        head = t;
        while(t->next)
        {
            if(compare(s,t->next) < 0)
            {
                s->next = _____(3)_____;
                _____(3)_____ = s; break;
            }
            t = t->next;
        }
        if (t->next == NULL)           //מקרה: הכנסה לסוף הרשימה
        {
            t->next = s;
            _____(4)_____;
        }
    }
    return head;
}
```

שאלה 2 - שאלת חובה (10 נקודות).

להלן תכנית בשפת C. הנח כי הטיפוס int מיוצג באמצעות ארבעה בתים, הטיפוס char מיוצג באמצעות בית אחד, וכתובת של משתנה מיוצגת באמצעות ארבעה בתים.

```
#include <stdio.h>
void f1(struct St1 st1);
struct St1
{
    int ptrInt;
    char *ptrChar;
    struct St1 *pSt1;
};
int arrInt[10] = {1,2,3,4,5,6,7,8,9,10};
char *arrChar[5] = {"hh","jj","kk","zz","xx"};

// המשך התכנית בעמוד הבא
```

```

int main()
{
    struct St1 st1Arr[3] = {
        { *arrInt, arrChar[3], st1Arr+1 },
        { (arrInt[2]), *(arrChar+2), st1Arr },
        { *&arrInt[7], *arrChar, st1Arr+2 }
    };

    struct St1 *pst1Arr[3] = { st1Arr, st1Arr+1, st1Arr+2 };
    struct St1 **ppst1Arr = pst1Arr;
    printf("size=%d\n", sizeof(arrChar));
    printf("size=%d\n", sizeof(&st1Arr));
    printf("size=%d\n", sizeof(st1Arr));
    printf("size=%d\n", sizeof(ppst1Arr));
    printf("*****\n");
    printf("%c \n", ((*ppst1Arr).ptrChar));
    printf("%d \n", (pst1Arr[2])->ptrInt);
    printf("%s \n", ((+(*ppst1Arr))->ptrChar));
    printf("%d \n", ((*ppst1Arr)+++ 1)->pSt1->ptrInt);
    ppst1Arr = pst1Arr;
    f1(*ppst1Arr);
    printf("%d \n", (*ppst1Arr)->ptrInt);
    f1(*ppst1Arr);
    getchar();
    return 0;
}

void f1(struct St1 st1)
{
    static int num=5;
    st1.ptrInt = ++num;
    printf("num=%d \n", num);
}

```

רשום במחברתך את הפלט המדויק של התכנית .

פרק שני (15 נקודות)

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

שאלה 3

לפניך שלושה סעיפים שאינם תלויים זה בזה. ענה על כולם.

א. לפניך פונקציה שכותרתה: `char* func(unsigned n)`.

הפונקציה מחזירה מערך דינמי של n תווים בסדר הפוך, כך שניתן יהיה להדפיס אותו במחרוזת. גודל המערך יהיה רק לפי הנדרש בדיוק. (אם המספר המקורי מסתיים באפסים, אז המחרוזת תכיל גם אותם בהתחלה.) הנח שהמספר n גדול מ-0.
דוגמה לשימוש בפונקציה:

```
void main()
{
    char *p = func(123456);
    printf("%s\n", p);           // (654321)
    free(p);
}
```

בפונקציה 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. הכנסת הספרות של המספר מימין לשמאל, כלומר החל בספרת האחדות שבצד ימין אל המערך הדינמי משמאל לימין, כמקובל, הופכת את תווי המספר.

```
char* func(unsigned n)
{
    unsigned size=1;
    char *str = NULL;
    while(n)
    {
        str=____(1)____;
        if(!str){ printf ("\nNot enough memory!"); exit(1); }
        str[____(2)____]=____(3)____;    // הכנסת הספרה לפי ערך האסקי שלה
        n/=10;
    }
    ____ (4) ____;
    return ____ (5) ____;
}
```

ב. לפניך תכנית בשפת C.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int strfunc(char *s,char *t)
{
    int i;
    int result = 0;
    int s_length;
    int t_length;
    char temp[30];
    char *temp_p;
    s_length = strlen(s);
    t_length = strlen(t);
    if(t_length <= s_length)
        for(i=0;i<(s_length - t_length+1)&& !result;i++)
        {
            strcpy(temp,s);
            temp_p =temp+i;
            temp[t_length+i] = \0;
            if(strcmp(temp_p,t)==0) result = 1;
        }
    return result;
}
int main(void)
{
    char s1[30] = "question number e in c exam";
    char s2[30] = "n n";
    char s3[30] = "exam";
    char s4[30]="inc";
    if(strfunc(s1,s2)) printf("Yes\n"); else printf ("No\n");
    if(strfunc(s1,s3)) printf("Yes\n"); else printf ("No\n");
    if(strfunc(s1,s4)) printf("Yes\n"); else printf ("No\n");
    return 0;
}
```

רשום במחברתך את הפלט המדויק של התכנית הנתונה.

ג. נתונים המבנים הבאים:

```
typedef struct
{
    char name[LEN];    // שם הנהג
    int id;             // מספר הרישיון של הנהג
}Driver;              //נהג

typedef struct
{
    int price;          // מחיר המכונית
    Driver d;
}Car;                 // מכונית
```

נתונים ב־main שני מערכים של מכוניות.
נתון שכל אחד מהמערכים ממין בסדר עולה לפי מספר הרישיון של הנהג.

לפניך פונקציה: `func(const Car A[], int n1, const Car B[], int n2)`, `Car **`,
כאשר `n1` – גודל המערך `A`. `n2` – גודל המערך `B`. הפונקציה יוצרת מערך דינמי חדש של
מצביעים למכוניות, כך שכל מצביע יצביע למבנה חדש של מכונית אחרת.

במערך החדש יופיעו כל המכוניות משני המערכים, והוא יהיה ממין בסדר עולה לפי מספרי
הרישיון של נהגי המכוניות.

אילוצים:

אין להשתמש במערך־עזר. אין לשנות את המערכים המקוריים.
אסור להשתמש בלולאות מקוננות, כלומר לא תופיע לולאה בתוך לולאה.
(לא יתקבלו פתרונות עם לולאות מקוננות.)

לאור האילוצים האלו להלן הפונקציה:

בפונקציה זו חסרים **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.

```

Car **func(const Car A[],int n1,const Car B[],int n2)
{
    int i=0,j=0,k=0,t;

    Car **C=____(1)____ ;           //מקצה את המצביעים
    if(!C) {printf("Not enough memory!");exit(1);}
    for(t=0;t<n1+n2;t++)              //מקצה את המבנים החדשים
    {
        C[t]= ____ (2) ____ ;
        if(!C[t]){ printf ("Not enough memory!");exit(2);}
    }
    while(____(3)____)                //סורק את שני המערכים
    {
        if(A[i].d.id<B[j].d.id )      ____ (4) ____;
        else
            if (A[i].d.id>B[i].d.id)  ____ (5) ____;
            else
            {
                ____ (4) ____;      ____ (5) ____;
            }
    }
    while(____(6)____)    *C[k++]=A[i++];
    while(____(7)____)    *C[k++]=B[j++];
    return C;
}

```


שאלה 4

לפניך הגדרות:

מטריצה T מסדר $M \times N$ תיקרא מטריצת טפליץ אם היא מקיימת את התנאי הזה:
לכל i ו- j כאשר: $1 \leq i \leq M-1$, $1 \leq j \leq N-1$ מתקיים: $T[i][j] = T[i-1][j-1]$.

דוגמה 1: להלן מטריצת טפליץ T בגודל 5×4 :

5	4	1	2	7
3	5	4	1	2
6	3	5	4	1
8	6	3	5	4

למטריצת טפליץ מסדר $M \times N$ ייתכנו לכל היותר $M + N - 1$ ערכים שונים, ולכן ניתן לאחסן את **מטריצת טפליץ** במערך חד-ממדי בגודל $M + N - 1$ והוא ייקרא **וקטור טפליץ**.

וקטור טפליץ יכיל את איברי השורה ה-0 של **מטריצת טפליץ** (בסדר שבו הם מופיעים בשורה זו) ואחריהן – כל איברי העמודה ה-0 (בסדר שבו הם מופיעים בעמודה זו), **חוץ מאשר את האיבר הראשון** שבעמודה זו.

דוגמה 2: בעבור מטריצת טפליץ T שבדוגמה 1, וקטור טפליץ הנו: 5 4 1 2 7 3 6 8

הערה: בסעיפים א'–ג' הנח שמטריצת טפליץ היא מסדר $M \times N$.

להלן הגדרות והכרזות בשפת C:

```
#define M 4
```

```
#define N 5
```

א.

לפניך פונקציה שכותרתה `void convertMatrix (int T[][N], int V[])` אשר מקבלת שני פרמטרים, T ו- V , כאשר T הנו מטריצת טפליץ. הפונקציה משכנת את וקטור טפליץ של T במערך חד-ממדי V .

בפונקציה 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.

```
void convertMatrix(int T[ ][N], int V[ ])
{
    int i, j;
    for(i=0; i < N; i++)
        V[i] = _____ (1) _____;
    for (j = _____ (2) _____; j < M; j++)
        _____ (3) _____ = T[j][0];
}
```

ב.

המטריצה המוחלפת של מטריצה A מסדר $M \times N$ היא מטריצה המסומנת ב- A^T (מסדר $N \times M$) אשר מתקבלת כדלקמן:

שורה 0 של A הופכת להיות עמודה 0 ב- A^T .

שורה 1 של A הופכת להיות עמודה 1 ב- A^T .

:

:

וכן הלאה.

כלומר, לכל $0 \leq i < M$ השורה ה- i של A הופכת להיות עמודה i במטריצה A^T .

דוגמה 3: בעבור המטריצה T שבדוגמה 1 המטריצה המוחלפת שלה היא:

5	3	6	8
4	5	3	6
1	4	5	3
2	1	4	5
7	2	1	4

שים לב: המטריצה המוחלפת של מטריצת טפליץ היא גם מטריצת טפליץ.

לפניך פונקציה שכותרתה `void transposeMatrix (int V[], int VT[])` אשר מקבלת שני פרמטרים V ו-VT, כאשר V הוא וקטור טפליץ המייצג את המטריצה A. הפונקציה משכנת לתוך מערך חד-ממדי VT את וקטור טפליץ המייצג את המטריצה AT, ללא שחזור המטריצות A ו-AT.

בפונקציה 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.

```
void transposeMatrix(int V[ ],int VT[ ])
{
    int i,j;
    VT[0]=V[0];
    for (j=1;j<__(1)__;j++)    VT[j] = V[ __(2)___ ];
    for(i=1;i<N;i++)    VT[ __(3)___ ] = V[i];
}
```

ג. לפניך פונקציה שכותרתה `void addMatrix(int U[], int V[], int W[])` אשר מקבלת כפרמטרים שלושה מערכים חד-ממדיים U , V ו- W , כאשר U ו- V מכילים שני וקטורי טפליץ שמייצגים שתי מטריצות טפליץ A ו- B בהתאמה. הפונקציה משכנת לתוך W וקטור טפליץ המייצג מטריצה שהיא סכום המטריצות A ו- B ללא שחזור המטריצות A ו- B .

בפונקציה 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.

```
void addMatrix(int U[ ],int V[ ],int W[ ])
{
    int i;
    for(i=0; i < ____ (1) ____; i++)
        ____ (2) ____;
}
```

נושא ב': שפת סף (50 נקודות)

פרק שלישי (20 נקודות)

ענה על שאלה 5 – שאלת חובה.

שאלה 5

להלן תכנית אשר מאתרת במחרוזת נתונה רצף מסוים של תווים ומחליפה אותו ברצף אחר. התכנית מבצעת את פעולת האיתור בשלושה שלבים הבאים לידי ביטוי בשלושת הקטעים א' – ג' שלהלן.

```
DATA SEGMENT

MSG DB 'DEFDEFDEFDEF'

MSG_LEN=$-MSG

WHAT_TO_REPLACE DB 'DEF'

REPLACE_BY DB 'ABC'

LEN=$-REPLACE_BY

MONE DW MSG_LEN

DATA ENDS

SSEG SEGMENT STACK 'STACK'

DB 100H DUP(?)

SSEG ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA
```

```

START:MOV  AX, DATA

        MOV  DS, AX

        PUSH DS

        POP  ES

        LEA  DI, MSG

        LEA  SI, WHAT_TO_REPLACE

        MOV  BX, DI

;=====

AGAIN:  MOV  _____ (1) _____, [BX]

        CMP  AL, [SI]

        JE   _____ (2) _____

        INC  BX                                ; קטע א'

        DEC  MONE

        JNZ  AGAIN

        JMP  SOF

;=====

CHECK:  CMP  MONE, LEN

        JB   _____ (3) _____

        MOV  CX, LEN

        MOV  DI, BX                                ; קטע ב'

        REPE _____ (4) _____

        JNE  NEXT

;=====

```

```

MOV CX, LEN

MOV DI, BX

LEA SI, REPLACE_BY

REP MOVSB                                ; קטע ג'

MOV BX, DI

LEA SI, _____ (5) _____

SUB MONE, LEN

CMP MONE, _____ (6) _____

JE SOF

JMP AGAIN

NEXT:  LEA SI, WHAT_TO_REPLACE

INC BX

DEC MONE

JNZ AGAIN

;=====

SOF:   MOV AH, 4CH

INT 21H

CODE ENDS

END START

```

- א.** קטע א' של התכנית מחפש את התו 'D' במחרוזת DEFDEFDEFDEF. אם התו 'D' נמצא, אז התכנית תעבור לביצוע קטע ב', אחרת התכנית מסתיימת. בקטע א' חסרים שני ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחרתך את מספרי הביטויים החסרים (1) – (2), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.
- ב.** קטע ב' של התכנית בודק אם תת-המחרוזת DEF נמצאת במחרוזת DEFDEFDEFDEF שכתובת MSG. אם תת-המחרוזת DEF נמצאת במחרוזת DEFDEFDEFDEF, אז התכנית תעבור לביצוע קטע ג', אחרת התכנית מסתיימת. בקטע ב' חסרים שני ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחרתך את מספרי הביטויים החסרים (3) – (4), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.
- ג.** קטע ג' של התכנית מחליף את המחרוזת DEF במחרוזת ABC. בקטע ג' חסרים שני ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחרתך את מספרי הביטויים החסרים (5) – (6), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

פרק רביעי (30 נקודות)

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

שאלה 6

לפניך תכנית אשר מקרינה על הצג את המספר NUM בן 16 סיביות בייצוג בינארי.

דוגמה: בעבור המספר 49 התכנית תציג כפלט את 0000000000110001.

```
SSEG SEGMENT STACK 'STACK'
    DB 100H DUP(?)
SSEG ENDS

CODE SEGMENT
    ASSUME CS:CODE
    NUM DW 81H
START: MOV CX, _____ (1) _____
        MOV AH, _____ (2) _____
AGAIN: ROL NUM, 1
        PUSH NUM
        JNC ZERO
        _____ (3) _____
        INT 21H
        JMP NEXT
ZERO:  _____ (4) _____
        INT 21H
NEXT: LOOP AGAIN
SOF:  MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

- א. בתכנית הנתונה חסרים **ארבעה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחברתך את מספרי הביטויים החסרים (1) – (4), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.
- ב. אם נחליף את ההוראה `ROL NUM, 1` בהוראה `SHL NUM, 1`, האם שינוי זה יכול להשפיע על הערך שיוקדן על הצג? ענה "כן" או "לא".
- ג. אם נחליף את ההוראה `ROL NUM, 1` בהוראה `SHL NUM, 1`, האם שינוי זה יכול להשפיע על תוכנו של `NUM`? ענה "כן" או "לא".
- ד. הנח כי תוכנו של האוגר `SP` הוא: 1004. העתק למחברתך את הטבלה הבאה המייצגת אזור מסוים של מחסנית:

תוכן הבית	כתובת הבית
.	.
.	.
.	.
.	.
	1000
	1001
	1002
	1003
	1004

לאחר ביצוע שני צעדי לולאה, רשום בטבלה את התוכן בבסיס הקסדצימלי של הבתים שכתובתם 1003H ועד (וכולל) 1000H בלבד.

שאלה 7

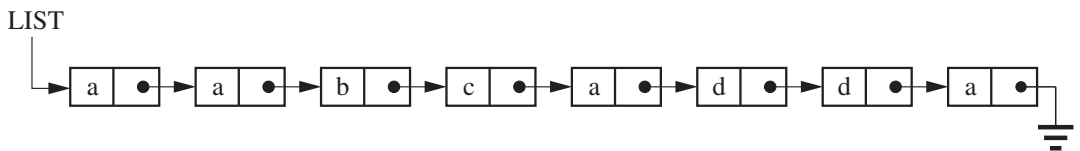
נתונה רשימה מקושרת **לא ריקה** הבנויה משני צמתים **לפחות**. כל צומת ברשימה מכיל את שני השדות האלה:

info - שדה מידע (אינפורמציה), שגודלו 8 ביטים, המכיל תו.

next - המצביע אל הצומת הבא ברשימה, שגודלו מילה (16 ביטים).

המצביע next בצומת האחרון הוא 0.

נוסף על כך נתון כי המשתנה LIST מכיל את הכתובת של הצומת הראשון ברשימה



לפניך קטע התכנית שלהלן:

```
START:MOV AX,DATA

      MOV DS,AX

      MOV DI,LIST

      MOV BX,[DI+1]

NEXT:  CMP WORD PTR[BX+1],0

      JE EXIT

      MOV BX,[BX+1]

      CMP WORD PTR[BX+1],0

      JE EXIT

      MOV BX,[BX+1]

      MOV DI,[DI+1]

      JMP NEXT

EXIT:  MOV BX,[DI+1]

      MOV WORD PTR [DI+1],0

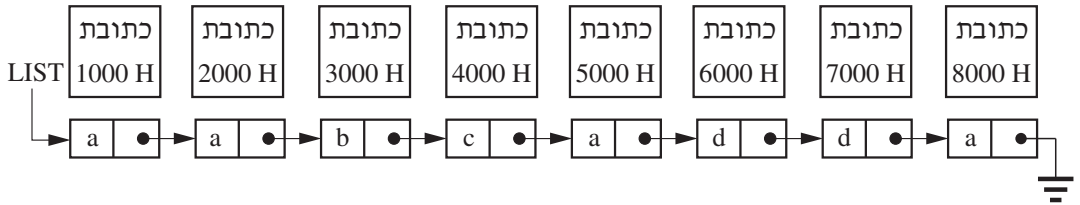
SOF:  MOV AH,4CH

      INT 21H

CODE ENDS

END START
```

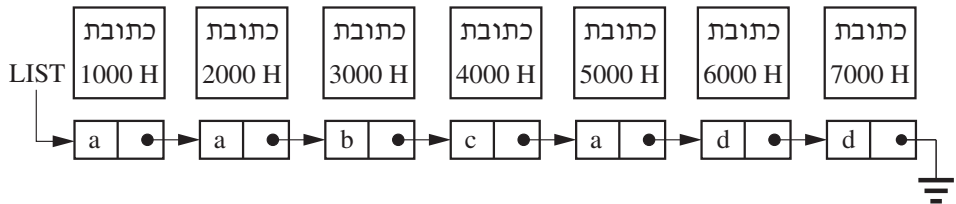
א. הרץ ידנית את קטע התכנית על הרשימה שלהלן:



מעל כל צומת רשום מספר (בבסיס 16) המציין את כתובתו של הצומת.

רשום במחברתך טבלת מעקב בעבור האוגרים BX ו-DI עד שהתכנית מגיעה לתווית SOF.

ב. הרץ ידנית את קטע התכנית על הרשימה שלהלן:



מעל כל צומת רשום מספר (בבסיס 16) המציין את כתובתו של הצומת.

רשום במחברתך טבלת מעקב בעבור טבלת מעקב בעבור האוגרים BX ו-DI עד שהתכנית מגיעה לתווית SOF.

ג. אם נחליף את ההוראה `MOV DI, LIST` בהוראה `LEA DI, LIST`, האם ישתנו ביצועי התכנית? ענה "כן" או "לא".

שאלה 8

בהינתן מספר שלם וחיובי ניתן ליצור עבורו סדרה $x_1 \dots x_n$ באופן הזה:

בסדרה זו קיים n שעבורו $x_n = 1$.

$$x_{i+1} = \begin{cases} \frac{x_i}{2} & x_i \text{ זוגי} \\ 3x_i + 1 & x_i \text{ אי-זוגי} \end{cases}$$

הערות:

- סדרה כזו תמיד מסתיימת במספר 1.

- n הוא מספר האיברים בסדרה.

דוגמה: עבור המספר 7 נקבל:

$7 \rightarrow 22 \rightarrow 11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

ובעבור המספר 20 נקבל:

$20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

לפניך תכנית **רקורסיבית** המקבלת מספר שלם חיובי שהוא המספר הראשון בסדרה.

התכנית יוצרת סדרה, כמתואר לעיל, ומציבה במערך ARR את כל המספרים המתקבלים בסדרה זו מן הסוף להתחלה. עבור המספר 7 המערך ARR יכיל את המספרים האלה (קרא משמאל לימין):

1, 2, 4, 8, 16, 5, 10, 20, 40, 13, 26, 52, 17, 34, 11, 22

הנחת יסוד: גודלו של המערך ARR מספיק לאחסון המספרים המתקבלים בסדרה.

התכנית שלהלן נעזרת בשגרה רקורסיבית בשם SIDRA. בשגרה זו חסרים **שבעה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחברתך את מספרי הביטויים החסרים (1) – (7), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
SSEG SEGMENT STACK 'STACK'

    DB 100H DUP(?)

SSEG ENDS

CODE SEGMENT

    ASSUME CS:CODE

    NUM DW 5

    ARR DW 100 DUP(0)

START:  PUSH CS

        POP DS

        PUSH NUM

        LEA BX,ARR

        CALL SIDRA

SOF:    MOV AH,4CH

        INT 21H
```

// המשך התכנית בעמוד הבא

```
SIDRA:  PUSH BP

        MOV BP, SP

        MOV AX, [BP+4]

        _____ (1) _____

        JE SI_EXIT

        TEST AX, 1

        JNZ SI_ODD

        _____ (2) _____ AX, 1

        JMP _____ (3) _____

SI_ODD:  ADD AX, AX

        ADD AX, _____ (4) _____

        ADD AX, 1

SI_NEXT: _____ (5) _____

        CALL SIDRA

SI_EXIT: MOV AX, [BP+4]

        MOV _____ (6) _____, AX

        INC BX

        INC BX

        POP BP

        RET _____ (7) _____

CODE ENDS

END START
```

בהצלחה!