

## **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 38 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 (24 נקודות)

ענה על שאלה 1 (סעיפים א'–ח') – שאלת חובה.

#### שאלה 1

ביום ספורט שמתקיים בבית-הספר מופעלות K תחנות שונות הממוספרות בין 0 ל-1 K (כולל).  
ביום הזה משתתפות G קבוצות הממוספרות בין 0 ל-1 G (כולל). בכל קבוצה 10 תלמידים.

K ו-G קבועים וידועים מראש, כאשר  $K > G > 0$ .

כל קבוצה עוברת בכל התחנות. התלמידים בכל אחת מן הקבוצות עוברים יחד מתחנה אחת לתחנה אחרת. בכל תחנה, כל אחד ממשתתפי הקבוצה מבצע מטלה ועליה הוא מקבל ציון בתחום שבין 0 ל-15 (כולל).

כל תחנה היא בעלת משקל קבוע וידוע מראש בתחום שבין 0 ל-1 (לא כולל 0 ו-1). סכום המשקלים בעבור כל התחנות הוא 1.

**הציון של קבוצה** בתחנה כלשהי הוא ממוצע הציונים של כל חברי הקבוצה בתחנה זו, מוכפל במשקל התחנה. **הציון הכולל** של קבוצה הוא סכום **ציוניה** בכל K התחנות.

בסיום יום הספורט מתפרסמת טבלה הכוללת את כל הקבוצות הממוינת בסדר יורד, על-פי **הציונים הכוללים** (הניחו שאין שתי קבוצות שקיבלו **ציון כולל** זהה).

**הערות:** תחנה תיקרא **תחנה פנויה** אם אף אחת מהקבוצות לא נמצאת בה.  
קבוצה תיקרא **קבוצה פנויה** אם היא לא משובצת לתחנה כלשהי.

להלן ייצוג של טיפוס הנתונים **יום ספורט**:

יום ספורט הוא מבנה (רשומה) המכיל:

- **מונה קבוצות**, שהנו משתנה מטיפוס שלם (int) אשר מונה את מספר הקבוצות שסיימו לעבור בכל התחנות (ערך השווה ל-G ייתן אינדיקציה שיום הספורט הסתיים);
- **מערך משקלים** בגודל K שכל איבר בו מטיפוס ממשי (float) המכיל את משקלי התחנות (קבועים וידועים מראש);
- **מערך תחנות** בגודל K שכל איבר בו הוא מטיפוס **תחנה** (יפורט בהמשך);
- **מערך קבוצות** בגודל G שכל איבר בו הוא מטיפוס **קבוצה** (יפורט בהמשך);

להלן הגדרות והכרזות בשפת C של טיפוס הנתונים **יום ספורט**:

```
#define K 15 // Number of stations
#define G 12 // Number of groups
#define N 10 // Number of students in each group;
#define MAXGRADE 15 // Maximum grade
typedef enum {FALSE,TRUE} boolean;
```

```
typedef struct sportDayType // טיפוס יום ספורט
{
    int passedGroups;          // מונה קבוצות
    float weights[K];          // מערך משקלים
    station stations[K];       // מערך תחנות
    group groups[G];           // מערך קבוצות
} sportDay ,*dayPtr;
```

להלן הגדרה בשפת C של המבנה (רשומה) **תחנה**:

```
typedef struct stationType // טיפוס תחנה
{
    int currentGroup; // אם התחנה פנויה, המשתנה יכיל ערך (1 -). אחרת יכיל את
                     // מספר הקבוצה שנמצאת בתחנה זו

    int passed; // מספר התלמידים מהקבוצה שנמצאת בתחנה זו אשר סיימו את
               // מטלותיהם

    int acumulatedGrade; // צובר את ציוני כל התלמידים שסיימו את מטלותיהם בתחנה זו
                       // השייכים לקבוצה שנמצאת בתחנה זה עתה

    boolean idle; // יכיל ערך "אמת" אם התחנה פנויה, ולא - יכיל את הערך "שקר"
}station;
```

להלן הגדרה בשפת C של המבנה (רשומה) **קבוצה**:

```
typedef struct groupType // טיפוס קבוצה
{
    int groupNo; // מספר קבוצה

    float averageGroupGrade; // ציון כולל של הקבוצה

    int currentStation; // מספר התחנה שבה נמצאת הקבוצה

    boolean stationsPassed[K]; // תא i במערך זה יכיל את הערך "אמת" אם הקבוצה
                              // ביקרה בתחנה ה-i ולא יכיל את הערך "שקר"

    boolean busy; // יכיל ערך "שקר" אם הקבוצה פנויה, ולא - יכיל את הערך "אמת"

    }group ,*groupPtr;
```

בתכנית מוגדר **המשתנה הגלובלי** שלהלן:

```
sportDay D;
```

להלן הפעולה **אתחל יום ספורט** (init) אשר מתבצעת בפתיחת יום הספורט:

```
void init(void)
{
    int i,j;
    D.weights[ ] ={0.045,0.1,0.2,0.05,0.15,0.05,0.08,0.1,0.04,0.03,
                    0.015,0.05,0.05,0.02,0.07};
    D.passedGroups = 0;
    for (i=0;i < G ;i++ )
    {
        D.groups[i].groupNo = i;           // מספר קבוצה
        D.groups[i].averageGroupGrade=0;
        D.groups[i].currentStation = i; // i-ית משובצת לתחנה ה-
        D.groups[i].busy = TRUE; // קבוצה ה- i לא פנויה כיוון שהיא משובצת בתחנה
        for(j=0;j < K;j++)
            D.groups[i].stationsPassed[j] = FALSE; // תלמידי הקבוצה לא סיימו
                                                    // את מטלתם בתחנה ה- j

        D.stations[i].currentGroup = i; // i נמצאת בקבוצה i
        D.stations[i].passed = 0;
        D.stations[i].acumulatedGrade = 0;
        D.stations[i].idle = FALSE;      // תחנה ה- i לא פנויה כי קבוצה i נמצאת בה
    }
    // טיפול בתחנות שעדיין לא שובצו בהן קבוצות
    for(j = G;j < K;j++)
    {
        D.stations[j].currentGroup = -1;
        D.stations[j]. passed = 0;
        D.stations[j].acumulatedGrade = 0;
        D.stations[j].idle = TRUE;
    }
}
```

נתונה ספרייה לטיפוס הנתונים **יום ספורט** המכילה את הפעולות האלה:

תיעוד	הפעולה
פעולה המאתחלת את יום הספורט.	אתחל_יום_ספורט()  void <b>init</b> (void)
הפעולה מקבלת את מספר התחנה stationNo, ומחזירה ערך "אמת" אם התחנה stationNo פנויה; אחרת היא מחזירה את הערך "שקר".  הנחה: $0 \leq \text{stationNo} < K$	האם תחנה פנויה? (stationNo)  boolean <b>isStationIdle</b> (int stationNo)
הפעולה מקבלת את מספר הקבוצה groupNo, ומחזירה ערך "אמת" אם הקבוצה groupNo פנויה; אחרת היא מחזירה את הערך "שקר".  הנחה: $0 \leq \text{groupNo} < G$	האם קבוצה פנויה? (groupNo)  boolean <b>isFree</b> (int groupNo)
הפעולה מקבלת את מספר התחנה stationNo, ומעדכנת את התחנה stationNo למצב פנוי.  הנחה: $0 \leq \text{stationNo} < K$	עדכן תחנה למצב פנוי (stationNo)  void <b>updateIdleStation</b> (int stationNo)
הפעולה מקבלת את מספר הקבוצה groupNo, ומעדכנת את הקבוצה groupNo למצב פנוי.  הנחה: $0 \leq \text{groupNo} < G$	עדכן קבוצה למצב פנוי (groupNo)  void <b>freeGroup</b> (int groupNo)
הפעולה מקבלת את מספר התחנה stationNo, ואת הציון grade, ומוסיפה את הציון grade לתחנה stationNo.  הנחה: $0 \leq \text{stationNo} < K$	הוסף ציון לתחנה (stationNo, grade)  void <b>acumulateGradeInStation</b> (int stationNo, int grade)
הפעולה מקבלת את מספר הקבוצה groupNo, ואת מספר התחנה stationNo וקובעת שקבוצה groupNo ביקרה בתחנה stationNo.	עדכן קבוצה סיימה תחנה (int groupNo, int stationNo)  int <b>increseStationsPassed</b> (int groupNo, int stationNo)
הפעולה מקבלת את מספר התחנה stationNo, ומחזירה ערך "אמת" אם כל חברי הקבוצה שנמצאת בתחנה stationNo סיימו את מטלותיהם בתחנה זו; אחרת היא מחזירה את הערך "שקר".  הנחה: $0 \leq \text{stationNo} < K$	קבוצה סיימה תחנה? (stationNo)  boolean <b>groupPassed</b> (int stationNo)

<p>הפעולה מקבלת את מספר התחנה stationNo ואת מספר הקבוצה groupNo, ומכניסה את קבוצה groupNo לתחנה stationNo.</p> <p>הנחה: <math>0 \leq \text{stationNo} &lt; k</math>, <math>0 \leq \text{groupNo} &lt; G</math></p>	<p>הכנס קבוצה לתחנה (stationNo, groupNo)</p> <p>void <b>insertGroup</b>(int stationNo,int groupNo)</p>
<p>הפעולה מקבלת את מספר התחנה stationNo ואת מספר הקבוצה groupNo, שכל תלמידה סיימו זה עתה את מטלתם בתחנה stationNo. הפעולה מחשבת את הציון המשוקלל של התחנה stationNo ומוסיפה אותו לציון הכולל של הקבוצה groupNo.</p> <p>הנחה: <math>0 \leq \text{groupNo} &lt; G</math>, <math>0 \leq \text{stationNo} &lt; K</math></p>	<p>הוסף ציון לקבוצה (stationNo, groupNo)</p> <p>void <b>addGradeToGroup</b>(int stationNo,int groupNo)</p>
<p>הפעולה מוסיפה 1 למונה הקבוצות בעבור קבוצות שסיימו את כל התחנות.</p>	<p>הוסף קבוצה למונה הקבוצות שסיימו()</p> <p>void <b>updatePassedGroups</b>(void)</p>
<p>הפעולה מקבלת את מספר התחנה stationNo, ומחזירה את מספר הקבוצה הנמצאת בה. אם התחנה פנויה, אז הפעולה מחזירה -1.</p> <p>הנחה: <math>0 \leq \text{stationNo} &lt; K</math></p>	<p>החזר קבוצה בתחנה (stationNo)</p> <p>int <b>currentGroupInStation</b>(int stationNo)</p>
<p>הפעולה מקבלת את מספר התחנה stationNo, ומחזירה את מספר הקבוצה הפנויה שצריכה להיכנס לתחנה stationNo. אם לא קיימת קבוצה פנויה, אז הפעולה מחזירה -1.</p> <p>הנחה: <math>0 \leq \text{stationNo} &lt; K</math></p>	<p>מצא קבוצה פנויה לתחנה (stationNo)</p> <p>int <b>findnextGroupToEnter</b>(int stationNo)</p>
<p>הפעולה מקבלת את מספר הקבוצה groupNo, ומחזירה את מספר התחנה הפנויה שבה עדיין לא ביקרה הקבוצה groupNo. אם לא קיימת תחנה כזו, אז הפעולה מחזירה -1.</p> <p>הנחה: <math>0 \leq \text{groupNo} &lt; G</math></p>	<p>מצא התחנה הבאה של קבוצה (groupNo)</p> <p>int <b>findnextStationToEnter</b>(int groupNo)</p>
<p>הפעולה מקבלת את מספר הקבוצה groupNo, ומחזירה ערך "אמת" אם הקבוצה סיימה את כל התחנות; אחרת היא מחזירה את הערך "שקר".</p> <p>הנחה: <math>0 \leq \text{groupNo} &lt; G</math></p>	<p>הקבוצה סיימה את כל התחנות? (groupNo)</p> <p>boolean <b>groupPassedAll</b>(int groupNo)</p>
<p>הפעולה מחזירה את הערך "אמת" אם כל הקבוצות סיימו את כל התחנות; אחרת היא מחזירה את הערך "שקר".</p>	<p>סוף יום ספורט ()</p> <p>boolean <b>EndOfDay</b>(void)</p>

שים לב:

הנח כי הפעולות שבספרייה הנתונה כבר כתובות וניתן להשתמש בהן בכל הסעיפים הבאים.  
חלק מהפעולות שבספרייה זו ימומשו בסעיפים א'–ז' שלהלן.

כדי לענות על הסעיפים א'–ז' שלהלן השתמש בפעולות המצוינות בלועזית שבספרייה לטיפול הנתונים יום ספורט.

א. לפניך מימוש של הפונקציה `void insertGroup(int stationNo,int groupNo)`

בפונקציה 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.

```
void insertGroup(int stationNo,int groupNo)
{
    D.stations[stationNo].currentGroup =____(1)____;
    D.stations[stationNo].passed = 0;
    D.stations[stationNo].acumulatedGrade = 0;
    D.stations[stationNo].idle =____(2)____;
    ____ (3) ____ = TRUE;    // טיפול בקבוצה
    ____ (4) ____ = stationNo;
}
```



**ב. לפניך מימוש של הפונקציה**

```
void addGradeToGroup (int stationNo,int groupNo)
```

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

```
void addGradeToGroup(int groupNo,int stationNo)
```

```
{
    int grade;
    float gradef;
    grade = ____ (1) ____;
    gradef = (float) (grade * ____ (2) ____);
    ____ (3) ____ += gradef / N;
}
```

**ג. לפניך מימוש של הפונקציה**

```
.void increaseStationsPassed(int groupNo,int stationNo)
```

בפונקציה חסר ביטוי אחד, המסומן במספר בין סוגריים עגולים. רשום במחברתך את מספר הביטוי החסר (1), וכתוב לידו את הביטוי החסר שהוא מייצג.

```
void increaseStationsPassed(int groupNo,int stationNo)
```

```
{
    ____ (1) ____ = TRUE ;
}
```

ד. לפניך מימוש של הפונקציה `int findnextGroupToEnter (int stationNo)`

בפונקציה 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.

```
int findnextGroupToEnter (int stationNo)
{
    int i;

    boolean visited;

    int nextGroup = -1;

    int groupNum;

    for(i=0; i < G; i++)
    {
        groupNum = D.groups[i].groupNo;
        visited = ____ (1) ____;
        if ( ____ (2) ____ && !visited )
        {
            nextGroup = ____ (3) ____;
            ____ (4) ____;
        }
    }

    return nextGroup;
}
```

ה. לפניך מימוש של הפונקציה `int findnextStationToEnter (int groupNo)`

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

```
int findnextStationToEnter(int groupNo)
{
    int i;
    int nextStation = -1;
    for(i=0; i<K; i++)
    {
        if ( _____ (1) _____ && _____ (2) _____ )
        {
            nextStation = i;
            break;
        }
    }
    _____ (3) _____;
}
```

1. לפניך מימוש של הפונקציה `boolean groupPassedAll (int groupNo)`.

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

```
boolean groupPassedAll(int groupNo)
{
    boolean status ;
    int i;
    for(i=0;i < ____ (1) ____;i++)
    {
        if ( ____ (2) ____ ==FALSE) break;
    }
    status = ____ (3) ____;
    return status;
}
```

2. לפניך מימוש של הפונקציה `boolean EndOfDay (void)`.

בפונקציה חסר ביטוי, המסומן במספר בין סוגריים עגולים. רשום במחברתך את מספר הביטוי החסר (1) וכתוב לידו את הביטוי החסר שהוא מייצג.

```
boolean EndOfDay(void)
{
    boolean status;
    status = ____ (1) ____;
    return status;
}
```

ח. במהלך יום הספורט, האחראי על יום הספורט מזין את הנתונים למערכת. הנתונים הם זוגות של מספרים שלמים, כאשר בכל זוג נתון המספר הראשון הוא מספר התחנה והמספר השני הוא הציון שקיבל תלמיד כלשהו בתחנה זו.

לפניך פונקציה בשם `main()` לניהול יום ספורט.

באלגוריתם 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.

```
int main()
{
    int groupNo, stationNo, nextStation, nextGroup;
    int grade;

    init();
    while (!EndOfDay())
    {
        printf("\nEnter Station Number:");
        scanf ("%d" , &stationNo);
        printf ("\nEnter grade: ");
        scanf ("%d", &grade);
        groupNo=currentGroupInStation(stationNo);
        if(isStationIdle(stationNo)) continue;
        acumulateGradeInStation(stationNo, grade);
        if (groupPassed(stationNo))
        {
            _____ (1) _____ ;
            addGradeToGroup (stationNo, groupNo);
        }
    }
}
```

```
increaseStationsPassed(groupNo, stationNo);
nextGroup = ____ (2) ____;
if(nextGroup !=-1)
    insertGroup (stationNo, nextGroup);
else
    ____ (3) ____;
if(!groupPassedAll(groupNo))
{
    nextStation = ____ (4) ____;
    if(nextStation!= -1)
        ____ (5) ____;
}
else
    ____ (6) ____;
}
} //of while
return 0;
} //of main()
```

## פרק שני (26 נקודות)

ענה על שתיים מבין השאלות 2 – 4 (לכל שאלה – 13 נקודות).

### שאלה 2

בשאלה זו חמישה סעיפים (א'–ה') שאינם תלויים זה בזה. עליך לענות על כולם.

א. רשום במחברתך את הפלט המדויק של התכנית שלהלן.

```
#include <stdio.h>

#include <string.h>

void func1(char *s, char c);

void main()

{

    char * name1="abracadabra";

    func1(name1, 'a');

    printf("%s", name1);

}

void func1(char *s, char c)

{

    while(*s)

    {

        if(*s==c)

            strcpy(s, s+1);

        else

            s++;

    }

}
```

ג. רשום במחברתך את הפלט המדויק של התכנית שלהלן.

```
#include <stdio.h>

#include <string.h>

int func2(char *s,int n);

void main()
{
    int i;

    char * name3 = "abracadabra";

    i = func2(name3,4);

    printf("%s  %d ",name3, i);
}

int func2(char *s,int n)
{
    int len;

    len = strlen(s);

    if (len < n)

        return 0;

    else

        s[len-n] = '\\0';

    return 1;
}
```



ג. רשום במחברתך את הפלט המדויק של התכנית שלהלן.

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

#define N 20

void main()
{
    char s[3][N]= {"exval", "yoram", "bacar"};
    char res[N]= {" "};
    int i, cnt;
    for(i=0; i<3; i++)
        strcat(res, &s[i][i+1]);
    cnt = strlen(res);
    printf("%s  %d", res, cnt);
}
```

ד. לפניך תכנית:

```
#include <stdio.h>

#include <string.h>

void main()
{
    char str1[21] ="abcaxa";
    char str2[21] ="xabz";
    char str3[41];
    int i,j,t,length;

    strcpy(str3,str1);
    strcat(str3,str2);
    length = strlen(str3);
    for(i=0;str3[i]!='\0';i++)
        for(j=length; j > i;j--)
            if(str3[j]==str3[i])
                for(t=j;t<length;t++) str3[t]=str3[t+1];
    printf("The new string is: %s\n",str3);
    return 0;
}
```

1. רשום במחברתך פלט מדויק של התכנית שלעיל.

2. רשום במחברתך את מספרו של ההיגד הנכון מבין ההיגדים הבאים:

א. התכנית מוצאת ומדפיסה מחרוזת (str3) המכילה את התווים השייכים ל-str1 או ל-str2, כך שכל תו ב-str3 מופיע פעם אחת בלבד.

ב. התכנית מוצאת ומדפיסה מחרוזת (str3) המכילה את התווים השייכים ל-str1 וגם ל-str2, כך שכל תו ב-str3 מופיע פעם אחת בלבד.

ג. התכנית מוצאת ומדפיסה מחרוזת (str3) המכילה את התווים השייכים ל-str2 אך אינם שייכים ל-str1 וכל תו ב-str3 מופיע פעם אחת בלבד.

ד. התכנית מוצאת ומדפיסה מחרוזת המכילה את התווים השייכים ל-str1 אך אינם שייכים ל-str2 וכל תו ב-str3 מופיע פעם אחת בלבד.

ה. רשום במחברתך פלט מדויק של התכנית שלהלן.

```
#define SM(a,b) (a) * ((a) + (b))
```

```
void main()  
{  
    int x=3,y=4,z;  
    z = SM(x,y);  
    printf("\n\t x=%d \t y=%d \t z =%d ",x,y,z);  
    z=SM(x++,--y);  
    printf("\n\t x=%d \t y=%d \t z =%d ",x,y,z);  
}
```

### שאלה 3

במבחן דרכים השתתפו שש מכוניות הממוספרות בין 0 ל-5 (כולל). כל מכונית נבחנה בחמש קטגוריות הממוספרות בין 0 ל-4 (כולל) שהן: תאוצה (A), צריכת דלק (B), בלימת זעזועים (C), עיצוב פנים (D) ומחיר (E).

לכל קטגוריה  $j$  משקל  $P_j$ , כאשר  $0 \leq j \leq 4$  וגם  $\sum_{j=0}^4 P_j = 100$ .

נסמן ב- $T_{ij}$  את הציון של המכונית ה- $i$  בקטגוריה ה- $j$ , כאשר  $0 \leq j \leq 4$ ,  $0 \leq i \leq 5$ .

וגם  $0 \leq T_{ij} \leq 100$ . כל מכונית מקבלת ציון בכל אחת מהקטגוריות, ולבסוף משקללים עבור המכונית את **הציון הסופי**.

$$\frac{\sum_{j=0}^4 T_{ij} \times P_j}{100}$$

**הציון הסופי** בעבור המכונית ה- $i$  לכל  $0 \leq i \leq 5$  הנו:

$$\frac{\sum_{i=0}^5 T_{ij}}{6}$$

**הציון הממוצע** בעבור הקטגוריה ה- $j$  לכל  $0 \leq j \leq 4$  הנו:

תוצאות המבחן נרשמו בבסיס נתונים ממוחשב, הכולל את המערכים שלהלן המתבססים על טיפוס הנתונים וההגדרות האלה:

```
#define NUMCARS 6

#define NUMCAT 5

typedef struct categoryType // טיפוס קטגוריה
{
    char name[12];           // שם קטגוריה
    float average;           // ציון ממוצע
}category, *catPtr;
```

- מערך של קטגוריות:

```
category categories[NUMCAT] ;
```

- מערך של משקלי הקטגוריות באחוזים:

```
int weights[NUMCAT] = {30,20,25,15,10};
```

**דוגמה:** אם  $\text{weights}[2]=25$ , אזי המשקל של הקטגוריה מס' 2 הוא 25 אחוז.

- מערך דו-ממדי שבו נרשמו הציונים בקטגוריות השונות בעבור המכוניות השונות:

```
int testArr[NUMCARS][NUMCAT] = {{7,8,7,7,6},
                                   {5,8,9,9,4},
                                   {8,8,8,8,8},
                                   {9,8,8,9,8},
                                   {7,7,8,8,7},
                                   {5,5,5,5,5}};
```

**לדוגמה:** אם  $\text{testArr}[2][4] = 8$ , אז בקטגוריה מס' 4 קיבלה מכונית מס' 2 את הציון 8.

- מערך בשם:

```
int bestQualities[NUMCAT];
```

המכיל את מספרי המכוניות כאשר  $\text{bestQualities}[j]$  - מכיל את מספר המכונית שהציון שלה הוא הגבוה ביותר בקטגוריה ה- $j$  לכל  $0 \leq j \leq 4$ .

- מערך בשם:

```
float carGrades[NUMCARS];
```

של ציונים סופיים בעבור המכוניות, כאשר  $\text{carGrades}[i]$  יכיל את הציון הסופי של המכונית ה- $i$  לכל  $0 \leq i \leq 5$ .

- מערך בשם:

```
char *catNames[NUMCAT] = {"A", "B", "C", "D", "E"}
```

המכיל את שמות הקטגוריות.

- מערך המכיל את שמות המכוניות:

```
char *names[NUMCARS]={"FORD", "KIA", "OPEL", "NISSAN", "TOYOTA", "SKODA"} ;
```

א. לפניך קטע קוד אשר מדפיס את פרטי כל הקטגוריות הממוינים בסדר עולה לפי הציון הממוצע שלהן. בקטע הקוד מזמנים פונקציה בשם incmp אשר תפורט לאחר קטע הקוד.

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

```
/* Init categories */
for (j= 0;j < NUMCAT;j++)
{
    strcpy(categories[j].name ,catNames[j]);
    categories[j].average = 0;
}

/* averages per category */
for(j=0;j < NUMCAT;j++)
{
    for (i=0;i< NUMCARS ;i++)
        categories[j].average += (float)testArr[i][j];
    _____(1)_____ ;
}

qsort(_____(2)_____);

printf("\n Category \t Average ");

for (j = 0;j < NUMCAT ; j++)
    printf(" \n%s \t \t%5.2f",categories[j].name,
        categories[j]. average);
```

```
int intcmp(____ (3) ____)  
{  
    catPtr p = v1;  
    catPtr q = v2;  
    return ( p->average - q->average) ;  
}
```

ב. לפניך קטע קוד אשר מחשב ציון משוקלל עבור כל מכונית ומאתר את המכונית בעלת הציון הסופי הגבוה ביותר ומדפיס את שמה ואת הציון הסופי שלה (הנח שקיימת מכונית אחת כזו).

בפונקציה 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.

```
/*Calculate grades * /  
  
for(i=0;i < NUMCARS;i++)  
{    for (j=0;j < NUMCAT;j++)  
        carGrades[i] +=____ (1) ____;  
    carGrades[i] /= 100;  
}  
  
/* best Car*/  
  
bestCar = 0;  
  
for(i=1;i < NUMCARS;i++)  
    bestCar=____ (2) ____? i:bestCar;  
  
printf("\n\nThe best car is %s with grade %5.2f",  
        ____ (3) ____, ____ (4) ____);
```

ג. לפניך קטע קוד אשר מדפיס לכל קטגוריה את שמה ואת שם המכונית שקיבלה בקטגוריה זו את הציון הגבוה ביותר (הנח שקיימת מכונית אחת בלבד בעלת הציון הגבוה ביותר בכל קטגוריה).

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.

```
for (j=0; j < NUMCAT; j++)
{
    bestQ =0;
    for (i=1; i< ____ (1) ____; i++)
        bestQ = (testArr[i][j] > ____ (2) ____)? i:bestQ;
    ____ (3) ____;
}

for (j=0; j < NUMCAT; j++)
    printf("\n\n Best quailty %s has the Car %s ",
        categories[j].name, ____ (4) ____);
getchar();
return 0 ;
}
```



## שאלה 4

בשאלה זו שלושה סעיפים, א'–ג', שאינם תלויים זה בזה. עליך לענות על כולם.

א. לפניך קטע קוד בשפת C אשר מניח את קיומה של רשימה מקושרת חד-כיוונית שלראשה מצביע `head`.

```
#include <stdio.h>
#include <stdlib.h>
#define N 4
typedef struct eivarType // טיפוס צומת ברשימה מקושרת חד-כיוונית
{
    int num;
    struct eivarType *next;
}eivar, *eivarPtr;

eivarPtr cList(eivarPtr head, int n);

void main()
{
    eivarPtr head=NULL;
    eivarPtr newHead,iPtr;
    eivarPtr p;
    int inNum;
    printf("\n The Original List is :\n");
    for(iPtr = head; iPtr!= NULL; iPtr = iPtr->next)
        printf("%d ",iPtr->num);
    newHead = cList(head,N);
    printf("\n The New List is : \n");
    for(iPtr=newHead;iPtr !=NULL;iPtr = iPtr->next)
        printf("%d :",iPtr->num);
    getchar(); return 0;
```

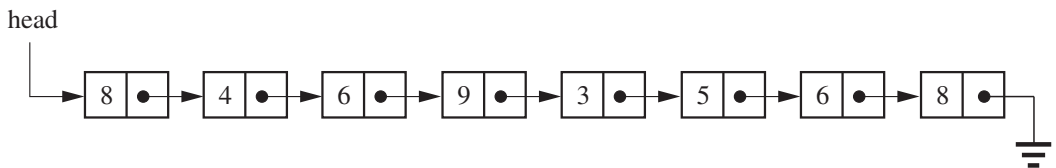
```

}

eivarPtr cList(eivarPtr head,int n)
{ eivarPtr prev,p ,newHead,end;
  int i;
  prev = head;
  p = head;
  if (n==1) return head;
  for(i=1;i < n;i++)
  {
    prev = p;
    p = p->next;
    if(p==NULL) return head;
  }
  for(end = head; end->next != NULL;end = end->next);
  newHead = p;
  prev->next = NULL;
  end->next = head;
  return newHead;
}

```

1. רשום במחברתך את הפלט המדויק של התכנית שלעיל בעבור הרשימה שלהלן:



איור לשאלה 4 א'

2. להלן היגדים העוסקים בפונקציה `cList (eivarPtr head,int n)`.

רשום במחברתך את מספרו של ההיגד הנכון מבין ההיגדים הבאים:

- א. הפונקציה מעבירה את  $n - 1$  האיברים האחרונים לראש הרשימה באותו הסדר שבו היו ברשימה הנתונה.
- ב. הפונקציה יוצרת מהרשימה הנתונה רשימה מעגלית.
- ג. הפונקציה מעבירה את  $n - 1$  האיברים הראשונים לסוף הרשימה באותו הסדר שבו היו ברשימה הנתונה.
- ד. הפונקציה מעבירה את מחצית אברי הרשימה הנתונה לסוף הרשימה.

3. לפניך הפונקציה הבאה:

```
typedef struct nodeType // טיפוס צומת ברשימה מקושרת חד־כיוונית
{
    int key;
    struct nodeType *next;
}node, *nodePtr;

nodePtr TR7 (nodePtr list)
{
    nodePtr pev = NULL;
    nodePtr p=list,q;
```

```
while (p)
{
    q = p->next;
    p->next = pev;
    pev = p;
    p = q;
}
return pev;
}
```

להלן היגדים העוסקים בפונקציה (nodePtr list) **TR7** nodePtr.  
רשום במחברתך את מספרו של ההיגד הנכון מבין ההיגדים הבאים.

1. הפונקציה הופכת את הרשימה לרשימה מעגלית חד-כיוונית.
2. הפונקציה יוצרת באופן כללי שתי רשימות נפרדות כך שאחת מהן תכיל איבר אחד והשנייה תכיל את יתר האיברים.
3. הפונקציה ממיינת את איברי הרשימה הנתונה.
4. הפונקציה הופכת את סדר האיברים ברשימה הנתונה כך שהאיבר הראשון ברשימה ימוקם אחרון בה, והאיבר האחרון ברשימה ימוקם ראשון בה, וכן הלאה.

ג. לפניך פונקציה המקבלת מערך A של מספרים שלמים חיוביים. הפונקציה מחזירה מערך דינמי חדש המכיל את איברי המערך A ובו כל איבר מופיע פעם אחת בלבד.

המערך הדינמי יסתיים בזקיף EON, **לדוגמה**:

עבור המערך A הבא:

```
int A[ ] = {1,1,4,2,2,1,2,3,4,2,1,3}
```

הפונקציה תחזיר מערך **דינמי חדש** המכיל את המספרים: 3, 2, 4, 1 בלבד ולאחריהם הזקיף EON.

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

```
#define EON -1

int *func(const int A[ ],int n)
{
    int i,j,flag, *pArr = NULL, size=1;
    for(i = 0;i < n; i++)
    {
        flag = 0;
        for(j=0; j <____(1)____;j++)
            if(____(2)____)
            {
                flag = 1; '
                break;
            }
        if(!flag)
        {
            pArr = (int*)realloc (pArr,____(3)____);
            if (!pArr)
            {
                printf("\nNot enough memory");    exit(1);
            }
            pArr [size-2]= A[i];
        }
    }

    ____ (4) ____;
    ____ (5) ____;
}
```

## נושא ב': שפת סף (50 נקודות)

### פרק שלישי (20 נקודות)

ענה על שאלה 5 – שאלת חובה.

#### שאלה 5

בשאלה זו שני סעיפים שאינם תלויים זה בזה. ענה על כולם.

א. לפניך חלק מסגמנט הנתונים:

```
DATA SEGMENT
    ARR      DD      10 DUP (?)
    MSG      DB      'HAVE A NICE DAY$'
    ADR      DD      MSG1
DATA ENDS
DATA1 SEGMENT
    MA      DB      40 DUP (?)
    MSG1     DB      'KEEP SMILING$'
DATA1 ENDS
```

נתון כי **הכתובת הפיזית** של DATA SEGMENT היא 18AB0H. לפניך שש שאלות ולכל שאלה עליך לתת את התשובה המבוקשת בבסיס 16 (בסיס הקסאדצימלי).

1. מהו ההיסט של MSG?
2. מהו ההיסט של MSG1?
3. מהו התוכן של הבית שהיסטו ADR?
4. מהו התוכן של הבית שהיסטו ADR + 1?
5. מהו התוכן של הבית שהיסטו ADR + 2?
6. מהו התוכן של הבית שהיסטו ADR + 3?

**ב. לפניך תכנית:**

```
DATA SEGMENT
    ARR      DB      40 DUP (?)
    MSG      DB      'HAVE A NICE DAY$'
    ADR      DD      MSG1
DATA ENDS

DATA1 SEGMENT
    MA      DB      40 DUP (?)
    MSG1     DB      'KEEP SMILING$'
DATA1 ENDS

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

CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START:     MOV AX, 18ABH
            MOV DS, AX
            LEA DX, MSG      ; (*)
            MOV AH, 9
            INT 21H
            MOV AH, 4CH
            INT 21H
CODE ENDS
```

1. מהו הטקסט שיוקרא על הצג בתום ביצוע התכנית?
2. אם נחליף את הפקודה המסומנת ב- (\*) ב- `MOV DX, OFFSET MSG1`, מה יהיה הטקסט שיוקרא על הצג בתום ביצוע התכנית?

## פרק רביעי (30 נקודות)

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

### שאלה 6

לפניך תכנית אשר בודקת אם סכום הספרות העשרוניות הזוגיות ב־NUM שווה לסכום הספרות העשרוניות האי־זוגיות ב־NUM. אם כן, אז התכנית מציבה באוגר BL את הערך 0; אם לא – היא מציבה באוגר BL ערך **השונה מ־0**.

**דוגמה:** בעבור המספר 24765 התכנית תציב באוגר BL את הערך 0 כי סכום הספרות העשרוניות האי־זוגיות (7 + 5) שווה לסכום הספרות העשרוניות הזוגיות (2 + 4 + 6).

התכנית נעזרת בשגרה רקורסיבית בשם RECUR.

```
M            EQU            1

DATA SEGMENT

            NUM            DW            27645

DATA ENDS

SSEG SEGMENT STACK 'STACK'

            DB            100 DUP()

SSEG ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

            MOV AX,DATA

            MOV DS,AX

            XOR BX,BX        ;BL- sum of even BH sum of odd digitals

            MOV CX , _____(1)_____

            PUSH NUM

            CALL RECUR

            XOR BL,BH        ;if BL = BH then BL=0,else BL != 0
```



```

MOV AH, 4CH

INT 21H

RECUR:

PUSH BP

MOV BP, SP

MOV AX, [BP+4]

XOR DX, DX

DIV CX          ;          (*)

_____ (2) _____ DL, M

JNZ ODD

ADD _____ (3) _____, DL

_____ (4) _____

ODD:  ADD _____ (5) _____, DL

NEXT:  _____ (6) _____

CMP AX, 0

JE RE_SOF

_____ (7) _____

CALL RECUR

E_SOF: POP BP

RET 2

CODE ENDS

```

**א.** בתכנית הנתונה 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.

**ב.** אם נחליף את השורה DIV CX המסומנת ב-(\*) בשורה – DIV CL, האם שינוי זה יכול להשפיע על הערך המחוזר באוגר BL? ענה "כן" או "לא".

## שאלה 7

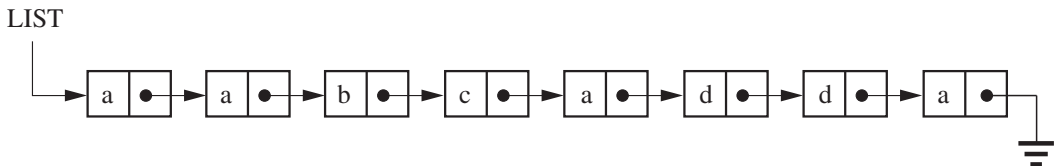
נתונה רשימה מקושרת **לא ריקה** הבנויה מצמתים. כל צומת ברשימה מכיל את שני השדות האלה:

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

next – מצביע 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 (ראה איור), והמשתנה TAV מכיל תו כלשהו.

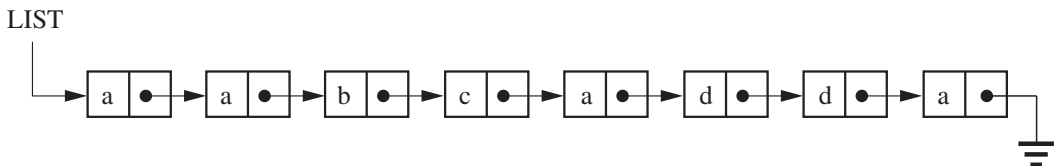


### איור א' לשאלה 7

לפניך קטע תכנית אשר מוחק ברשימה, שלראשה מצביע LIST, את כל הצמתים ששדה המידע (info) שלהם מכיל את התו הזהה לתו שבמשתנה TAV.

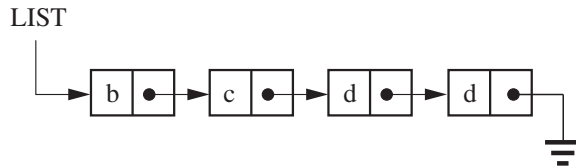
**דוגמה:**

בעבור הרשימה הבאה:



### איור ב' לשאלה 7

אם ה-TAV מכיל אות התו 'a', אזי LIST יצביע לראש הרשימה הבאה:



איור ג' לשאלה 7

```

SSEG SEGMENT STACK 'STACK'

        DB      100 DUP ( )

SSEG ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:   MOV AX,DATA

        MOV DS,AX

        PUSH LIST_ADDR

        PUSH TAV

        CALL SCAN

        POP LIST_ADDR

        MOV AH,4CH

        INT 21H

SCAN:    PUSH BP

        MOV BP,SP

        MOV AL,[BP+4]          ; AL=TAV

        MOV BX,[BP+6]          ; BX points to List

FIRST_LOOP: CMP [BX],AL

        JNE SECOND_LOOP        ; first node's info field != TAV
    
```

```
MOV BX, ____ (1) ____ ; remove this node from List
MOV ____ (2) ____, BX ; update List's address
CMP BX, 0
JE FINISH
JMP FIRST_LOOP

SECOND_LOOP:

CMP WORD PTR [BX+1], 0
JE FINISH

MOV DI, ____ (3) ____ ; DI points to next node
AG:
CMP [DI], AL
JNE NEXT_NODE

MOV DI, ____ (4) ____ ; remove this node
MOV ____ (5) ____, DI
CMP DI, 0
JE FINISH
JMP AG

NEXT_NODE:
MOV BX, ____ (6) ____
MOV DI, [DI+1]
____ (7) ____ ; is this the List's END ?
JE FINISH
____ (8) ____

FINISH:
POP BP ; (*)
RET ____ (9) ____

CODE ENDS

END START
```

א. בתכנית הנתונה חסרים **תשעה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחברתך את מספרי הביטויים החסרים (1) – (9), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

ב. אם נשמיט את הפקודה POP BP המסומנת ב-(\*) , האם ישתנו ביצועי התכנית? ענה "כן" או "לא".

## שאלה 8

בשאלה זו חמישה סעיפים שאינם תלויים זה בזה. ענה על כולם.

א. לפניך קטע קוד:

```
MOV CL , 2
PUSH AX
SHL AX , CL
POP BX
ADD AX , BX
```

לפני ביצוע קטע הקוד נתון ש- $AX = 42H$  ו- $DL = 05H$ . רשום במחברתך **פקודה אחת** המחליפה את הפקודות הנתונות שבקטע הקוד הנתון.

ב. לפניך קטע קוד אשר מחליף בין הסיביות 7 ו-0 בלבד בנתון הנמצא ב- $AX$  :

```
MOV M, 81H
TEST AL, M
JP SOF
____ (1) ____
SOF:
```

בקטע הקוד הנתון חסר ביטוי אחד, המסומן ב-(1). רשום במחברתך את מספר הביטוי החסר (1) וכתוב לידו את הביטוי החסר שהוא מייצג.

ג. לפניך קטע קוד:

```
MOV CX, 2
SHR AL, 1
SHL AL, CL
SHR AL, 1
```

לפני ביצוע קטע הקוד נתון ש- $AL = 0DBH$ .

1. מה יכיל האוגר AL אחרי ביצוע קטע הקוד? תן תשובתך בבסיס 16 (בסיס הקסאדצימלי).
2. רשום במחברתך **פקודה לוגית אחת** המחליפה את הפקודות הנתונות שבקטע הקוד הנתון.

ד. לפניך קטע קוד:

```
MOV CX, 8
ROR AX, CL
```

לפני ביצוע קטע הקוד נתון ש- $AX = 1A2BH$ .

1. מה יכיל האוגר AX לאחר ביצוע קטע הקוד? תן תשובתך בבסיס 16 (בסיס הקסאדצימלי).
  2. רשום במחברתך **פקודה אחת** המחליפה את זוג הפקודות.
- ה. לפניך הקטע הבא:

```
A DW 20 DUP (?)
```

1. רשום במחברתך **פקודה אחת** הטוענת לאוגר AX את **תוכן** המילה הרביעית של A.
2. רשום במחברתך **פקודה אחת** הטוענת לאוגר AX את **כתובת** המילה הרביעית של A.

## בהצלחה!