

סוג הבחינה: גמר לבתי-ספר לטכנאים ולהנדסאים

מועד הבחינה: אביב תשפ"א, 2021

סמל השאלון: 714001

נספחים: א. נספח לחלק ב': שפת סף

ב. מילון מונחים

תכנות מערכות בשפת C ושפת סף

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני חלקים – **חלק א': תכנות מערכות בשפת C, חלק ב': שפת סף.**

בשאלון שמונה שאלות בסך הכול, ועליך לענות על **שש** שאלות מביניהן,

על-פי ההנחיות.

סך-הכול – 100 נקודות.

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

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

בשאלון זה 35 עמודים ו-3 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר,

אך מכוונות הן לנבחנות והן לנבחנים.

בהצלחה!

השאלות

חלק א': תכנות מערכות בשפת C (60 נקודות)

הערה: בחלק זה ניתן להשתמש בכל פונקציות הספרייה הסטנדרטית של שפת C.

פרק ראשון (45 נקודות)

ענה על שתי השאלות 1-2.

שאלה 1 – שאלת חובה (35 נקודות)

ענה על שבעה מבין הסעיפים א'–ט' (לכל סעיף – 5 נקודות).

הוחלט להקים בעיר מגורכם מערך של השכרת אופניים כדי לשפר את בטיחותם ובריאותם של התושבים ולצמצם את זיהום האוויר.

ניהול מערך האופניים נעשה באמצעות יישומון (אפליקציה).

המערכת של היישומון הזה תתמודד, בין היתר, בפעולות שלהלן:

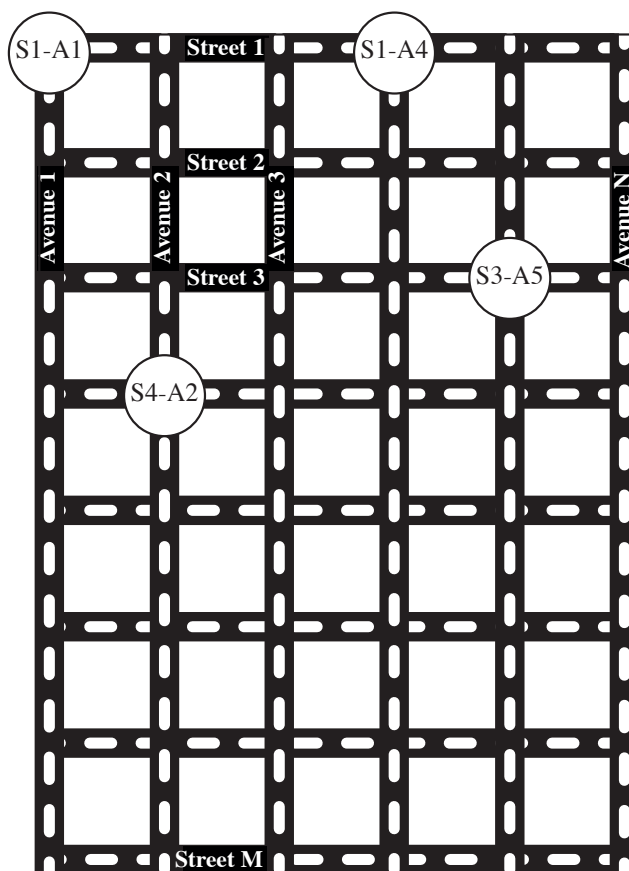
1. הוספת זוג אופניים חדש למאגר האופניים בעיר
2. השכרת זוג אופניים על-ידי הלקוח
3. החזרת זוג אופניים על-ידי הלקוח
4. הדפסת רשימה של כל האופניים הזמינים בתחנת השכרה כלשהי
5. הדפסת רשימה של כל האופניים המושכרים כעת
6. הדפסת רשימה ממוינת של מספרי הזהות של כל הלקוחות הפעילים כעת

הנחות יסוד:

1. העיר בנויה מרשת של רחובות (streets) הממוספרים מ-1 עד M, אשר מצטלבים עם רשת של שדרות (avenues) הממוספרות מ-1 עד N, ובכל הצטלבות ממוקמת תחנה להשכרת אופניים.
2. המרחק בין כל שני רחובות עוקבים וכן המרחק בין כל שתי שדרות עוקבות הוא זהה.
3. לקוח שוכר זוג אופניים מתחנת ההשכרה הקרובה ביותר למקום הימצאו, שיש בה זוג אופניים פנוי, ומחזיר אותם לתחנת ההשכרה הקרובה ביותר ליעד הנסיעה שלו.
4. לקוח יכול לשכור זוג אופניים אחד ברגע נתון.

5. לכל זוג אופניים יש מספר זיהוי ייחודי הנקבע על-ידי המערכת.
6. כל זוג אופניים חדש המוכנס למערכת משויך לתחנת ההשכרה הממוקמת בהצטלבות של רחוב 1 ושדרה 1.
7. מרגע שהלקוח מזמין זוג אופניים באמצעות היישומון, זוג האופניים נשמר עבורו, וחייב התשלום מתחיל.
8. ניתן להחזיר זוג אופניים לתחנה כלשהי ללא הגבלה.
9. בכל רגע נתון קיים לפחות זוג אופניים אחד פנוי באחת מן התחנות.
10. בכל סעיפי השאלה ניתן להניח שהקלט תקין.
11. אין צורך לבדוק את תקינותן של ההקצאות הדינמיות בתוכנית.
12. התעריף בעבור השכרת זוג אופניים הוא קבוע, ועומד על 5 שקלים לשעה מלאה, או לחלק ממנה.
13. המערכת לא שומרת מידע על הלקוחות, ולכן בכל השכרה יש להזין מחדש את פרטי הלקוח, גם אם הוא שכר אופניים בעבר.

להלן איור של מפת העיר:



איור א' לשאלה 1

לפניך תיאור טיפוסי הנתונים התומכים במימוש הפעולות הנדרשות מן המערכת הממוחשבת (הפעולות יפורטו בהמשך).

עתה נפרט את טיפוס הנתונים **bike**, המייצג זוג אופניים **ברשימה המקושרת** של האופניים **הזמינים** המשוויכים לתחנת

השכרה **ספציפית**. כמו כן, הוא מייצג זוג אופניים **ברשימה המקושרת הגלובלית** של האופניים **המושכרים כעת**.

הערה: זוג אופניים יכול להיות מושכר ללקוח **או** זמין להשכרה בכל רגע נתון.

להלן מבנה של צומת **ברשימה מקושרת של אופניים** בשפת C:

```
typedef struct bike
{
    unsigned long int bikeID;           // מספר הזיהוי של זוג האופניים
    struct bike *next;                  // שדה קישור
}bikeRec, *bikePtr;
```

עתה נפרט את טיפוס הנתונים **customer** המייצג **לקוח** ברשימה המקושרת **הגלובלית** של הלקוחות **הפעילים**.

לקוח מזוהה על-ידי **idNumber**, שהוא מספר הזהות של הלקוח.

להלן מבנה של צומת **ברשימה המקושרת של הלקוחות הפעילים** בשפת C:

```
typedef struct customer
{
    unsigned long int idNumber;         // מספר הזהות של הלקוח
    char *creditCardNumber;             // מספר כרטיס האשראי של הלקוח
    time_t begin;                       // זמן תחילת ההשכרה
    bikePtr yourBike;                   // מצביע לזוג האופניים שהושכרו ללקוח
    struct customer *next;              // שדה קישור
}customerRec, *customerPtr;
```

עתה נפרט את טיפוס הנתונים **map**, שהוא **מטריצה** המייצגת את **מפת העיר**:

שורות המטריצה **map** מייצגות את רחובות העיר, ועמודות המטריצה מייצגות את שדרות העיר.

כל תא במטריצה מייצג תחנת השכרה הממוקמת בהצטלבות של רחוב ושדרה, ומכיל מצביע **לרשימה מקושרת** של זוגות האופניים הזמינים הנמצאים כרגע בתחנת ההשכרה.

להלן מבנה המטריצה בשפת C:

```
typedef bikePtr map[M][N]; // מטריצה לייצוגן של הצטלבויות בין רחובות ושדרות העיר
```

עתה נפרט את טיפוס הנתונים **location**, המייצג **מיקום תחנת השכרה** – ההצטלבות של רחוב ושדרה.

להלן המבנה של **מיקום תחנת השכרה** בשפת C:

```
typedef struct location
{
    int street;           // מספר רחוב
    int avenue;           // מספר שדרה
}locationRec, *locationPtr;
```

שים לב:

מספור הרחובות והשדרות מתחיל מ-1, ואילו אינדקס השורות והעמודות במטריצה **map** מתחיל מ-0.

כלומר: התחנה הצפונית-מערבית ביותר היא $\text{location} = \{1,1\}$ והיא משוכנת במטריצה **map** באינדקס (0,0).

להלן הגדרות התקפות לכל הסעיפים שיבואו בהמשך:

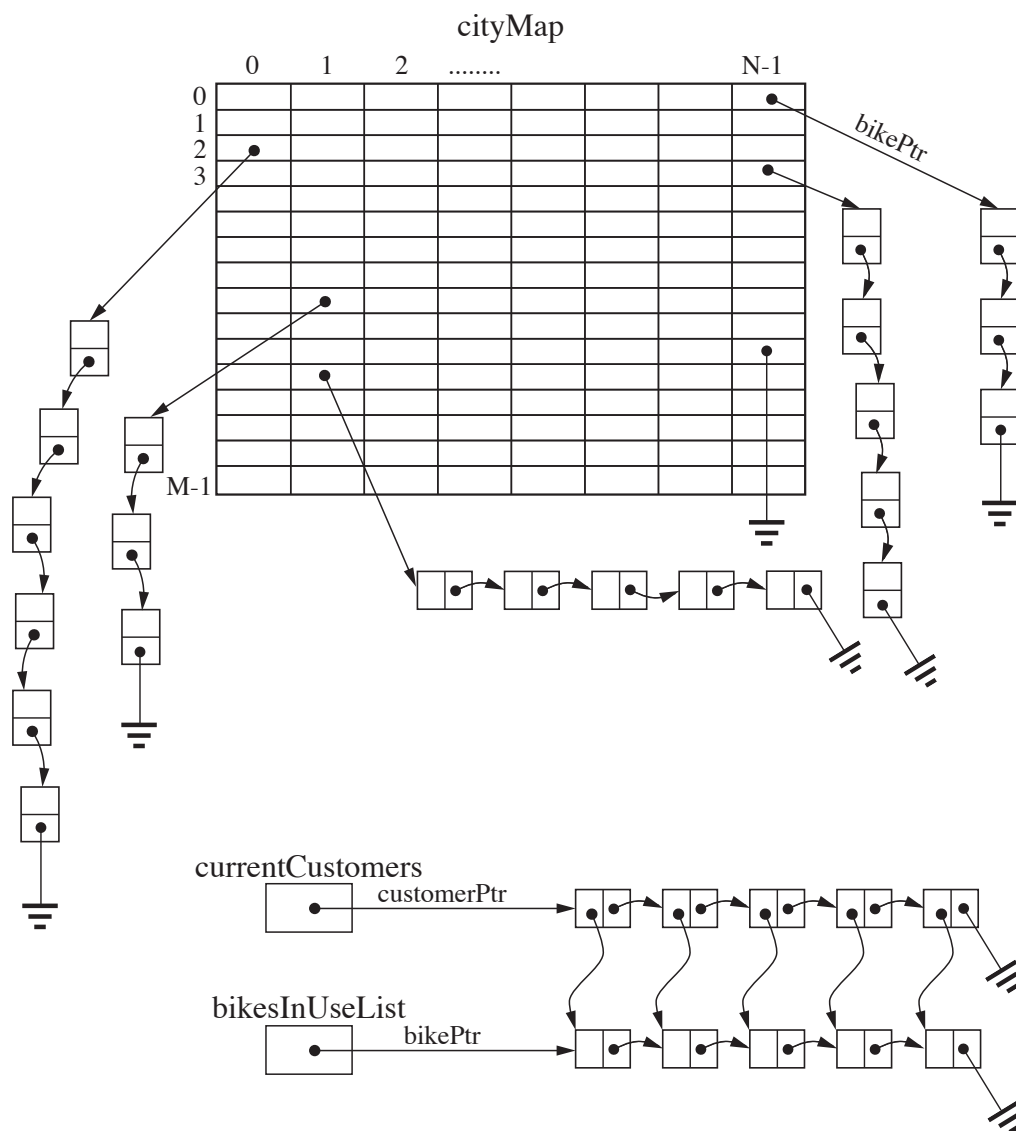
```
#define M 200 // מספר הרחובות בעיר
#define N 100 // מספר השדרות בעיר
#define MAX_DISTANCE M + N + 1 // מרחק מקסימלי אפשרי בין שתי הצטלבויות
#define SECONDS_IN_HOUR 360 // מספר השניות בשעה
#define HOUR_RATE 5 // תעריף לשעת השכרה

typedef enum Boolean {FALSE, TRUE} Status;
```

להלן הגדרות של משתנים גלובליים:

```
map cityMap = {NULL}; // מפת העיר
bikePtr bikesInUseList = NULL; // רשימת האופניים המושכרים כעת
customerPtr currentCustomers = NULL; // רשימת הלקוחות הפעילים
```

באיור שלהלן מתוארים מבני הנתונים התומכים בפעולות המערכת הממוחשבת:



איור ב' לשאלה 1

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

פונקצייה זו מבקשת מהלקוח להזין את מספר תעודת הזהות שלו, ומחזירה את הערך הזה.	<code>unsigned long int inputUserID(void);</code>
פונקצייה זו מבקשת מהלקוח להזין את מספר כרטיס האשראי שלו, ומחזירה אותו כמחרוזת.	<code>char *inputCreditCardNumber(void);</code>
פונקצייה זו מבקשת מהלקוח להזין את מיקומו הנוכחי, ומחזירה את המיקום כערך מטיפוס <code>locationPtr</code> .	<code>locationPtr inputCurrentLocation(void);</code>
פונקצייה זו מקבלת את <code>bike</code> , שהוא זוג אופניים, ואת <code>location</code> שהיא הצטלבות. הפונקצייה מוסיפה את זוג האופניים <code>bike</code> לרשימת האופניים הזמינים להשכרה בתחנה הנמצאת בהצטלבות <code>location</code> , ומחזירה מצביע לזוג האופניים הזה.	<code>bikePtr addBikeToIntersection(bikePtr bike, locationPtr location);</code>
פונקצייה זו מקבלת את <code>bike</code> , שהוא זוג אופניים. הפונקצייה מסירה את זוג האופניים מרשימת האופניים שמושכרים כעת, ומחזירה מצביע לזוג האופניים הזה.	<code>bikePtr deleteBikeFromUseList(bikePtr bike);</code>
הפונקצייה מקבלת את <code>customer</code> , שהוא לקוח. הפונקצייה מוסיפה את הלקוח הזה לרשימת הלקוחות הפעילים כעת.	<code>void addCustomer(customerPtr customer);</code>

הפונקצייה מקבלת את ID , שהוא מספר הזהות של לקוח. הפונקצייה מחפשת את הלקוח שמספר הזהות שלו הוא ID בשימת הלקוחות הפעילים כעת, ומחזירה מצביע ללקוח הזה.	<pre>customerPtr findCustomer(unsigned long int ID);</pre>
פונקצייה זו מקבלת את sum , שהוא סכום הכסף שיש לגבות מהלקוח, ואת הלקוח customer . הפונקצייה מבצעת את הגבייה.	<pre>int paymentCollection(int sum, customerPtr customer);</pre>
פונקצייה זו מקבלת את seconds , המציין את משך ההשכרה בשניות. הפונקצייה מחשבת ומחזירה את הסכום שהלקוח נדרש לשלם בעבור השכרת האופניים.	<pre>int calculatePayment(double seconds);</pre>

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

הערה: ניתן להשתמש בכל פונקציות הספרייה הסטנדרטית של שפת C .

להלן כמה פונקציות וטיפוסים השייכים לספרייה הסטנדרטית. תוכל להיעזר בהן בעת הצורך:

הקובץ time.h הוא חלק מן הספרייה הסטנדרטית של שפת C . בקובץ זה מוגדרים טיפוסים משתנים, פקודות מאקרו וחותמות של פונקציות שונות, המשמשים לניהול תאריכים וזמנים, כאשר אחד מטיפוסים המשתנים הוא: time_t .

time_t הוא משתנה מטיפוס שלם (long int) המשמש לאחסון מידע על זמן, והמידע המאוחסן בו הוא סך השניות שחלפו מאז 1.1.1970 בחצות ועד לנקודת הזמן הנוכחית.

להלן פונקציות נוספות המוגדרות בקובץ `time.h`:

פונקצייה זו מקבלת שני פרמטרים מטיפוס <code>time_t</code> : <code>time2</code> המכיל את הזמן הסופי, ו- <code>time1</code> המכיל את הזמן ההתחלתי. הפונקצייה מחזירה ערך מטיפוס <code>double</code> המייצג את <code>time2 - time1</code> , כלומר ההפרש בשניות בין הזמן הסופי (<code>time2</code>) ובין הזמן ההתחלתי (<code>time1</code>).	<code>double difftime(time_t time2, time_t time1);</code>
פונקצייה זו מקבלת כפרמטר את <code>NULL</code> . הפונקצייה מחזירה את ערך הזמן הנוכחי בפורמט <code>time_t</code> .	<code>time_t time(NULL);</code>

5 נק') א. לפניך פונקצייה שכותרתה:

```
bikePtr newBike(void)
```

פונקצייה זו מכניסה זוג אופניים חדש למאגר האופניים בעיר, ומחזירה מצביע לאופניים שהוכנסו.

הערה: זוג אופניים חדש מוצב תמיד בהצטלבות של רחוב 1 ושדרה 1.

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

```
bikePtr newBike(void)
```

```
{  
    static unsigned long int counter = 0;  
    locationRec intersectionID = {1,1};  
    bikePtr bike = _____(1)_____;  
    counter++;  
    bike->bikeID = _____(2)_____;  
    bike->next = NULL;  
    addBikeToIntersection(bike, _____(3)_____;  
    return bike;  
}
```

(5 נק') ב. להלן כותרת של פונקצייה:

```
bikePtr addBikeToInUseList(bikePtr bike)
```

פונקצייה זו מקבלת את bike, שהוא מצביע לזוג אופניים.

הפונקצייה מוסיפה את זוג האופניים שעליהן מצביע bike לרשימת האופניים המושכרים כעת, ומחזירה מצביע לזוג האופניים הזה.

כתוב במחברתך **מימוש מלא** לפונקצייה הזו, כך שסיבוכיות זמן הריצה שלה תהיה $O(1)$.

(5 נק') ג. נתונה פונקצייה שכותרתה:

```
locationPtr findNearestBike(locationRec currentLocation)
```

פונקצייה זו מקבלת את currentLocation, שהוא מצביע למיקומו הנוכחי של הלקוח.

הפונקצייה מחפשת את תחנת ההשכרה הקרובה ביותר למיקומו הנוכחי של הלקוח, שיש בה זוג אופניים זמין לשימוש, ומחזירה מצביע למיקום זה.

הנחות:

1. תמיד תימצא לפחות תחנה אחת שיש בה זוג אופניים זמין לשימוש.
2. אם יש כמה תחנות הנמצאות במרחק זהה למיקומו הנוכחי של הלקוח ועונות על הנדרש – הפונקצייה תחזיר מצביע לאחת מהן.

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

```

locationPtr findNearestBike(locationRec currentLocation)
{
    int distance[M][N];
    int streetIndex, avenueIndex, minStreet, minAvenue, min;
    minStreet = minAvenue = min = MAX_DISTANCE;
    locationPtr nearestLocation = (locationPtr)malloc(sizeof(locationRec));
    for (streetIndex = 0; streetIndex < M; streetIndex++)
        for (avenueIndex = 0; avenueIndex < N; avenueIndex++)
            distance[streetIndex][avenueIndex] = _____(1)_____;
    for (streetIndex = 0; streetIndex < M; streetIndex++)
        for (avenueIndex = 0; avenueIndex < N; avenueIndex++)
        {
            if (cityMap[streetIndex][avenueIndex] != _____(2)_____)
                distance[streetIndex][avenueIndex] =
                    _____(3)_____(currentLocation.street - 1 - streetIndex) +
                    _____(4)_____(_____ (5) _____);
        }
    for (streetIndex = 0; streetIndex < M; streetIndex++)
        for (avenueIndex = 0; avenueIndex < N; avenueIndex++)
            if (distance[streetIndex][avenueIndex] < min)
            {
                minStreet = streetIndex;
                minAvenue = avenueIndex;
                min = distance[streetIndex][avenueIndex];
            }
    nearestLocation->street = _____(6)_____;
    nearestLocation->avenue = _____(7)_____;
    return nearestLocation;
}

```

(5 נק') ד. לפניך פונקצייה שכותרתה:

```
bikePtr deleteBikeFromIntersection(locationPtr location)
```

פונקצייה זו מקבלת את location , המייצג תחנת השכרה (הצטלבות).

הפונקצייה מסירה זוג אופניים אחד מרשימת האופניים הזמינים להשכרה בתחנת ההשכרה location .

הפונקצייה מחזירה מצביע לאופניים שהוסרו.

אם רשימת האופניים הזמינים שממנה מעוניינים להסיר זוג אופניים היא ריקה, אזי הפונקצייה מחזירה את הערך NULL .

שים לב: פונקצייה זו אינה מטפלת בשיוך האופניים שהוסרו לרשימת האופניים המושכרים כעת. השיוך ייעשה באמצעות פונקצייה אחרת.

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

```
bikePtr deleteBikeFromIntersection(locationPtr location)
{
    bikePtr bike, bikeList;
    int streetIndex, avenueIndex;
    streetIndex = _____ (1) _____;
    avenueIndex = _____ (2) _____;
    bike = bikeList = cityMap[streetIndex][avenueIndex];
    if (_____ (3) _____)
        return NULL;
    cityMap[streetIndex][avenueIndex] = _____ (4) _____;
    bike->next = NULL;
    return bike;
}
```

(5 נק') ה. לפניך פונקצייה שכותרתה:

bikePtr rentBike(void)

פונקצייה זו מבקשת מן הלקוח להזין את פרטיו האישיים, את מיקומו הנוכחי ואת פרטי כרטיס האשראי שלו. הפונקצייה מאתרת את זוג האופניים הזמין הקרוב ביותר למיקומו הנוכחי של הלקוח ו"משכירה" לו אותו. לסיום, הפונקצייה שומרת במערכת הממוחשבת את המידע על הלקוח ועל ההשכרה, ומציינת כי זוג האופניים אינו זמין עוד להשכרה.

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

bikePtr rentBike(void)

```
{
    locationPtr currentLocation, nearestLocation;
    bikePtr bike;
    int flag = 0;
    customerPtr currentCustomer = (customerPtr)malloc(sizeof(customerRec));
    currentCustomer->idNumber = inputUserID();
    while (!flag)
    {
        currentCustomer->creditCardNumber = inputCreditCardNumber();
        if (checkCreditCardNumber(_____(1)_____(1)))
            flag = TRUE;
        else
            printf("Wrong credit card number");
    }
    time(_____(2)_____(2));
    currentCustomer->next = NULL;
    currentLocation = _____(3)_____(3);
    _____(4)_____(4) = findNearestBike(*currentLocation);
    bike = _____(5)_____(5) (nearestLocation);
    currentCustomer->yourBike = bike;
    bike = addBikeToInUseList(_____(6)_____(6));
    addCustomer(_____(7)_____(7));
    return bike;
}
```

5 נק') 1. לפניך פונקצייה שכותרתה:

```
void deleteCustomer(customerPtr customer)
```

פונקצייה זו מקבלת את customer, שהוא מצביע ללקוח ברשימת הלקוחות הפעילים כעת.

הפונקצייה מוחקת מן המערכת את כל פרטי הלקוח ואת ההשכרה.

שים לב: פונקצייה זו אינה מטפלת בהחזרת האופניים בתום ההשכרה. ההחזרה תיעשה באמצעות פונקצייה אחרת.

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

```
void deleteCustomer(customerPtr customer)
{
    customerPtr temp, prev;
    temp = currentCustomers;
    prev = NULL;
    if (currentCustomers == NULL)
    {
        printf("Can't delete from empty list.\n");
    }
    else if (customer == temp)
    {
        currentCustomers = _____ (1) _____;
    }
    else
    {
        while ((temp != NULL) && (_____ (2) _____))
        {
            prev = _____ (3) _____;
            temp = temp->next;
        }
        if (temp == NULL)
        {
            printf("Customer not in list, can't be deleted.\n");
        }
        else
        {
            prev->next = _____ (4) _____;
        }
    }
    _____ (5) _____;
    return;
}
```

(5 נק') ז. לפניך פונקצייה שכותרתה:

bikePtr returnBike(void)

פונקצייה זו מבקשת מן הלקוח להזין את מספר הזהות שלו ואת מיקומו הנוכחי.

הפונקצייה מוסיפה את זוג האופניים שאותו מעוניינים להשיב לתחנת ההשכרה המוצבת במיקום שהתקבל.

כמו כן, היא מחשבת את משך ההשכרה ואת הסכום הנדרש לתשלום, ולבסוף – מטפלת בגביית התשלום ובמחיקת פרטי הלקוח מן המערכת.

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

bikePtr returnBike(void)

```
{
    locationPtr returnLocation;
    unsigned long int userID;
    bikePtr bike;
    customerPtr customer;
    time_t now;
    double secondsUsed;
    int payment;
    userID = inputUserID();
    returnLocation = inputCurrentLocation();
    customer = _____(1)_____;
    now = _____(2)_____;
    secondsUsed = difftime(now, customer->begin);
    bike = _____(3)_____;
    bike = deleteBikeFromUseList(bike);
    payment = calculatePayment(_____(4)_____);
    paymentCollection(payment, customer);
    bike = addBikeToIntersection(_____(5)_____, _____(6)_____);
    _____(7)_____;
    return bike;
}
```

(5 נק') ח. לפניך כותרת של פונקציית **רקורסיבית**:

```
void printBikeList(bikePtr p)
```

פונקציית זו מקבלת את p, שהוא מצביע לראש רשימה מקושרת של אופניים. הפונקציית מדפיסה את רשימת מספרי הזיהוי של האופניים המופיעים ברשימה.

כתוב במחברתך **מימוש רקורסיבי מלא** לפונקציית הזו.

(5 נק') ט. לפניך כותרת של פונקציית:

```
int calculatePayment(double seconds)
```

פונקציית זו מקבלת את seconds, המציין את משך ההשכרה בשניות. הפונקציית מחשבת ומחזירה את הסכום שהלקוח נדרש לשלם בעבור השכרת האופניים.

תזכורת: התעריף להשכרת זוג אופניים מוגדר על-ידי הקבוע HOUR_RATE, והוא זהה לשעה מלאה או לחלק ממנה. לדוגמה: בעבור הקלט 8523, המייצג שעתיים ו-22 דקות ו-3 שניות, יידרש הלקוח לשלם בעבור שלוש שעות מלאות.

כתוב במחברתך **מימוש מלא** לפונקציית הזו.

שאלה 2 – שאלת חובה (10 נקודות)

מעוניינים לאחסן אוסף של משפטים, שכל משפט בו הוא אוסף של מילים.

אוסף המשפטים ייוצג על-ידי רשימה מקושרת של רשימות מקושרות. כל איבר ברשימה המקושרת הראשונית ייצג משפט, וכל איבר ברשימה המקושרת המשנית ייצג מילה במשפט.

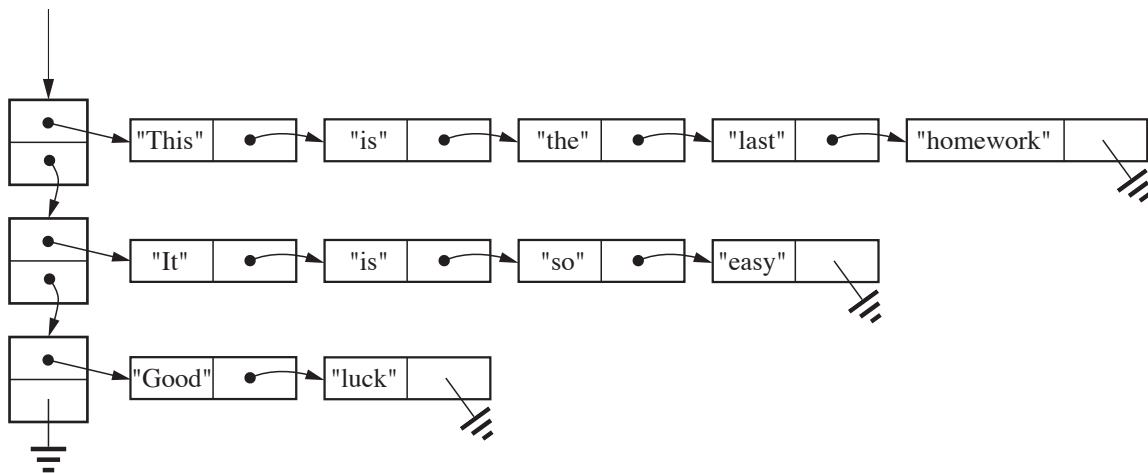
דוגמה: בעבור אוסף המשפטים שלהלן:

This is the last homework

It is so easy

Good luck

מבנה הנתונים המתאים יהיה:



איור א' לשאלה 2

לשם כך הוגדרו המבנים שלהלן:

```
struct word // מבנה המייצג מילה
{
    char *s;
    struct word *next;
};
```

```
struct sentence // מבנה המייצג משפט
{
    struct word *wordlist;
    struct sentence *next;
};
enum{FALSE, TRUE};
```

הנח כי מבנה הנתונים קיים, ומכיל אוסף של משפטים בהתאם להגדרה שלעיל.

(2 נק') א. לפיך הפונקצייה שכותרתה:

```
void printSentence(struct sentence *list)
```

פונקצייה זו מקבלת את list, שהוא מצביע למשפט, ומדפיסה את כל המילים שמופיעות במשפט הזה. בפונקצייה שלהלן חסרים שני ביטויים, המסומנים במספרים בין סוגריים עגולים. כתוב במחברת הבחינה את מספרי הביטויים החסרים (1) – (2), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
void printSentence(struct sentence *list)
{
    struct word *tmp;
    for (tmp = _____(1)_____; tmp; tmp = tmp->next)
        printf(_____(2)_____);
    printf("\n");
}
```

(4 נק') ב. לפיך הפונקצייה שכותרתה:

```
void findWord(struct sentence *list, char *s)
```

הפונקצייה מקבלת את list, שהוא מצביע לראש הרשימה המקושרת של המשפטים, ואת s, שהוא מצביע למחרוזת. הפונקצייה מדפיסה את כל המשפטים שבמבנה הנתונים שעליו מצביע list, אשר מופיעה בהם המחרוזת s. **דוגמה:** בעבור אוסף המשפטים המוצג באיור א', והמחרוזת "is", הפונקצייה תדפיס את המשפטים:

This is the last homework

It is so easy

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

```
void findWord(struct sentence *list, char *s)
{
    struct word *tmp;
    for (; list; list = list-> next)
    {
        for (_____(1)_____; tmp; tmp = _____(2)_____)
        {
            if (strcmp(_____(3)_____) == _____(4)_____)
            {
                PrintSentence(list);
                break;
            }
        }
    }
}
```

(4 נק') ג. לפניך הפונקצייה שכותרתה:

```
int isDuplicate(struct sentence *list)
```

הפונקצייה מקבלת את list, שהוא מצביע לראש הרשימה המקושרת של המשפטים.

הפונקצייה מחזירה 1 אם קיימים לפחות שני משפטים זהים במבנה הנתונים שעליו מצביע list, אחרת – הפונקצייה מחזירה 0.

לדוגמה: בעבור אוסף המשפטים המוצג בדוגמה שבפתיח לשאלה, הפונקצייה תחזיר 0.

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

```
int isDuplicate(struct sentence *list)
{
    struct sentence *s;
    struct word *w1, *w2;
    int flag;
    for (; list->next; list = list->next)
    {
        for (s = list->next; s; _____(1)_____)
        {
            flag = 1;
            for (w1 = list->wordlist, w2 = s->wordlist; _____(2)_____ &&
                w2 != NULL; w1 = w1->next, _____(3)_____)
            {
                if (_____(4)_____(w1->s, w2->s) != 0)
                {
                    flag = _____(5)_____;
                    break;
                }
            }
            if ((_____(6)_____) && (_____(7)_____) && (w2 == NULL))
            {
                return 1;
            }
        }
    }
    return 0;
}
```

פרק שני (15 נקודות)

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

שאלה 3

(10 נק') א. נתונה תוכנית הקולטת משפט מן המשתמש.

הנח כי המשפט מכיל אותיות אנגליות בלבד (גדולות וקטנות), וכן כי כל תו אחר ('!', '!', '%', ...) מציין הפרדה בין שתי מילים במשפט, כאשר התו 'n' מציין את סוף המשפט.

כמו כן, כל מילה במשפט מכילה 24 תווים לכל היותר.

התוכנית מדפיסה את מילות המשפט בזו אחר זו, כאשר כל מילה במשפט מודפסת בסדר אותיות הפוך.

דוגמה:

בעבור המשפט: Hi, what's up?

הפלט שיתקבל הוא: iH, tahw's pu?

לצורך ביצוע הפעולה, התוכנית משתמשת בפונקציית **רקורסיבית** שכותרתה:

```
void print_rev(char *word, int size)
```

המקבלת את word, שהוא מצביע למחרוזת המייצגת מילה בודדת, ואת size, המייצג את אורך המחרוזת שעליה מצביע word.

פונקציית זו מדפיסה בסדר הפוך את אותיות המילה שקיבלה.

לדוגמה: בעבור המחרוזת Hi, הפונקציית תחזיר את המחרוזת iH.

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

```

#include <stdio.h>
#define MAX_LEN 24

void print_rev(char *word, int size);

int main()
{
    char c, word[MAX_LEN];
    int i = 0;
    int in_word = 0;
    printf("Please Enter a sentence\n");
    while ((c=getchar()) != '\n')
    {
        if(_____(1)_____)
        {
            word[i] = c;
            _____(2)_____;
            in_word = 1;
        }
        else
        {
            if (in_word)
                print_rev(_____(3)_____);
            printf("%c", c);
            i = _____(4)_____;
            _____(5)_____ = 0;
        }
    }
    if (in_word)
        _____(6)_____ (word, i);
    putchar('\n');
}

void print_rev(char *word, int size)
{
    if (size > 0)
    {
        printf("%c", word[size - 1]);
        print_rev(word, _____(7)_____);
    }
}

```

(2.5 נק') ב. לפניך תוכנית בשפת C :

```
#include <stdio.h>
int main(void)
{
    int a = 0;
    while(a++)
    {
        printf("good morning");
    }
    return 0;
}
```

כתוב במחברתך את מספרו של ההיגד הנכון:

1. תתרחש שגיאה בעת ריצתה של התוכנית
2. התוכנית תרוץ, אך לא תדפיס דבר
3. התוכנית תדפיס את המשפט good morning פעם אחת
4. התוכנית תדפיס את המשפט good morning מספר רב של פעמים

(2.5 נק') ג. לפניך תוכנית בשפת C :

```
#include <stdio.h>
int main()
{
    typedef int a;
    a b=1, c=2, d=3;
    d += b*b + c;
    printf("%d", d);
    return 0;
}
```

כתוב במחברתך את מספרו של ההיגד הנכון:

1. תתרחש שגיאת קומפילציה
2. הערך שיודפס הוא 2
3. הערך שיודפס הוא 6
4. הערך שיודפס הוא 12

שאלה 4 (15 נקודות)

(10 נק') א. לפניך פונקצייה שכותרתה:

```
int fixWord(char wordIn[], char wordOut[])
```

פונקצייה זו מקבלת את הפרמטרים האלה:

wordIn – מחרוזת מילים באנגלית

wordOut – פרמטר שיכיל את המחרוזת wordIn לאחר שנעשו בה שינויים

הפונקצייה מבצעת על המחרוזת wordIn את הפעולות שלהלן:

1. מחליפה כל אות קטנה באות גדולה מתאימה (b→B).

2. מסירה את אותיות הניקוד הגדולות והקטנות (A, E, I, O, U, a, e, i, o, u) מכל המילים שבמחרוזת.

הפונקצייה משכנת במחרוזת wordOut את המחרוזת wordIn לאחר שנעשו בה השינויים, ומחזירה כערך את אורכה של המחרוזת wordOut.

תזכורת: islower היא פונקציית ספרייה המקבלת תו ומחזירה 1 אם התו הוא אות קטנה, אחרת - הפונקצייה מחזירה 0.

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

```
int fixWord(char wordIn[], char wordOut[])
{
    int i, j;
    for(i = 0, j = 0; _____(1)_____ != '\0'; i++)
    {
        if (_____(2)_____ ("aeiouAEIOU", wordIn[i]))
            continue;
        if (islower(_____(3)_____))
            wordOut[_____(4)_____] = wordIn[i] - (_____(5)_____);
        else
            wordOut[_____(6)_____] = wordIn[i];
    }
    wordOut[j] = _____(7)_____;
    return _____(8)_____;
}
```

(2.5 נק') ב. לפניך תוכנית בשפת C :

```
#include <stdio.h>

void main(void)
{
    int a = 2;
    int *p = &a;
    int *q = p++;
    int b = p - q;
    printf("%d\n", b);
}
```

כתוב במחברתך את מספרו של ההיגד הנכון:

1. הפלט שיודפס תלוי במספר הבתים המשמשים לייצוג משתנה מטיפוס int
2. הפלט שיודפס הוא 1
3. הפלט שיודפס הוא 2
4. הפלט שיודפס הוא 4

(2.5 נק') ג. לפניך תוכנית בשפת C :

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

#define square(x) x*x

int main()
{
    int i;
    i = 27/square(3);
    printf("%d", i);
    return 0;
}
```

מה יהיה הפלט של התוכנית? כתוב במחברתך את מספרו של ההיגד הנכון.

1. 1
2. 3
3. 9
4. 27

חלק ב': שפת סף (40 נקודות)

בנספח א' לחלק ב' נתונים קודי ASCII של תווים נפוצים – תוכל להיעזר בו.

פרק שלישי (16 נקודות)

ענה על שאלה 5 – שאלת חובה.

שאלה 5 (16 נקודות)

לפניך תוכנית בשפת אסמבלי.

הנח כי במערכים ARR1 ו-ARR2 מאוחסנים מספרים לא-מסומנים.

```
SSEG SEGMENT STACK 'STACK'
    DB 100H DUP()
SSEG ENDS
DATA SEGMENT
    LEN EQU 6
    ARR1 DB 10H, 1AH, 0FH, 8H, 0H, 2CH
    ARR2 DB LEN DUP(0)
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:
    MOV AX, DATA
    MOV DS, AX
    MOV CX, LEN
    MOV SI, OFFSET ARR1
    MOV DL, 0
L1:    MOV AL, [SI]
    MOV [SI + LEN], AL
    SHR AL, 1          ; (*)
    JC  L2
    SHR AL, 1          ; (*)
    JC  L2
    INC DL
    MOV [SI + LEN], AL
L2:    ADD SI, 1
    LOOP L1
    MOV AH, 4CH
    INT 21H
CODE ENDS
END START
```

- א. (3 נק') השלם את תמונת הזיכרון של המערך ARR1 **לפני** ביצוע התוכנית, וציין את תוכנו של כל בית (BYTE) כמספר בבסיס עשרוני.

העתק למחברתך את הטבלה שלהלן, והשלם בה את תמונת הזיכרון כנדרש.

היסט	5	4	3	2	1	0
תוכן						

- ב. (3 נק') השלם את תמונת הזיכרון של המערך ARR2 **לאחר** ביצוע התוכנית, וציין את תוכנו של כל בית (BYTE) כמספר בבסיס **הקסאדצימלי**.

העתק למחברתך את הטבלה שלהלן, והשלם בה את תמונת הזיכרון כנדרש.

היסט	5	4	3	2	1	0
תוכן						

- ג. (2 נק') מה יהיה ערכו של האוגר DL, בבסיס עשרוני, בתום הרצת התוכנית?

- ד. (3 נק') תאר **בקצרה** (עד שני משפטים) את תפקידו של האוגר DL.

- ה. (3 נק') תאר **בקצרה** (עד שני משפטים) מה מבצעת התוכנית באופן כללי. אין צורך להתייחס לכל הוראה בנפרד.

- ו. (2 נק') אם נחליף בקטע הקוד הנתון את כל השורות המסומנות ב- (*) בשורה 1, SAR AL, האם ישתנו ביצועי התוכנית? ענה "כן" או "לא" ונמק בקצרה.

פרק רביעי (24 נקודות)

ענה על שתיים מבין השאלות 6-8 (לכל שאלה – 12 נקודות).

שאלה 6 (12 נקודות)

לפניך תוכנית בשפת אסמבלי, הכוללת שגרה רקורסיבית בשם REC.

שגרה זו מקבלת – באמצעות המחשנית – את הארגומנט NUM שבו מאוחסן מספר, ומדפיסה אותו על המסך בתצוגה עשרונית.

אם המספר המאוחסן ב־NUM הוא מספר שלילי (לפי שיטת המשלים ל־2), אזי השגרה תדפיס לשמאלו גם את הסימן מינוס (-).

```
SSEG SEGMENT STACK 'STACK'
```

```
DB 100 DUP()
```

```
SSEG ENDS
```

```
DATA SEGMENT
```

```
NUM DW 65535
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS: CODE, DS:DATA
```

```
START:
```

```
MOV AX, DATA
```

```
MOV DS, AX
```

```
MOV BX, 10
```

```
CMP NUM, 0 } סימן שתי השורות יחד ;
```

```
JNL CON
```

```
NEG NUM
```

```
MOV AH, 2
```

```
MOV DL, '-'
```

```
INT 21H
```

```
CON: PUSH NUM
```

```
CALL REC
```

```
EXIT:MOV  AH,  4CH
      INT  21H

REC:
      PUSH BP
      MOV  BP,  SP
      MOV  AX,  [BP+4]
      XOR  DX,  DX
      DIV  BX
      MOV  [BP+4], DL
      OR   AX,  AX
      JE   SOF
      PUSH AX
      CALL REC
SOF: MOV  AH,  2
      MOV  DL,  [BP+4]
      ADD  DL,  '0'
      INT  21H
      POP  BP
      RET  2      ; (*)

CODE ENDS

END START
```

- (3 נק') א. עקוב אחר הרצתה של התוכנית הנתונה, וכתוב מה תדפיס התוכנית על המסך בתום הריצה.
- (3 נק') ב. מהו תפקידו של קטע הקוד המסומן בסוגריים?
- (3 נק') ג. בעבור איזה ערך של NUM תדפיס התוכנית בתום ריצתה את הערך 5 ?
- (3 נק') ד. אם נחליף בתוכנית הנתונה את הפקודה המסומנת ב-(*) בפקודה RET , האם ישתנו ביצועי התוכנית? ענה "כן" או "לא", ונמק בקצרה.

שאלה 7 (12 נקודות)

לפניך תוכנית בשפת אסמבלי, שנתונים בה שני מערכים: ARR1 ו-ARR2.

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

```
SDATA SEGMENT
    ARR1 DB 00000001B
           DB 00000010B
           DB 00000100B
           DB 00001000B
           DB 00010000B
           DB 00100000B
           DB 01000000B
           DB 10000000B
    ARR2 DB 30 DUP(0)
```

```
SDATA ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS: CODE, DS: SDATA
```

```
START:
```

```
    MOV AX, SDATA
    MOV DS, AX
    XOR DI, DI
    MOV DX, 7
    XOR SI, SI
XX1: MOV BX, SI
    MOV CX, DX
XX2: INC BX
    MOV AL, ARR1[SI]
    OR AL, ARR1[BX]
    MOV ARR2[DI], AL
    INC DI
    LOOP XX2
    INC SI
    DEC DX
    JNE XX1
    MOV AH, 4CH
    INT 21H
```

```
CODE ENDS
```

```
END START
```

(4 נק') א. כתוב את תוכנם של 13 הבתים הראשונים של המערך ARR2, בבסיס בינארי, לאחר הרצתה של התוכנית.

(4 נק') ב. כמה בתים סך-הכול ייכתבו במערך ARR2 במהלך כל הרצתה של התוכנית?

(4 נק') ג. מה מאפיין את הנתונים שייכתבו לתוך המערך ARR2 במהלך הרצתה של התוכנית?

שאלה 8 (12 נקודות)

לפניך תוכנית בשפת אסמבלי, המטפלת בשני המשתנים שלהלן:

NUM_STR – משתנה המכיל מחרוזת בת ארבעה תווים, אשר כל אחד מתוויה הוא ספרה עשרונית

NUMBER – משתנה בעל 16 סיביות, אשר מאוחסן בו מספר עשרוני בן 4 ספרות

לדוגמה:

```
NUM_STR DB '2368'
```

```
NUMBER DW 2368
```

התוכנית בודקת אם ערכו של המשתנה NUMBER שווה לערך המספר המופיע כמחרוזת במשתנה NUM_STR.

אם כן – התוכנית תציב במשתנה RESULT את הערך 1, אחרת – התוכנית תציב את הערך 0.

לדוגמה:

1. בעבור הערכים האלה:

```
NUM_STR DB '1753'
```

```
NUMBER DW 1753
```

לאחר הרצת התוכנית, ערכו של המשתנה RESULT יהיה 1.

2. בעבור הערכים האלה:

```
NUM_STR DB '2368'
```

```
NUMBER DW 2863
```

לאחר הרצת התוכנית, ערכו של המשתנה RESULT יהיה 0.

3. בעבור הערכים האלה:

```
NUM_STR DB '5176'
```

```
NUMBER DW 4739
```

לאחר הרצת התוכנית, ערכו של המשתנה RESULT יהיה 0.

להלן התוכנית:

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

SDATA SEGMENT
    NUM_STR    DB    '9876'
    NUMBER     DW    9876
    DEC_ARR    DB    4 DUP(0)
    RESULT     DW    1
SDATA ENDS
```

SCODE SEGMENT

ASSUME CS:SCODE, DS:SDATA

START:

MOV AX, SDATA

MOV DS, AX

MOV SI, 0

MOV CX, 2

MOV AX, NUMBER

MOV BL, 100 ; (*)

DIV BL

MOV BH, AH

ABC: XOR AH, AH

MOV BL, 10

DIV BL

MOV DEC_ARR[SI], AL

MOV DEC_ARR[SI+1], AH

MOV AL, NUM_STR[SI]

AND AL, 0FH ; (**)

CMP AL, DEC_ARR[SI]

JNE NOT_EQ

MOV AL, NUM_STR[SI+1]

AND AL, 0FH

CMP AL, DEC_ARR[SI+1]

JNE NOT_EQ

MOV SI, 2

MOV AL, BH

DEC CX

JNZ ABC

JMP EXIT

NOT_EQ: DEC RESULT

EXIT: MOV AH, 4CH

INT 21H

SCODE ENDS

END START

- א. (4 נק') כתוב את תוכנם של תאי המערך DEC_ARR, בבסיס עשרוני, בתום הרצת התוכנית.
- ב. (4 נק') אם נחליף בתוכנית הנתונה את השורה המסומנת ב-(*) בשורה 10, MOV BL, 10, האם ישתנו ביצועי התוכנית? ענה "כן" או "לא", ונמק בקצרה.
- ג. (4 נק') אם נחליף בתוכנית הנתונה את השורה המסומנת ב-(**) בשורה 0', SUB AL, '0', האם ישתנו ביצועי התוכנית? ענה "כן" או "לא", ונמק בקצרה.

בהצלחה!

נספח א', לחלק ב': שפת סף
לשאלון 714001, אביב תשפ"א

קודי ASCII של תווים נפוצים

קוד עשרוני	קוד הקסאדצימלי	תו
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z

קוד עשרוני	קוד הקסאדצימלי	תו
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z

קוד עשרוני	קוד הקסאדצימלי	תו
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9

נספח: מילון מונחים (2 עמודים)

לשאלון 714001, אביב תשפ"א

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
חלק א' – תכנות מערכות בשפת C :			
initialization	инициализация	ابتداء	אתחול
composition	включение в себя, содержание в себе	ضمّ	הכלה
memory allocation	выделение памяти	تخصيص ذاكرة	הקצאת זיכרון
flag	обозначение конца, конечный элемент	عَلَم	זקיף
one-direction	однонаправленный	ذات اتّجاه واحد	חד־כיווני
type	тип	نوع	טיפוס
data structure	структура данных	بُنْيَة البيانات	מבנה נתונים
dynamic array	динамический массив	مصفوفة غير ثابتة	מערך דינמי
pointer	указатель	مُؤشّر	מצביע
global variable	глобальная переменная	متغيّر عامّ	משתנה גלובלי
bits series	последовательность битов	سلسلة أرقام ثنائية / سلسلة بتات	סדרת סיביות
parameter	параметр	متغيّر (بارامتر)	פרמטר
node	узел, вершина	مَفْرَق	צומת
fixed	константа	ثابت	קבוע
binary file	двоичный файл	مِلَفّ ثنائيّ	קובץ בינארי
linked list	связный список	قائمة مرتبطة	רשימה מקושרת
field	поле	حَقْل	שדה
concatenation	конкатенация	تَرابُط	שרשור
cell	ячейка	خَلِيّة	תא

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
חלק ב' – שפת סף:			
register	регистр	מخزن (ريغستر)	אוגר
main diagonal	главная диагональ	מائل رئيسي	אלכסון ראשי
hexadecimal base	шестнадцатеричная система счисления, основание 16	قاعدة الست عشريّة	בסיס הקסדצימלי
stack	стек	باغة	מחסנית
decimal base number	десятичное число	عدد عشريّ	מספר בבסיס עשרוני
marked number	обозначенное число, число со знаком	عدد مُعَلَّم	מספר מסומן
bits	биты	أرقام ثنائيّة (بتات)	סיביות
palindrome	палиндром	سِيَّاقٌ مُتَنَازِر (بولينדרום)	פלינדרום
compilation	компиляция	ترجمة الأوامر	קומפילציה (הידור)
routine	функция, рутина	رتابة / روتين	שגרה
recursive routine	рекурсивная функция	إجراء تراجعيّ	שגרה רקורסיבית