

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 39 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.

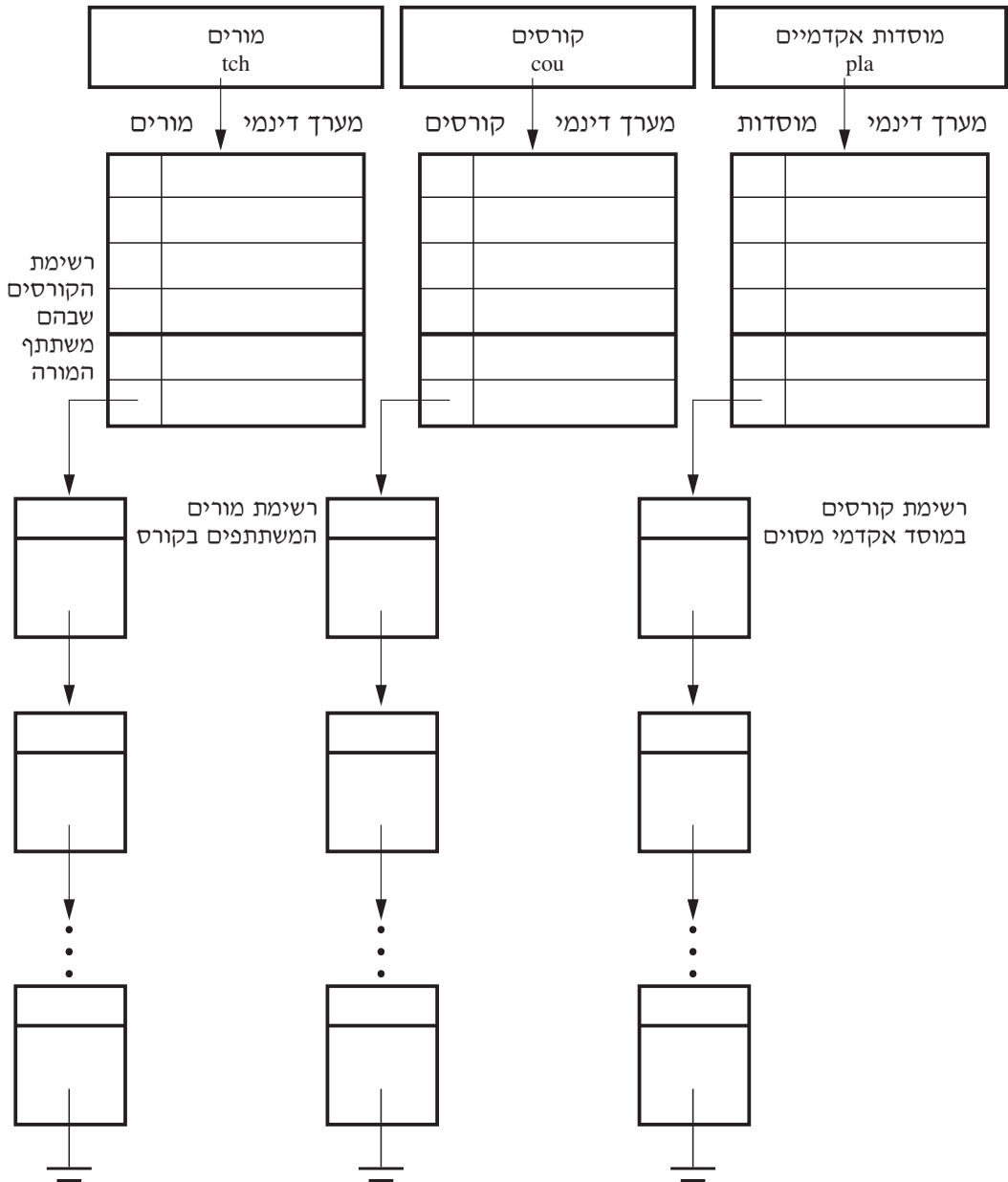
שאלה 1 – שאלת חובה (25 נקודות)

משרד החינוך בנה מערכת ממוחשבת המספקת מידע על אודות המורים המשתתפים בקורסים המתקיימים במוסדות אקדמיים בארץ.

הנחות יסוד:

1. בכל מוסד אקדמי מתקיים קורס אחד או יותר בנושאים שונים.
2. **בנושא מסוים** מתקיים קורס אחד בלבד.
3. קורס בנושא מסוים מתקיים במוסד אקדמי אחד בלבד.
4. מורה רשאי להירשם לקורס אחד או יותר במוסד אקדמי אחד או בכמה מוסדות אקדמיים.
5. במערכת הממוחשבת יהיו רשומים רק המורים שמשתתפים בקורס אחד לפחות.
6. מורה שמבטל את השתתפותו בכל הקורסים שבהם הוא רשום, יימחק מן המערכת הממוחשבת.
7. מוסד אקדמי רשאי לבטל קורס מסוים המועבר במוסד זה.
8. כל קורס מתקיים פעם אחת בלבד בשבוע.

לפניך תיאור סכמתי של **מבנה הנתונים** אשר יתמודד במימוש הפעולות הנדרשות מן המערכת הממוחשבת לניהול קורסי המורים.



איור לשאלה 1

נחזיק מבנה (רשומה) שנכנה אותו בשם "המבנה הראשי", המכיל את השדות האלה:

שדה 1 - tech – מערך דינמי של כל המורים המשתתפים בקורסים. כל תא במערך הזה מייצג מורה ומכיל את פרטיו האישיים, ובין היתר – מספר הזהות שלו (ID). מערך זה ממוין בסדר עולה, על-פי המפתח שהוא מספר הזהות של המורה. נוסף על כך, כל תא במערך זה, המייצג מורה מסוים – x , מכיל מצביע לראש רשימה מקושרת חד-כיוונית, כאשר כל צומת ברשימה זו מייצג קורס שבו משתלם המורה x .

שדה 2 - cou – מערך דינמי של כל הקורסים. כל תא במערך זה מייצג קורס ומכיל את פרטי הקורס ובין היתר את מספר הקורס. מערך זה ממוין בסדר עולה, על-פי מפתח שהוא מספר הקורס. נוסף על כך, כל תא במערך הזה, המייצג קורס מסוים – y , מכיל מצביע לראש רשימה מקושרת חד-כיוונית, כאשר כל צומת ברשימה זו מייצג מורה המשתתף בקורס y .

שדה 3 - pla – מערך דינמי של מוסדות אקדמיים (מערך זה אינו ממוין). כל תא במערך זה מייצג מוסד אקדמי ומכיל את פרטי המוסד האקדמי הזה. נוסף על כך, כל תא במערך זה, המייצג מוסד אקדמי מסוים – M , מכיל מצביע לראש רשימה מקושרת חד-כיוונית, כאשר כל צומת ברשימה זו מייצג קורס המועבר במוסד M .

המערכת הממוחשבת תתמוך, בין היתר, בפעולות שלהלן:

1. **אתחול מערכת** – `init` – הפעולה מאתחלת את המערכת הממוחשבת. הנח שפעולה זו מתבצעת פעם אחת בלבד.
2. **הוספת קורס חדש למוסד** – `addCourse` – פעולה זו מקבלת את פרטיו של קורס חדש – x ואת פרטי מוסד אקדמי מסוים – y , שבו אמור להתקיים הקורס הזה. אם הקורס x קיים במערכת הממוחשבת, אזי הפעולה מחזירה הודעה מתאימה – אחרת:
 - היא מוסיפה את קורס x למערך הקורסים, כך שלאחר הכנסתו, מערך הקורסים יהיה ממוין.
 - אם המוסד y אינו קיים במערכת הממוחשבת, אז הפעולה מוסיפה את המוסד הזה למערך המוסדות האקדמיים.
 - כמו כן, הפעולה מוסיפה את הקורס x לרשימת הקורסים המועברים במוסד האקדמי y .
3. **רישום מורה לקורס** – `registerTeacher` – פעולה זו מקבלת את פרטי המורה z ואת מספר הקורס x שאליו המורה נרשם.
אם המורה z אינו קיים במערכת הממוחשבת, אזי הפעולה מוסיפה את המורה הזו למערך המורים כך שלאחר הכנסתו של המורה z , מערך המורים יישאר ממוין.
נוסף על כך, הפעולה מוסיפה את הקורס x לרשימת הקורסים שבהם משתתף המורה z .
כמו כן, הפעולה מוסיפה את המורה z לרשימת המורים שמשותפים בקורס x .
4. **ביטול קורס** – `abolishCourse` – פעולה זו מקבלת את מספר הקורס ומבטלת אותו מן המערכת הממוחשבת.
5. **ביטול רישום של מורה בקורס מסוים** – `cancelRegistration` – פעולה זו מקבלת את מספר הזהות של מורה ומספר קורס מסוים, ומבטלת את ההשתתפות של המורה בקורס הזה.
6. **הצגת הקורסים שבהם משתתף מורה מסוים** – `printTeacherParticipation` – פעולה זו מקבלת את מספר הזהות של מורה, ומציגה את פרטי כל הקורסים שבהם משתתף מורה זה.
7. **הצגה של רשימת המורים המשתתפים בקורס מסוים** – `printCourseParticipants` – פעולה זו מקבלת מספר קורס ומציגה את שמות כל המורים המשתתפים בקורס זה.
8. **הצגה של רשימת הקורסים המתקיימים במוסד אקדמי** – `printInstitutesCourses` – פעולה זו מקבלת את מספרו של מוסד אקדמי, ומציגה את פרטי כל הקורסים המתקיימים בו.
9. **מידע על קורס מסוים** – `printCourseInfo` – פעולה זו מקבלת את מספר הקורס, ומציגה את כל פרטיו. נוסף על כך, הפעולה מציגה את הכתובת ואת שם העיר שבה מתקיים הקורס הזה.
10. **יציאה מן מערכת** – `quit` – הפעולה מסיימת את התכנית.

להלן הגדרת המבנה הראשי בשפת C :

```
typedef struct headType
{
    teacherPtr tch; // מערך דינמי של מורים
    coursePtr cou; // מערך דינמי של קורסים
    placePtr pla; // מערך דינמי של מוסדות אקדמיים
}header, *headPtr;
```

עתה נפרט את מבנה הנתונים בעבור השדה 1 של "המבנה הראשי".

להלן מבנה של תא במערך דינמי של מורים בשפת C :

```
typedef struct teacherType
{
    long teacherID; // מספר זהות מורה
    char lastName[20]; // שם משפחה
    char firstName[15]; // שם פרטי
    char tel[10]; // מספר טלפון
    struct courseListType *rgs; // מצביע לראש רשימה חד-כיוונית מקושרת המייצגת
    // את כל הקורסים שבהם השתתף מורה זה.
}teacherRec, *teacherPtr;
```

להלן המבנה של צומת ברשימת הקורסים של מורה מסוים:

```
typedef struct coursesListType
{
    int crs; // מספר קורס
    struct coursesListType *next; // שדה קישור
}coursesListRec, *coursesListPtr;
```

הערה: במערכת הממוחשבת קיימת רשימה נפרדת לכל מורה.

עתה נפרט את מבנה הנתונים בעבור **השדה 2** של "**המבנה הראשי**".

להלן מבנה של תא **במערך דינמי של קורסים** בשפת C :

```
typedef struct courseType
{
    int courseNo; // מספר הקורס
    char title[20]; // שם הקורס
    char lecturer[20]; // שם המרצה
    int totalHours; // מספר השעות הכולל של קורס זה
    int credit; // מספר נקודות הזכות שמקנה קורס זה
    char startDate[11]; // תאריך התחלת הקורס
    char endDate[11]; // תאריך סיום הקורס
    char dayOfWeek[4]; // היום בשבוע שבו מתקיים הקורס
    int instituteIndex; // מיקום (אינדקס) במערך המוסדות של מוסד, שבו מועבר
    // הקורס הזה
    participationRec *prt; // מצביע לראש רשימה חד-כיוונית
    // מקושרת של מורים המשתתפים בקורס זה
} courseRec, *coursePtr;
```

להלן מבנה של צומת ברשימה חד-כיוונית מקושרת של מורים המשתתפים בקורס מסוים:

```
typedef struct participationType
{
    long tch; // מספר זהות מורה
    struct participationType *next; // שדה קישור
} participationRec , *participationPtr;
```

הערה: במערכת הממוחשבת קיימת רשימה נפרדת לכל קורס.

עתה נפרט את מבנה הנתונים בעבור השדה 3 של "המבנה הראשי".

להלן מבנה של תא במערך דינמי מוסדות אקדמיים בשפת C :

```
typedef struct placeType
{
    int instituteNo; // מספר המוסד האקדמי
    char address[40]; // כתובת המוסד האקדמי
    char city[20]; // שם עיר
    struct coursesListType *scr; // מצביע לראש רשימה חד-כיוונית מקושרת
    // המכילה את כל קורסי ההשתלמות המתקיימים במוסד זה
} placeRec ,*placePtr;
```

שים לב שהמבנה של צומת ברשימת קורסים במוסד אקדמי מסוים זהה למבנה של צומת ברשימת הקורסים שבהם משתתף המורה.

הערה: לכל מוסד אקדמי קיימת במערכת הממוחשבת רשימה נפרדת.

נוסף על **המבנה הראשי** נחזיק במערכת הממוחשבת מערך מוסדות הכולל בעבור כל מוסד: שם מוסד ומספר מוסד.

```
#define NUM_OF_INSTITUTES 10
struct institute institutes[NUM_OF_INSTITUTES]; // טבלת מוסדות אקדמיים
```

להלן מבנה של תא בטבלת מוסדות אקדמיים בשפת C :

```
struct institute
{
    int instituteNo; // מספר המוסד האקדמי
    char instituteName[20]; // שם המוסד האקדמי
} institutein;
```


ההגדרות שלהלן תקפות עבור כל הסעיפים הבאים:

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

הנחות יסוד נוספות:

- בכל פעם שמוסיפים או שמוחקים מורה מסוים או קורס מסוים, מערך המורים או מערך הקורסים ימוין מחדש.
- אם קורס מסוים מתבטל, אזי יתעדכנו הן רשימות הקורסים בעבור המורים שאמורים להשתתף בקורס הזה והן רשימת הקורסים של המוסד שבו מתקיים הקורס.

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

פונקציה זו מקבלת <code>rp</code> , שהוא מצביע למערך הקורסים, ואת <code>key</code> , שהוא מספר קורס מסוים. אם הקורס שמספרו <code>key</code> קיים במערך הקורסים, אזי הפונקציה מחזירה את מיקומו במערך - אחרת, הפונקציה מחזירה את הערך <code>-1</code> .	<code>int findCourse(coursePtr rp, int key);</code>
פונקציה זו מקבלת <code>tp</code> , שהוא מצביע למערך המורים, ואת <code>key</code> , שהוא מספר הזהות של מורה מסוים. אם המורה בעל מספר הזהות <code>key</code> קיים במערך המורים, אזי הפונקציה מחזירה את מיקומו במערך - אחרת, הפונקציה מחזירה את הערך <code>-1</code> .	<code>int findTeacher (teacherPtr tp, long key);</code>
פונקציה זו מקבלת <code>p</code> , שהוא מצביע לראש רשימת הקורסים של מורה מסוים, ומוחקת מן הרשימה את הצומת שמייצג קורס שמספרו <code>courseNo</code> , ומחזירה מצביע לראש רשימת הקורסים.	<code>coursesListPtr deleteFromCoursesList (coursesListPtr *p, int courseNo);</code>
פונקציה זו מקבלת <code>p</code> , שהוא מצביע לראש רשימת הקורסים במוסד אקדמי מסוים, ומוחקת מן הרשימה את הצומת שמייצג קורס שמספרו <code>ID</code> , ומחזירה מצביע לראש רשימת קורסים המועברים במוסד אקדמי זה.	<code>participationPtr deleteFromParticipationList (participationPtr *p, long ID);</code>
פונקציה זו מקבלת <code>t</code> , שהוא מצביע למערך הקורסים, ומוחקת מן המערך הזה את הקורס שפרטיו מיוצגים על-ידי המבנה שמצביע עליו <code>p</code> , ומעדכנת את מספר הקורסים במערכת.	<code>void deleteFromCoursesArr (coursePtr *t, coursePtr p);</code>
פונקציה זו מקבלת <code>t</code> , שהוא מצביע למערך המורים, ומוחקת מן המערך הזה את המורה שפרטיו מיוצגים על-ידי המבנה שמצביע עליו <code>p</code> , ומעדכנת את מספר המורים במערכת.	<code>void deleteFromTeachersArr (teacherPtr *t, teacherPtr p)</code>

הנח שהפונקציות האלו כתובות, וכי ניתן להשתמש בהן בכל הסעיפים הבאים, ואין צורך לכתוב אותן.

להלן הגדרות של משתנים גלובאליים:

```
header head;  
headPtr DS = &head;  
int numOfCourses=0;      // מספר הקורסים במערכת הממוחשבת  
int numOfTeachers=0;     // מספר המורים במערכת הממוחשבת  
int numOfPlaces=0;       // מספר המוסדות האקדמיים במערכת הממוחשבת שבהם מתקיימים קורסים
```

א. לפניך פונקציה שכותרתה:

```
statusType insertCourses(coursePtr *t, coursePtr p)
```

פונקציה זו מקבלת מצביע t למערך הקורסים ומצביע p למבנה של קורס מסוים.

אם הקורס אינו קיים במערך הקורסים, אזי הפונקציה מבצעת את אלה:

- מרחיבה את המערך באופן הבא: מקצה שטח נוסף כך שהשטח הזה מצורף לסוף המערך המוצבע על-ידי t .

- מאתחלת את כל שדותיו של השטח הזה (הכניסה האחרונה של המערך המורחב) פרט לשדה `instituteIndex`.

- מעדכנת את מיספר הקורסים.

נוסף על כך, הפונקציה ממיינת את המערך שמוצבע על-ידי t באמצעות הפונקציה `qsort` אשר משתמשת בפונקציה `intcmp` הבאה:

```
int intcmp(const void *p, const void *q)  
{  
    coursePtr t1,t2;  
    int n1,n2;  
    t1 = p;  
    t2=q;  
    n1 = t1->courseNo;  
    n2 = t2->courseNo;  
    return n1-n2;  
}
```

הפונקציה מחזירה את הערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

ILLEGAL_COURSE	אם הקורס שעליו מצביע <code>p</code> כבר קיים במערכת הממוחשבת.
SUCCESS	אם הקורס שעליו מצביע <code>p</code> לא קיים במערכת הממוחשבת.

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 insertCourses(coursePtr *t, coursePtr p)
{
    int i;
    statusType status=SUCCESS;
    i = _____ (1) _____;
    if (i== -1)
    {
        *t = _____ (2) _____;
        (*t)[numOfCourses].courseNo = p->courseNo;
        strcpy((*t)[numOfCourses].title, p->title);
        strcpy((*t)[numOfCourses].lecturer, p->lecturer);
        strcpy((*t)[numOfCourses].startDate, p->startDate);
        strcpy((*t)[numOfCourses].endDate, p->endDate);
        strcpy((*t)[numOfCourses].dayOfWeek, p->dayOfWeek);
        (*t)[numOfCourses].totalHours = p->totalHours;
        (*t)[numOfCourses].credit = p->credit;
        (*t)[numOfCourses].prt = NULL;
        _____ (3) _____;
        _____ (4) _____;
    }
    else status = ILLEGAL_COURSE;
    return status;
}
```

ב. לפניך פונקציה שכותרתה:

```
statusType insertPlace(placePtr *t, placePtr c, int cr)
```

פונקציה זו מקבלת t – שהוא מצביע למערך של מוסדות האקדמיים, c – מצביע למבנה מוסד מסוים ו- cr – שהוא מספר קורס שכבר קיים במערכת הממוחשבת ואמור להתקיים במוסד הזה.

אם המוסד אינו קיים במערך מוסדות, אזי הפונקציה מצרפת אותו לסוף המערך מוסדות. הפונקציה מוסיפה את הקורס שמספרו cr לראש רשימת הקורסים המועברים על-ידי המוסד הזה.

תזכורת:

מערך המוסדות אינו ממוין בהכרח.

פונקציה זו מחזירה תמיד את הערך SUCCESS מטיפוס statusType המציין שהפונקציה התבצעה כנדרש.

בפונקציה 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.

```
statusType insertPlace(placePtr *t,placePtr c,int cr)
{
    int i;
    statusType status=SUCCESS;
    coursesListPtr inst,r;
    inst = malloc(sizeof(coursesListRec));
    inst->crs = cr;
    for(i=0;i < numOfPlaces;i++)
    {
        if(_____ (1) _____) break;
    }
    if (_____ (2) _____)
    {
        *t = realloc(*t,(numOfPlaces+1)*sizeof(placeRec));
        (*t)[numOfPlaces].instituteNo = c->instituteNo;
        strcpy((*t)[numOfPlaces].address , c->address);
        strcpy((*t)[numOfPlaces].city , c->city);
        (*t)[numOfPlaces].scr =_____ (3) _____;
        _____ (4) _____;
    }
    r =  (*t)[i].scr;
    _____ (5) _____ = inst;
    _____ (6) _____ = r ;
    return status ;
}
```

ג. לפניך פונקציה שכותרתה:

```
statusType addCourse(coursePtr crs,placePtr pls)
```

פונקציה זו מקבלת `crs`, שהוא מצביע למבנה קורס מסוים, ו-`pls`, שהוא מצביע למוסד אקדמי מסוים.

הפונקציה הזאת מבצעת את הפעולות שלהלן:

- מוסיפה את הקורס שעליו מצביע `crs` למערך הקורסים.
- אם המוסד שעליו מצביע `pls` אינו קיים, אזי הפונקציה מוסיפה אותו למערך המוסדות.
- מצרפת את הקורס שעליו מצביע `crs` לרשימת הקורסים המועברים במוסד האקדמי שעליו מצביע `pls`.

הפונקציה מחזירה את הערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

אם הקורס שעליו מצביע <code>crs</code> אינו חוקי או כבר קיים במערכת הממוחשבת.	ILLEGAL_COURSE
אם הקורס שעליו מצביע <code>crs</code> אינו קיים במערכת הממוחשבת.	SUCCESS

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.

```
statusType addCourse(coursePtr crs,placePtr pls)
{
    int courseNo;
    statusType status = SUCCESS;
    courseNo = crs->courseNo;
    if (courseNo <=0) return ILLEGAL_COURSE;
    status = insentCourses (_____1_____);
    if(status==SUCCESS)
    {
        status =_____ (2) _____;
        if(status==SUCCESS) crs->instituteIndex = numOfPlaces - 1;
    }
    else status = ILLEGAL_COURSE;
    return status;
}
```


4. לפניך פונקציה שכותרתה:

```
statusType registerTeacher(long teacherID,char lastName[],  
char firstName[],char tel[],int courseNo)
```

פונקציה זו מקבלת את פרטיו של מורה מסוים שנרשם לקורס שמספרו courseNo .

אם המורה הזה אינו קיים במערכת הממוחשבת, אזי הפונקציה מוסיפה אותו למערך המורים.

כמו כן, היא מבצעת את הפעולות שלהלן:

- ממיינת את מערך המורים באמצעות הפונקציה qsort אשר משתמשת בפונקציה **longcmp** הבאה:

```
int longcmp(const void *p,const void *q)  
{  
    teacherPtr t1,t2;  
    long n1,n2;  
    t1 = p;  
    t2=q;  
    n1 = t1->teacherID;  
    n2 = t2->teacherID;  
    return n1-n2;  
}
```

- מעדכנת את מספר המורים במערכת הממוחשבת.
 - מוסיפה את הקורס שמספרו courseNo לראש רשימת הקורסים שבהם משתתף מורה זה.
 - מוסיפה את המורה הזה לרשימת המורים המשתתפים בקורס שמספרו courseNo .
- הנח כי המורה אינו רשום לקורס.

פונקציה זו מחזירה תמיד את הערך SUCCESS מטיפוס statusType המציין שהפונקציה התבצעה כנדרש.

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 registerTeacher
(long teacherID,char lastName[],char firstName[],char tel[],
int courseNo)
{
    statusType status = SUCCESS;
    int pindx,qindx;
    teacherPtr p;
    coursePtr q;
    coursesListPtr cl;
    participationPtr pr;

    pindx = _____ (1) _____;
    if (pindx == -1) //New teacher
    {
        DS->tch = _____ (2) _____;
        DS->tch[numOfTeachers].teacherID = teacherID;
        strcpy(DS->tch[numOfTeachers].lastName,lastName);
        strcpy(DS->tch[numOfTeachers].firstName,firstName);
        strcpy(DS->tch[numOfTeachers].tel,tel);
        DS->tch[numOfTeachers].rgs=NULL;
        numOfTeachers++;
        qsort(DS->tch,numOfTeachers,sizeof(teacherRec),longcmp);
        pindx = _____ (3) _____;
    }
}
```

```
p = DS->tch + pindx;
qindx = findCourse(DS->cou, courseNo);
q = DS->cou + qindx;
cl= malloc(sizeof(coursesListRec));
cl->crs = courseNo ;
cl->next=_____ (4) _____;
_____ (4) _____ = cl;
pr = malloc(sizeof(participationRec));
pr->tch = teacherID;
pr->next = _____ (5) _____;
_____ (5) _____ = pr;
return status;
}
```

ה. לפניך פונקציה שכותרתה:

```
statusType cancelRegistration(long ID,int courseNo)
```

פונקציה זו מקבלת את ID, שהוא מספר הזהות של מורה, ו-courseNo, שהוא מספר קורס מסוים.

פונקציה זו מבצעת את הפעולות שלהלן:

- מוחקת מרשימת הקורסים שבהם משתתף המורה שמספר הזהות שלו ID את הקורס שמספרו courseNo.
- מוחקת מרשימת המורים המשתתפים בקורס שמספרו courseNo את המורה שמספר הזהות שלו ID.

הערה: אם לאחר מחיקת הקורס courseNo רשימת הקורסים של המורה שמספרו ID ריקה, אזי הפונקציה מוחקת את המורה הזה ממערך המורים וממיינת את מערך המורים.

הפונקציה מחזירה את הערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

COURSE_NOT_EXIST	אם לא קיים קורס שמספרו <code>courseNo</code> .
SUCCESS	אם קורס שמספרו <code>courseNo</code> התקיים כנדרש.
INVALID_INPUT	אם המורה לא נמצא

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 cancelRegistration(long ID,int courseNo)`

```
{
    statusType status = SUCCESS;
    teacherPtr p;
    coursePtr q;
    int pindx,qindx;
    pindx = findTeacher(DS->tch,ID);
    if(pindx == -1) return INVALID_INPUT;
    p = _____ (1) _____;
    qindx = findCourse(DS->cou,courseNo);
    if(qindx == -1) return COURSE_NOT_EXIST;
    q = DS->cou + qindx;
    q->prt = _____ (2) _____;
    p->rgs = _____ (3) _____;
    if(p->rgs==NULL)
        _____ (4) _____;
    return status;
}
```

ו. לפניך פונקציה שכותרתה:

```
statusType abolishCourse(int courseNo)
```

פונקציה זו מקבלת את `courseNo` - שהוא מספר קורס מסוים.

פונקציה זו מבטלת קורס לפי השלבים הבאים:

- מוחקת את הקורס שמספרו `courseNo` מרשימת הקורסים המועברים במוסד שבו מתקיים קורס זה.
- מוחקת בעבור כל מורה `x` המשתתף בקורס `courseNo` את הקורס הזה מרשימת הקורסים של מורה `x`.

הערה: אם לאחר מחיקת הקורס `courseNo`, רשימת הקורסים של המורה שמספרו ID ריקה, אזי הפונקציה מוחקת את המורה הזה ממערך המורים וממיינת את מערך המורים.

- גורמת לכך שרשימת המורים המשתתפים בקורס שמספרו `courseNo` תהיה רשימה ריקה.
- נוסף על כך הפונקציה מוחקת את הקורס שמספרו `courseNo` ממערך הקורסים.

הפונקציה מחזירה את הערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

COURSE_NOT_EXIST	אם לא קיים קורס שמספרו <code>courseNo</code> .
SUCCESS	אם קורס שמספרו <code>courseNo</code> בוטל כנדרש.

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 abolishCourse(int courseNo)
{
    statusType status = SUCCESS;
    placePtr pl;
    coursePtr cp;
    participationPtr pr;
    int index, cindx, tindx;
    teacherPtr tp;
    cindx = findCourse(DS->cou, courseNo);
    cp = DS->cou + cindx;
    index = (_____(1)_____);
    pl=DS->pla + index;
    pl->scr = deleteFromCoursesList_____ (2) _____;
    pr = cp->prr;
    tindx = findTeacher(_____ (3) _____);
    while(pr && (tindx != -1))
    {
        tp = DS->tch + tindx;
        status = _____ (4) _____;
        pr = pr->next;
        if(pr) tindx = findTeacher(DS->tch, pr->tch);
    }
    deleteFromCoursesArr (&(DS->cou), cp);
    return status;
}
```

שאלה 2 – שאלת חובה (10 נקודות).

לפניך תכנית בשפת C. הנח כי טיפוס int מיוצג באמצעות ארבעה בתים, טיפוס char מיוצג באמצעות בית אחד וכתובת של משתנה מיוצגת באמצעות ארבעה בתים.
רשום במחברתך את הפלט המדויק של התכנית שלפניך.

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

void f1(char *ptr1, char *ptr2);
void f2(char *ptr1, char *ptr2);

struct St1
{
    char* ptr;
};

struct St2
{
    char ptr[6];
    struct St1 *pSt1;
};

int main()
{
    int i;
    char cArr[3][7] = {"tzipi", "yuval", "eti"};
    struct St1 st1Arr[3] = {cArr[2], *(cArr+1), &cArr[0][1]};
    struct St1* pSt1 = st1Arr;
    struct St2 st2Arr[3] = {"hila", pSt1+1}, {"ilay", ++pSt1},
    {"aania", pSt1++}};
    struct St2 *pSt2 = st2Arr;
    printf("size = %d\n", sizeof(cArr));
```

```

    printf("size = %d\n", sizeof(pSt1));
    printf("size = %d\n", sizeof(st2Arr));
    printf("%s\n", (*pSt1++).ptr);
    printf("%s\n", (pSt2++)->ptr);
    printf("%s\n", (*pSt2).pSt1->ptr);
    pSt2=st2Arr;
    f1(pSt2->ptr, (pSt2+2)->ptr);
    printf("%s\n", (*pSt2).ptr);
    printf("%s\n", (*(++pSt2+1)).ptr);
    pSt2 = st2Arr;
    f2(pSt2->ptr, (pSt2+2)->ptr);
    printf("%s\n", (*&pSt2[0].ptr));
    printf("%s\n", (*(pSt2+1)+1).ptr);
    return 0;
}

void f1(char *ptr1, char *ptr2)
{
    ptr1="K";
    ptr2 = "M";
}

void f2(char *ptr1, char *ptr2)
{
    *ptr1 = 'K';
    *ptr2 = 'M';
}

```


פרק שני (15 נקודות)

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

שאלה 3

להלן הגדרה חדשה:

חיפוש טרינארי הוא שיטת חיפוש אשר מנסה לאתר איבר x במערך ממין.

בעבור המערך הממין $A_0 \dots A_{n-1}$ השיטה הזאת מחלקת את איברי המערך לשלושה חלקים באופן הזה:

חלק 1 - יכיל את אברי המערך: $A_0 \dots A_{\frac{n}{3}-1}$.

חלק 2 - יכיל את אברי המערך: $A_{\frac{n}{3}} \dots A_{\frac{2n}{3}-1}$.

חלק 3 - יכיל את אברי המערך: $A_{\frac{2n}{3}} \dots A_{n-1}$.

כדי לאתר את האיבר x בשיטת **החיפוש הטרינארי**, נבצע את הצעדים האלה:

צעד ראשון - אם האיבר x שווה לאיבר $A[\frac{2n}{3}]$, אזי האיבר x נמצא ונחזיר את מיקומו.

צעד שני - אם האיבר x גדול מהאיבר $A[\frac{2n}{3}]$, אזי תהליך חיפוש האיבר x בשיטה הטרינארית

נמשך בחלק 3 של המערך.

צעד שלישי - אם האיבר x שווה לאיבר $A[\frac{n}{3}]$, אזי האיבר x נמצא ונחזיר את מיקומו.

צעד רביעי - אם האיבר x גדול מהאיבר $A[\frac{n}{3}]$, אזי תהליך חיפוש האיבר x בשיטה הטרינארית

נמשך בחלק 2 של המערך - אחרת, תהליך החיפוש של האיבר x בשיטה הטרינארית נמשך

בחלק 1 של המערך.

לפניך תכנית לחיפוש טרינארי שמשתמשת בפונקציה `findkey()`.

```
# define CAPACITY 9
typedefenum {FALSE,TRUE} boolean;
int findKey(int a[],int key ,int low,int high);
int main(void)
{
    int key,pos;
    int i;
    int a[CAPACITY] = {1,3,5,7,9,11,13,15,17};
    for(i=0;i < CAPACITY ; i++)
    {
        printf("\t%d",i);
    }
    printf("\na = ");
    for(i=0;i<CAPACITY;i++)
    {
        printf("\t%d",a[i]);
    }
    printf("\n Enter Key :");
    scanf("%d",&key);
    pos = findKey(a,key,0,CAPACITY-1);
    if (pos == -1) printf("\n %d Not Found",key);
    else printf("\n%d found in position : %d",key,pos);
    return 0;
}
```

לפניך פונקציה אשר מקבלת כפרמטרים את: המערך הממוין `a`, הערך `key`, תחום החיפוש `low` עד `high`, ומחזירה את מיקומו של האיבר, שערכו `key`, במערך בשיטת חיפוש טרינארי. אם האיבר לא נמצא, אזי הפונקציה תחזיר את הערך `-1`.

בפונקציה 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.

```

int findKey(int a[],int key ,int low,int high)
{
    int n,t,mid1,mid2;
    boolean found= FALSE;
    while (_____(1)_____)
    {
        if (high< low) _____(2)_____;
        t = (int)((high-low)/3);
        mid1 = low + t;
        mid2 = low+2*t;
        if (_____(3)_____)
        {
            n = mid2;
            found = TRUE;
            break;
        }
        if (key >a[mid2])
        {
            _____(4)_____;
            _____(5)_____;
        }
        if (key == a[mid1])
        {
            n = mid1;
            found = TRUE;
            break;
        }
        if (key > a[mid1])
        {
            low=_____(6)_____;
            high =_____(7)_____;
            _____(5)_____;
        }
        high =_____(8)_____;
    }
    return(found)? n:-1;
}

```

שאלה 4

נתונה התכנית הבאה בשפת C :

```
#include<stdio.h>

#define n 10

int main()
{
    int vec1[n]={1,8,8,3,4,6,2,7,8,10};
    int vec2[n]={8,8,8,5,4,5,7,6,9,10};
    int help_vec1[100]={0};
    int help_vec2[100]={0};
    int help_vec3[2*n]={0};
    int i,j;
    for(i=0;i<n;i++)
    {
        if (vec1[i]<1||vec1[i]>99)
        {
            printf ("illegal input vector 1");
            return 1;
        }
        if (vec2[i]<1||vec2[i]>99)
        { printf ("illegal input vector 1");
          return 1;
        }
        help_vec1[vec1[i]]++;
        help_vec2[vec2[i]]++;
    }
}
```

```
for(i=0,j=0;i<100; i++)
{
    if(((help_vec1[i]==help_vec1[i]+help_vec2[i]) ||
        (help_vec2[i]==help_vec2[i]+help_vec1[i])) &&
        (help_vec1[i]+help_vec2[i]>0))
        help_vec3[j++]=i;
}

for (i=0; i<j;i++)
    printf(" %d ",help_vec3[i]);

return 0;

}
```

א. עקוב אחר מהלך התכנית ורשום את הפלט המדויק המתקבל כתוצאה מהרצת התכנית.

ב. מה יהיה פלט התכנית אם נשנה את הגדרת המערכים `vec1` ו-`vec2` למערכות האלה:

```
int vec1[n]={15,11,14,7,6,5,4,3,2,1};
```

```
int vec2[n]={1,11,2,3,14,15,5,5,8,2};
```

ג. כתוב במחברתך את המספר שליד התשובה הנכונה. התכנית מדפיסה את:

1. שלושת האיברים הקטנים ביותר מבין האיברים שמופיעים במערך `vec1` ושאינם מופיעים במערך `vec2`.

2. האיברים שמופיעים במערך `vec2` ושאינם מופיעים במערך `vec1`.

3. האיברים שמופיעים במערך `vec1` ושמופיעים במערך `vec2`.

4. האיברים שמופיעים במערך `vec1` ושאינם מופיעים במערך `vec2`.

5. האיברים שמופיעים במערך `vec1` ושאינם מופיעים במערך `vec2` וגם את האיברים שמופיעים במערך `vec2` ושאינם מופיעים במערך `vec1`.

נושא ב': שפת סף (50 נקודות)

פרק שלישי (20 נקודות)

ענה על שאלה 5 – שאלת חובה.

שאלה 5

להלן תכנית אשר קולטת רצף של תווים כאשר כל תו ברצף הוא '0' או '1'. קריאת הקלט מסתיימת עם הקשת <enter>, או לאחר קליטת 16 תווים כאשר כל תו ברצף הוא '0' או '1'.

הערות:

- הנח שהקלט תקין.
- הקשת <enter> מחזירה לאוגר AL את הערך 13.

```
CR EQU 13
DATA SEGMENT
    NUM DW 0
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
        MOV CX,16
        MOV AH,1
NEXT: INT21H
        CMP AL,CR
        JE DISP
```

```
CMP AL, '1'
JE ONE
CLC
JMP GO
ONE:STC
GO: RCL NUM,1
    LOOP NEXT
DISP: XOR CX,CX
    MOV AX,NUM
    MOV BX,10
AGAIN:XOR DX,DX

    DIV BX
    PUSH DX
    INC CX
    OR AX,AX
    JNZ AGAIN
    MOV AH,2 ;NEW LINE
    MOV DX,0A0DH
    INT 21H
    XCHG DL,DH
    INT 21H
N_DIGIT:
    POP DX
    ADD DL,'0'
    INT 21H
    LOOP N_DIGIT
SOF:MOV AH,4CH
    INT 21H
CODE ENDS
END START
```

- א. מה יהיה תוכנו (בבסיס עשרוני) של המשתנה NUM לאחר קליטת הרצף הזה (קרא משמאל לימין): `1001<enter>`.
- ב. מה יהיה תוכנו (בבסיס עשרוני) של המשתנה NUM לאחר קליטת הרצף הזה (קרא משמאל לימין): `11100001<enter>`.
- ג. להלן ארבעה היגדים שאחד מהם מתאר את הפעולה שמבצעת התכנית הנתונה. רשום במחברתך את מספרו של ההיגד הנכון.
- **היגד 1:** התכנית מחברת את ערכי התווים שנקלטו בקלט ומציבה את סכומם במשתנה NUM.
 - **היגד 2:** התכנית יוצרת, מרצף התווים שבקלט, מספר בינארי, מציבה את ערכו של המספר הבינארי הזה במשתנה NUM ומקרינה אותו על הצג בייצוג עשרוני.
 - **היגד 3:** התכנית מקרינה על הצג את רצף התווים שנקלטו, בסדר הפוך. למשל, לאחר קליטת הרצף הזה (קרא משמאל לימין): `1100<enter>` התכנית מקרינה על הצג את הרצף `0011`.
 - **היגד 4:** התכנית מקרינה על הצג את סך הכול מספר ה-'0'ים ומספר ה-'1'ים שנקלטו על-ידי התכנית עד להקשת `<enter>`, או לאחר קליטת 16 תווים, כאשר כל תו ברצף הוא '0' או '1'.
- ד. אם הקלט מכיל רצף של 16 '1'ים, ואם נחליף את השורה **DIV BX** בשורה **DIV BL**, אזי האם השינוי הזה יכול להשפיע על הערך שיוקראן על הצג? ענה "כן" או "לא".
- ה. אם נחליף את השורה **OR AX,AX** בשורה **CMP AX,0**, אזי האם שינוי זה יכול להשפיע על הערך שיוקראן על הצג? ענה "כן" או "לא".
- ו. אם נחליף את הפקודה **CLC** בפקודה **STC**, ונחליף את הפקודה **STC** בפקודה **CLC**, אזי מה תקרין התכנית על הצג בעבור הקלט `1001<enter>`?

פרק רביעי (30 נקודות)

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

שאלה 6

נתונות שתי רשימות מקושרות חד־כיווניות **לא ריקות** הבנויות מצמתים.

כל צומת ברשימות הללו מכיל את שני השדות האלה:

info – שדה מידע (אינפורמציה), שגודלו 8 ביטים, המכיל מספר שלם וחיובי (מספר חסר סימן).

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

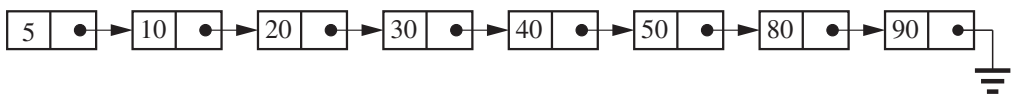
המצביע next בצומת האחרון הוא 0.

נוסף על כך, נתון כי המשתנה LIST1 הוא הכתובת של הצומת הראשון ברשימה הראשונה,

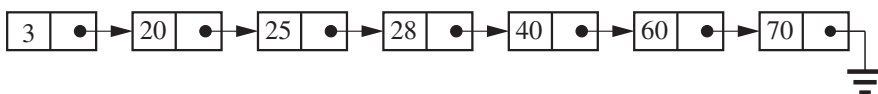
והמשתנה LIST2 הוא הכתובת של הצומת הראשון ברשימה השנייה (ראה איור).

כמו כן, הנח ששתי הרשימות האלה ממוינות בסדר עולה.

LIST1



LIST2

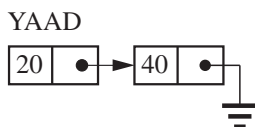


איור א' לשאלה 7

לפניך קטע של תכנית בשפת אסמבלי אשר סורקת את שתי הרשימות האלה ובונה **רשימה חדשה** אשר תכיל את האיברים שנמצאים ברשימה הראשונה וגם ברשימה השנייה.

המשתנה YAAD הוא הכתובת של הצומת הראשון **ברשימה החדשה** (ראה איור), בתנאי שהרשימה אינה ריקה; ולא – המשתנה YAAD יכיל את הערך 0.

בהמשך לאיור א', קטע התכנית ייצור את הרשימה החדשה הבאה:



איור ב' לשאלה 7

הערה: אם לשתי הרשימות אין איברים משותפים, אזי המשתנה YAAD יכיל את הערך 0.

נוסף על כך, הנח שקיימת שגרה בשם INSEND אשר מקבלת באמצעות מחסנית את:

- YAAD .

- NUM – מספר לא מסומן (חסר סימן).

השגרה יוצרת צומת חדש, מציבה בשדה ה- info של הצומת הזה את המספר NUM, ומוסיפה את הצומת הזה לסוף הרשימה החדשה YAAD .

בקטע התכנית הנתון חסרים **עשרה** ביטויים המסומנים במספרים בין סוגריים עגולים.

רשום במחברתך את מספרי הביטויים החסרים (1) – (10), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
LEA  BX , LIST1
LEA  DI  , LIST2

AGAIN:  MOV  AL , [BX]
        CMP  AL , [DI]
        JNE  CON
        ____ (1) ____
        ____ (2) ____
        ____ (3) ____

GO:     CMP  WORD PTR [BX+1] , 0
        JE   ____ (4) ____
        ____ (5) ____
        CMP  WORD PTR [DI+1] , 0
        JE   ____ (4) ____
        ____ (6) ____
        JMP  NEXT

CON:    ____ (7) ____  INC_DI
INC_BX:  CMP  WORD PTR [BX+1] , 0
        JE   EXIT
        MOV  BX , [BX+1]
        JMP  ____ (8) ____

INC_DI:  CMP  WORD PTR [DI+1] , 0
        JE   EXIT
        ____ (9) ____

NEXT:   ____ (10) ____

EXIT:
```

שאלה 7

לפניך תכנית בשפת אסמבלי הכוללת שגרה **רקורסיבית** המקבלת מספר עשרוני שלם חיובי וגדול מאפס NUM באמצעות מחסנית. השגרה מחזירה באמצעות אוגר AL את ערך הספרה הגדולה ביותר שב־NUM.

הנחת יסוד: המספר שמשוכן במשתנה NUM ערכו 2,500 לכל היותר.

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.

התכנית נעזרת בשגרה רקורסיבית בשם REC.

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

CODE SEGMENT
    ASSUME CS:CODE
    NUM DW 2364
    START: PUSH NUM
           CALL REC
           MOV AH,4CH
           INT 21H
    REC:   PUSH BP
           MOV BP,SP
           MOV AX, _____ (1) _____
           _____ (2) _____
           DIV BL
           OR AL,AL
           _____ (3) _____
           MOV [BP+4],AH
           MOV BYTE PTR[BP+5],0
           XOR AH,AH
           _____ (4) _____
           _____ (5) _____
    STOP_REC:
           CMP AL,[BP+4]
           _____ (6) _____
           MOV AL,[BP+4]
    CON:   _____ (7) _____
           RET 2

CODE ENDS
END START
```

שאלה 8

לפניך תכנית בשפת אסמבלי:

```
DATA SEGMENT
    A DB 7,3,4,5,6,2
    LEN=$-A
    P DW A,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 P
          PUSH P+2
          CALL TROUBLE
SOF:      MOV AH,4CH
          INT 21H
TROUBLE : PUSH BP
          MOV BP,SP
CHAZOR:   MOV CX,[BP+4]
          DEC CX
          XOR SI,SI
          MOV BX,[BP+6]
AGAIN:    MOV AL,[BX]
          CMP AL,[BX+1]
          JBE CON
          XCHG AL,[BX+1]
```

```
MOV [BX], AL
INC SI
CON: INC BX
      LOOP AGAIN
      OR SI, SI
      JNE CHAZOR
      POP BP
      RET 4

CODE ENDS
END START
```

- א. רשום במחברת הבחינה את הערכים שישוכנו, לאחר הרצת התכנית, במערך A, החל באיבר הראשון וכלה באיבר האחרון.
- ב. השורה 2, 3, 4, 5, 6, 7 ADB מוחלפת בשורה 14, 10, 5 ADB. רשום במחברת הבחינה את הערכים שישוכנו, לאחר הרצת התכנית, במערך A, החל באיבר הראשון וכלה באיבר האחרון.
- ג. מה מבצעת התכנית? **ענה במשפט אחד בלבד.**

בהצלחה!