מדינת ישראל
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תשע"ו, 2016
סמל השאלון: 714001
נספח: מילון מונחים

תכנות מערכות בשפת C ושפת סף

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

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

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

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

בשאלון זה 45 עמודים ו־2 עמודי נספחים.

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

בהצלחה!

המשך מעבר לדף

השאלות

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

פרק ראשון (35 נקודות)

ענה על שתי השאלות 1–2.

שאלה 1 – שאלת חובה (20 נקודות)

לפניך הגדרה:

מטריצה דלילה (Sparse Matrix) היא מטריצה שמרבית איבריה הם בעלי הערך אפס.

מטריצה דלילה ניתנת לייצוג על־ידי מבני נתונים שונים שמשוכנים בהם רק איברים השונים מאפס. מבין המבנים השונים האפשריים, הוחלט לייצג את המטריצה הדלילה באמצעות רשימות מקושרות חד־כיווניות מעגליות.

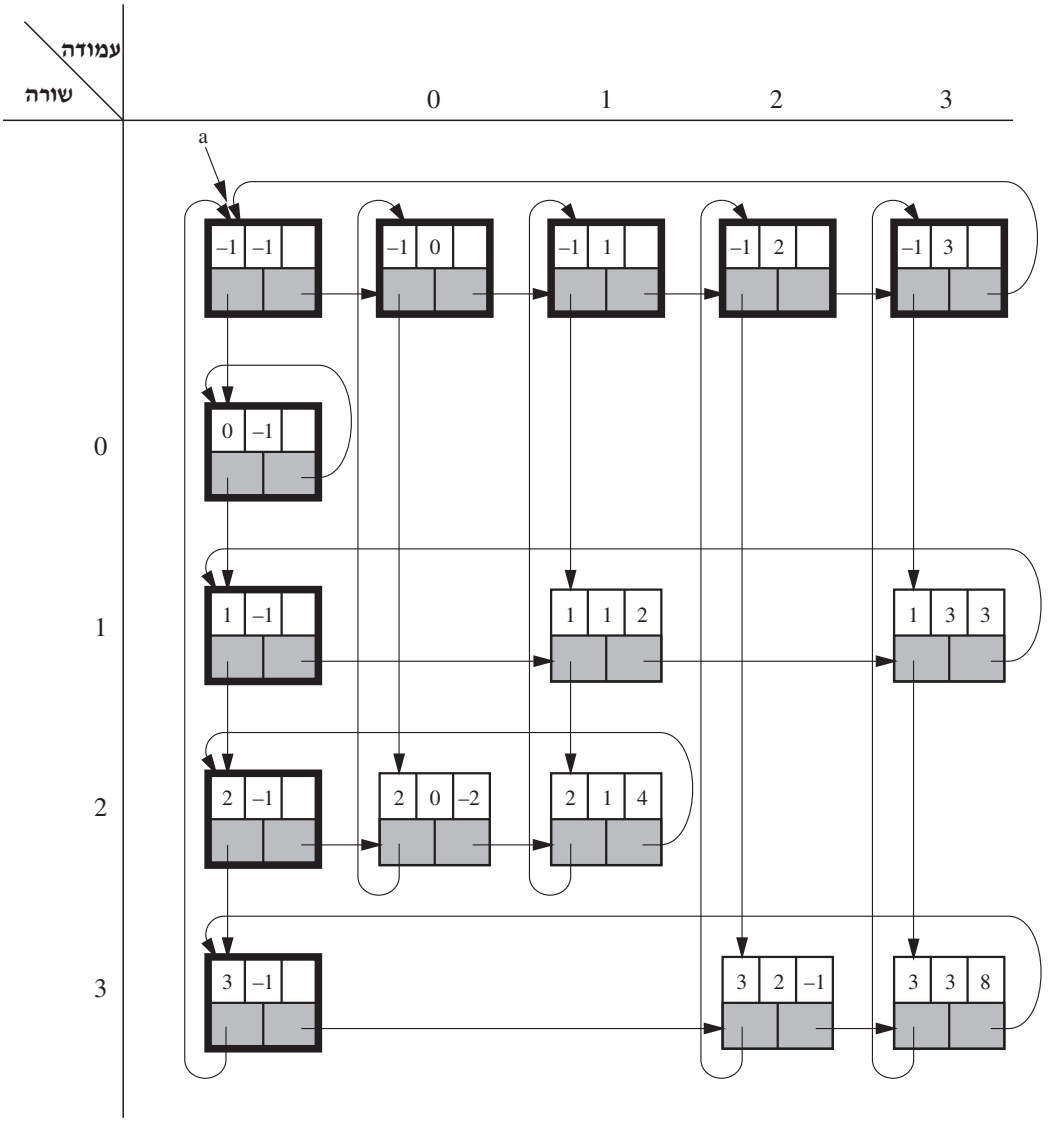
דוגמה:

נתונה **המטריצה הדלילה** הזאת:

	0	**1**	**2**	**3**
0	0	0	0	0
1	0	2	0	3
2	−2	4	0	0
3	0	0	−1	8

מטריצה זו תיוצג על־ידי מבנה הנתונים המתואר **באיור א'** שלהלן:

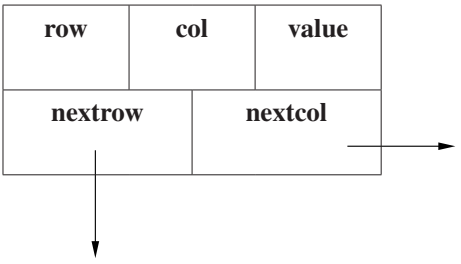
המשך בעמוד 3



איור א' לשאלה 1

המשך בעמוד 4

באיור ב' שלהלן מוצג תיאור סכמתי של צומת במבנה נתונים זה.



איור ב' לשאלה 1

row - מספר השורה

col - מספר העמודה

val - ערכו של האיבר המשוכן בצומת זה

nextrow - מצביע על האיבר הבא בעמודה col

nextcol - מצביע על האיבר הבא בשורה row

עתה נפרט את מבנה הנתונים שמייצג את המטריצה הדלילה.

כדי לאפשר גישה לאיבר הראשון שבכל **שורה**, מוסיפים עבור **כל** שורה צומת שיכונה **"צומת כותרת"**. לכל שורה יש **צומת כותרת נפרד**, גם כאשר השורה מכילה רק אפסים.
מצומתי הכותרות האלה ניצור רשימה מקושרת חד־כיוונית מעגלית.

באופן אנלוגי, כדי לאפשר גישה לאיבר הראשון שבכל **עמודה**, מוסיפים עבור **כל** עמודה צומת שיכונה **"צומת כותרת"**. לכל עמודה יש **צומת כותרת נפרד**, גם כאשר העמודה מכילה רק אפסים.
מצומתי הכותרות האלה ניצור רשימה מקושרת חד־כיוונית מעגלית **נוספת**.

המשך בעמוד 5

לשתי הרשימות הללו יש צומת **משותף** שיכונה **"ראש המטריצה"** כאשר:

בשדה **row** הוא יכיל את הערך (1–) .

בשדה **col** הוא יכיל את הערך (1–) .

בשדה **val** הוא לא יכיל ערך כלל.

השדה **nextrow** - יצביע לצומת הכותרת של השורה 0 .

השדה **nextcol** - יצביע לצומת הכותרת של העמודה 0 .

להלן מבנה של **צומת הכותרת** ברשימת צומתי הכותרת בעבור **השורות**.

צומת כותרת בעבור השורה ה־i יכיל:

בשדה **row** את הערך i

בשדה **col** את הערך (1–)

בשדה **val** שום ערך

בשדה **nextrow**: אם השורה ה־i היא השורה האחרונה במטריצה אזי

שדה זה יצביע ל**"ראש המטריצה"**;

אחרת –

שדה זה יצביע לצומת הכותרת של השורה ה־(i+1) .

בשדה **nextcol**: אם השורה ה־i היא שורת אפסים אזי

שדה זה יצביע לצומת הכותרת של השורה ה־i ;

אחרת –

שדה זה יצביע לצומת שמכיל את האיבר הראשון השונה מאפס בשורה ה־i .

המשך בעמוד 6

להלן מבנה של **צומת כותרת** ברשימת צומתי הכותרת בעבור **העמודות**.

צומת כותרת בעבור העמודה ה־j יכיל:

בשדה **row** את הערך (–1) .

בשדה **col** את הערך j .

בשדה **val** שום ערך.

בשדה **nextrow** : אם העמודה ה־j היא עמודת אפסים אזי

שדה זה יצביע לצומת הכותרת של העמודה ה־j ;

אחרת –

שדה זה יצביע לצומת שמכיל את האיבר הראשון השונה מאפס

שבעמודה ה־j .

בשדה **nextcol** : אם העמודה ה־j היא העמודה האחרונה של המטריצה אזי

שדה זה יצביע ל"**ראש המטריצה**";

אחרת –

שדה זה יצביע לצומת הכותרת של העמודה ה־ (j+1) .

להלן תיאור הייצוג של **האיברים השונים מאפס** במטריצה הדלילה:

בעבור השורה ה־i של המטריצה הדלילה מחזיקים רשימה חד־כיוונית מעגלית של צמתים המייצגים את האיברים **השונים מאפס** שבשורה זו.

לכל שורה נחזיק רשימה חד־כיוונית מעגלית **נפרדת** (ראה איור א').

בעבור העמודה ה־j של המטריצה הדלילה מחזיקים רשימה חד־כיוונית מעגלית של צמתים המייצגים את האיברים **השונים מאפס** שבעמודה זו.

לכל עמודה נחזיק רשימה חד־כיוונית מעגלית **נפרדת** (ראה איור א').

בעבור כל איבר (שערכו שונה מאפס) שנמצא בשורה ה־i ובעמודה ה־j של המטריצה הדלילה, נחזיק **צומת אחד בלבד** שיימצא ברשימה המעגלית של השורה ה־i וברשימה המעגלית של העמודה ה־j (ראה איור א').

המשך בעמוד 7

להלן הגדרת המבנה של צומת בשפת C שתואר באיור ב':

הערה: צומת זה מייצג **גם** צומת כותרת **וגם** צומת שבו משוכן איבר השונה מאפס במטריצה הדלילה.

```
typedef struct nodeType
{
   int row;    // מספר השורה
   int col;   // מספר העמודה
   int val;    // אם הצומת הוא צומת כותרת אזי שדה זה לא יכיל ערך;
               אחרת – שדה זה יכיל איבר השונה מאפס במטריצה הדלילה
   struct nodeType *nextrow;           // שדה קישור
   struct nodeType *nextcol;            // שדה קישור
} node,*nodeptr, *matrix;
```

להלן הגדרות של קבועים המציינים את גודל המטריצה:

```
#define M 8       //   מספר השורות במטריצה
#define N 10      // מספר העמודות במטריצה
```

המשך בעמוד 8

נתונה ספריית פונקציות המכילה בין היתר את הפונקציה הזאת:

```
nodeptr findnear(matrix a, int r, int c)
```

פונקציה זו מקבלת את הפרמטרים האלה:

a – מצביע ל"ראש המטריצה" במטריצה הדלילה,

r – מספר שורה במטריצה הדלילה,

c – מספר עמודה במטריצה הדלילה.

פונקציה זו מחזירה מצביע לצומת קיים הנמצא בשורה r , בעמודה הקטנה מ־c והקרובה ביותר אליה.

דוגמאות על סמך איור א':

1. **findnear**(a,1,1) – תחזיר מצביע לצומת שבו row שווה 1 ו־col שווה (–1) .
2. **findnear**(a,3,3) – תחזיר מצביע לצומת שבו row שווה 3 ו־col שווה 2 .
3. **findnear**(a,2,3) – תחזיר מצביע לצומת שבו row שווה 2 ו־col שווה 1 .

הנחות יסוד:

$-1 < r < M$, $-1 < c < N$ כאשר M ו־N הם קבועים וידועים מראש.

הערה: גם אם בשורה ה־r ובעמודה ה־c שבמטריצה לא קיים כלל איבר השונה מאפס, תמיד יוחזר מצביע לצומת העונה על הדרישה **(ראה לעיל דוגמה 3)**.

המשך בעמוד 9

הנח שהפונקציה הזאת כתובה וניתן להשתמש בה בכל הסעיפים הבאים בלי לכתוב אותה מחדש. כמו כן, בעבור כל סעיף תוכל להשתמש בכל פונקציה שמומשה בסעיפים שלפניו.

ענה על הסעיפים הבאים:

א. לפניך פונקציה שכותרתה:

```
nodeptr findabove(matrix a,int r,int c)
```

פונקציה זו מקבלת את הפרמטרים:

a – מצביע ל"**ראש המטריצה**" במטריצה הדלילה

r – מספר שורה במטריצה הדלילה

c – מספר עמודה במטריצה הדלילה

פונקציה זו מחזירה מצביע לצומת קיים הנמצא בעמודה c בשורה הקטנה מ־r והקרובה ביותר אליה.

דוגמאות (על סמך איור א'):

1. **findabove**(a,1,1) – תחזיר מצביע לצומת שבו row שווה 1– ו־ col שווה 1 .
2. **findabove**(a,3,3) – תחזיר מצביע לצומת שבו row שווה 1 ו־ col שווה 3 .

הנחות יסוד: $-1 < r < M$, $-1 < c < N$ כאשר M ו־N קבועים וידועים מראש.

הערה: גם אם בשורה ה־r ובעמודה ה־c שבמטריצה לא קיים כלל איבר השונה מאפס, תמיד יוחזר מצביע לצומת העונה על הדרישה.

המשך בעמוד 10

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

```
nodeptr findabove(matrix a,int r,int c)
 {
   nodeptr p,q,z;
   p = a;
   while(p->col < c) p =_______(1)_______;
   q = p;
   z = p->nextrow;
   while((p->row < r)&&(________(2)______))
   {
      q= p;
      p = p->nextrow;
      z = p->nextrow;
   }
   if(_______(3)______)
      return q;
   else
      return p;
 }
```

המשך בעמוד 11

ב. לפניך פונקציה שכותרתה:

```
void insertafter(nodeptr p,nodeptr q,int x)
```

הפונקציה מקבלת את הפרמטרים:

x – מספר שלם

p – מצביע לצומת כלשהו במטריצה הדלילה

q – מצביע לצומת כלשהו במטריצה הדלילה

פונקציה זו מוסיפה צומת חדש שערכו x **בשורה** p->row **ובעמודה** q->col .

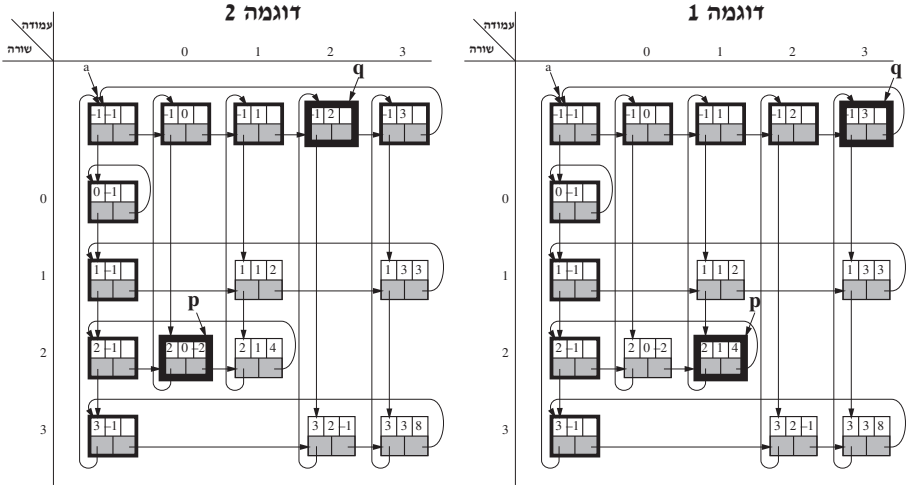
מיקומו של הצומת שנוסף יהיה מיד אחרי הצומת שעליו מצביע p ומיד אחרי הצומת שעליו מצביע q .

הנחות יסוד:

1. p ו־q **אינם** מצביעים על אותו הצומת.
2. בשורה p->row ובעמודה q->col לא קיים איבר במבנה המייצג את המטריצה הדלילה.
3. על מנת שהפונקציה תבצע את הנדרש, הפרמטרים p ו־q שהפונקציה תקבל יהיו כפופים לשני האילוצים האלה:
   - א. בשורה **p->row** לא קיים צומת מהעמודה p->col+1 ועד העמודה q->col .
   - ב. בעמודה **q->col** לא קיים צומת מהשורה q->row+1 ועד השורה p->row .

לדוגמה:

בהמשך לאיור א' שלעיל, בעת הקריאה לפונקציה, היא **לא תוכל לקבל** את הפרמטרים p ו־q המתוארים באיור ג' שלהלן:



איור ג' לשאלה 1

המשך בעמוד 12

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

```
void insertafter(nodeptr p,nodeptr q,int x)
 {
    nodeptr z;
    z = malloc(sizeof(node));
    z->row = p->row;
    z->col = q->col;
    z->val = x;
    z->nextrow =________(1)______;
    z->nextcol =__________(2)______;
    _______(3)_____= z;
    _______(1)_____= z;

 }
```

ג. לפניך פונקציה שכותרתה:

```
void deleteafter(nodeptr p,nodeptr q,int *x)
```

פונקציה זו מקבלת את הפרמטרים האלה:

p – מצביע לצומת כלשהו במטריצה הדלילה

q – מצביע לצומת כלשהו במטריצה הדלילה

פונקציה זו מבטלת את הצומת שעליו מצביעים p->nextcol ו־q->nextrow .

אם הצומת הזה הוא **צומת הכותרת** של שורה מסוימת או של עמודה מסוימת
או אם p->nextcol ו־q->nextrow אינם מצביעים על אותה צומת, אזי
הפונקציה מדפיסה את ההודעה "ILLEGAL_REQUEST" ;

המשך בעמוד 13

אחרת -

הפונקציה מבטלת את הצומת שעליו מצביעים p->nextcol ו־q->nextrow (כלומר, הצומת שנמצא מיד אחרי הצומת שעליו מצביע p בשורה שלו וגם מיד אחרי הצומת שעליו מצביע q בעמודה שלו). הפונקציה מחזירה במשתנה x את הערך של האיבר המשוכן בצומת שבוטל.

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.

```
void deleteafter(nodeptr p,nodeptr q,int *x)
 {
   nodeptr z;
   if((______(1)______)||(______(2)______)||(p->nextcol != q->nextrow))
      printf("ILLEGAL_REQUEST");
   else
    {
      z = p->nextcol;
      ______(3)______= z->nextcol;
      ______(4)______= z->nextrow;
      *x = z->val;
      free(z);
    }
 }
```

המשך בעמוד 14

ד. לפניך פונקציה שכותרתה:

```
void insert(matrix a,int r,int c,int v)
```

פונקציה זו מקבלת את הפרמטרים:

a – מצביע ל"**ראש המטריצה**" במטריצה הדלילה

r – מספר שורה במטריצה הדלילה

c – מספר עמודה במטריצה הדלילה

v – מספר שלם השונה מאפס

פונקציה זו מוסיפה את הערך v בשורה r ובעמודה c במטריצה שעליה מצביע a .

אם קיים צומת (המכיל איבר השונה מאפס) בשורה r ובעמודה c שבמטריצה שעליה מצביע a , אזי הפונקציה מציבה בצומת זה את הערך v ;

אחרת –
הפונקציה מוסיפה את v לשורה r ולעמודה c במטריצה שעליה מצביע a .

הנחות יסוד:

$-1 < r < M$, $-1 < c < N$ כאשר M ו־N קבועים וידועים מראש.

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

```
void insert(matrix a,int r,int c,int v)
  {
  nodeptr p,q;
  p =  ___________(1)___________;
  q = findabove(a,r,c);
  if ((q->nextrow->row == r) && (p->nextcol->col == c))
       ___________(2)___________;
  else
       ___________(3)___________;
  }
```

המשך בעמוד 15

ה. לפניך פונקציה שכותרתה:

```
void multiplyRow(matrix a,int r,int c)
```

פונקציה זו מקבלת את הפרמטרים:

a – מצביע ל"**ראש המטריצה**" במטריצה הדלילה

r – מספר שורה במטריצה הדלילה

c – מספר שלם השונה מאפס

הנח כי: $0 \le r < M$.

פונקציה זו כופלת את כל איברי השורה r במספר c .

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

```
void multiplyRow(matrix a,int r,int c)
 {
   nodeptr q,above;
   q = ____________(1)____________;
   q = q->nextcol;
   while(________(2)____________)
   {
     q->val = (q->val)*c;
     q = q->nextcol;
   }
 }
```

המשך בעמוד 16

ו. לפניך פונקציה שכותרתה:

```
void multRowAndAdd(matrix a,int row1,int row2, int c)
```

פונקציה זו מקבלת את הפרמטרים:

a – מצביע ל"**ראש המטריצה**" במטריצה הדלילה

row1 – מספר שורה במטריצה הדלילה

row2 – מספר שורה במטריצה הדלילה

c – מספר שלם השונה מאפס

פונקציה זו מחברת לשורה row1 את הכפולה של השורה row2 במספר c .

הנחות יסוד:

1. row1 שונה מ־row2 .
2. $-1 < row1 < M$, ו־ $-1 < row2 < M$ כאשר M קבוע וידוע מראש.

שים לב:

כתוצאה מפעולת החיבור, עשוי להיווצר בשורה row1 איבר שערכו 0 . בעבור איבר כזה לא יימצא צומת במבנה המקושר שתואר לעיל.

המשך בעמוד 17

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.

```
void multRowAndAdd(matrix a,int row1,int row2, int c)
 {
   nodeptr r1,r2,q,above;
   int x;
   q = _________(1)________;
   r1 = q->nextcol;
   r2 = _________(2)________;
   r2 = r2->nextcol;
   while(_________(3)_______)
   {
      while((r1->col < r2->col) && (_________(4)______))
      {
         q = r1;
         r1 = r1->nextcol;
      }
      if(r1->col == r2->col)
      {
         r1->val += c*(r2->val);
         if(________(5)_______)
```

המשך בעמוד 18

```
        {
            above = findabove(a,row1,r1->col);
            ____________(6)___________;
        }
        r1 = q->nextcol;
        r2 = r2->nextcol;
    }
    else
    {
        above = ___________(7)_______;
        _____________(8)____________;
        q  = q->nextcol;
        r1 = q->nextcol;
        r2 = r2->nextcol;
    }
  }
}
```

המשך בעמוד 19

שאלה 2 – שאלת חובה (15 נקודות)

א. לפניך תכנית בשפת C :

```
#include <stdio.h>
#include <string.h>
int main()
{
      char s1[20] = "abrahami 5";
      char s2[] = "hilaeuven";
      int i;
      char *p;
      p = &s1[3];
      printf("%d \n", *p - *s1);
      printf("%d \n", p - s1);
      printf("%s \n", ++p);
      p[-1] = 0;
      printf("%s \n", s1+1);
      strcpy(p,s1+2);
      printf("%s \n", p);
      strcat(p,s2+4);
      printf("%s \n", p);
      getchar();
      getchar();
      return 0;
}
```

רשום במחברתך רק את הפלט המדויק של התכנית הנתונה.

המשך בעמוד 20

ב. לפניך תכנית בשפת C :

```
#include <stdlib.h>
#include <stdio.h>
int main()
{
      int **a = NULL;
      int i = 0;
      a = (int**)malloc(4 * sizeof(int*));
      a[0] = (int*)malloc(16 * sizeof(int));
      for(i=0; i<16; i++) a[0][i] = (i+1)*4;
      a[1] = a[0]+4;
      a[2] = a[1]+5;
      a[3] = a[2]+3;
      printf("%d\n", a[0][4]);
      printf("%d\n", a[1] - a[0]);
      printf("%d\n", a[1][8]);
      printf("%d\n", a[2][3]);
      printf("%d\n", *(*(a+3)+1));

      free(a[0]);
      free(a);
      return 0;

}
```

רשום במחברתך רק את הפלט המדויק של התכנית הנתונה.

המשך בעמוד 21

ג. לפניך תכנית בשפת C :

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
 int main()
 {
     char st[6];
     char *s;
     char ch = 'a';
     int i;
     strcpy(st+1,"bc");
     s = st;
     (*s)=ch;
     s+=4;
     while((s-st)<6)
     {
       (*s)=(*st)+(s-st);
       s++;
     }
     (*s)='\0';
     printf("%s %s \n",st,st+4);
     *(st+3)=*(st+4)-1;
     printf("%s %s \n",st,st+4);
     return 0;
 }
```

רשום במחברתך רק את הפלט המדויק של התכנית הנתונה.

המשך בעמוד 22

ד. לפניך פונקציה שכותרתה:

```
char *trouble(char *d, char *s, int num1, int num2)
```

פונקציה זו מקבלת מחרוזת בשם s ומחזירה מחרוזת כדלהלן:

אם num1 + num2 גדול מאורכה של המחרוזת s , אז הפונקציה תחזיר מחרוזת ריקה.

אם num2 שווה ל־0 , אז היא תחזיר תת־מחרוזת של s החל מן האינדקס num1–1 עד לסופה של s ;

אחרת היא תחזיר תת־מחרוזת של s המכילה num2 תווים ברצף החל מן האינדקס num1–1 ואילך.

הנח כי: $0 < num1$ וגם $0 \leq num2$.

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.

```
char *trouble(char *d, char *s, int num1, int num2)
{
      char *t;
      int len, temp;
      len = __________(1)________;
      if (len < num1 + num2)
            return (NULL);
      else
      {
       if(num2 == 0)
             num2 = __________(2)________;
```

המשך בעמוד 23

```
        t = d;
        temp = num1 - 1;
        do
            {
                    _______(3)_______= ___________(4)________;
            } while(*s++ && num2--);
      }
      *(d - 1) = '\0';
      return (t);
}
```

פרק שני (15 נקודות)

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

שאלה 3 (15 נקודות)

א. להלן הגדרה של טיפוס הנתונים Worker המייצג רשומה של עובד במפעל מסוים:

```
typedef struct
{
      unsigned long id;
      char name[LEN]
} Worker;
```

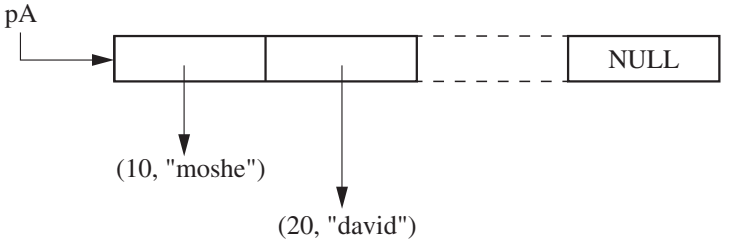
המשך בעמוד 24

כמו כן, נתונים שני מערכים, pA ו־idArr , המפורטים להלן:

1. pA – מערך של מצביעים למבנה Worker .
כל איבר במערך זה מצביע לרשומה אחת של עובד במפעל, ובצורה זו נשמרים פרטי כל עובדי המפעל.

הנח כי כל עובד מופיע במערך הזה פעם אחת בלבד.

מערך זה מסתיים בזקיף NULL (התא האחרון של המערך מצביע ל־NULL).



איור לשאלה 3

2. idArr – מערך של מספרים שלמים בלבד. כל איבר במערך זה הוא מספר הזהות של עובד או אורח המורשים להיכנס למתקן מסווג הנמצא בתוך המפעל.

הנח כי כל הערכים המשוכנים במערך הזה שונים זה מזה.

מערך זה מסתיים בזקיף EOID .

לפניך פונקציה המקבלת את שני המערכים האלה.

הפונקציה בונה מערך דינמי, pArr , ובו מספרי הזהות של **העובדים המורשים** בלבד (כלומר ללא מספרי הזהות של **האורחים המורשים**). אין צורך בהצבת זקיף בסופו של המערך הדינמי.

הפונקציה מחזירה את המערך pArr .

אם לא יימצאו ערכים, הפונקציה תחזיר את הערך NULL .

בפונקציה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים (1) – (5) . בכל אחד מן הסעיפים (1) – (5) שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות. רשום במחברתך את מספר הסעיף וציין לידו את האות המייצגת את התשובה הנכונה.

המשך בעמוד 25

```
unsigned long *func(const Worker **pA, const unsigned long *idArr)
{
   int i=0, j=0, size=0;
   unsigned long *pArr = NULL;
   while(pA[i])
   {
      j=0;
      while(____(1)____)
       {
          if(_____(2)_____)
           {
              pArr = (_____(3)_____);
              if(!pArr)
               {
                 printf("\nNot Enough Memory!");
                 exit(1);
               }
              pArr[size-1] = (____(4)____);
           }
          j++;
       }
      i++
   }
   ____(5)_____;
}
```

המשך בעמוד 26

1. הביטוי החסר (1) הוא:
   א. idArr[i]
   ב. idArr[j]
   ג. idArr[j] != NULL
   ד. idArr[j] != EOID

2. הביטוי החסר (2) הוא:
   א. pA[i]->id == idArr[j]
   ב. pA[i]->id != EOID
   ג. idArr[i] != EOID
   ד. pArr[j] == NULL

3. הביטוי החסר (3) הוא:
   א. realloc(pArr, size*sizeof(unsigned int))
   ב. realloc(pArr, ++size*sizeof(unsigned long))
   ג. realloc(pArr, size*sizeof(unsigned long))
   ד. realloc(pArr, size++*sizeof(unsigned int))

4. הביטוי החסר (4) הוא:
   א. idArr[size++]
   ב. idArr[j]
   ג. pA[i]
   ד. pA[j]

המשך בעמוד 27

5. הביטוי החסר (5) הוא:

א. return pA

ב. return idArr

ג. (pA != NULL) ? return pA : return NULL

ד. return pArr

ב. לפניך פונקציה LowArray אשר מקבלת שני פרמטרים, a ו־n , כדלהלן:

a – מערך של מצביעים למחרוזות לא ריקות המכילות אותיות גדולות בלבד.

הערה: אורכן של המחרוזות לא בהכרח שווה.

n – גודלו של המערך a .

פונקציה זו יוצרת ומחזירה מערך מצביעים חדש **s** אשר נבנה כדלהלן:

עבור המחרוזת שעליה מצביע a[i] , לכל 0<=i<n ושאורכה len , הפונקציה מבצעת את הפעולות האלה:

1. מציבה במשתנה j את האינדקס (המקום) המכיל את התו הקטן ביותר שבמחרוזת זו.
2. יוצרת מערך חדש שעליו יצביע s[i] שגודלו הוא len .
   הפונקציה מציבה במערך זה 1 באינדקס j , ומציבה 0 בכל אינדקס אחר.

לדוגמה:

אם נתון מערך המצביעים למחרוזות הבאות:

```
char* a[]={"GHCDEAF", "C", "XBXCGH", "DABC"};
```

אז הפונקציה יוצרת ומחזירה מערך דינמי s שלהלן (ראה אותיות מודגשות תואמות):

```
0000010

1

010000

0100
```

המשך בעמוד 28

הנך רשאי להשתמש בפונקציה int **minStr** (char* a) אשר מקבלת מצביע למחרוזת לא ריקה שמכילה אותיות גדולות בלבד, ומחזירה את האינדקס המכיל את התו הקטן ביותר שבמחרוזת.

הנחה: פונקציה זו נתונה ואין צורך לכתוב אותה מחדש.

```
int** LowArray(char** a, int n)
{
   int i,j;
   int** s = (int**)malloc(_____(1)_____);
   if(!s)
   {
     printf("not enough memory");
     exit(1);
   }
   for(i=0;i<n;i++)
   {
      s[i] = (int*)_____(2)_____(_____(3)_____, sizeof(int));
      if(!s[i])
      {
       printf ("not enough memory");
       exit(1);
      }
      j = _____(4)_____
      s[i][j] = 1;
   }
  _____(5)_____
}
```

המשך בעמוד 29

בפונקציה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים (1) – (5) . בכל אחד מן הסעיפים (1) – (5) שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות. רשום במחברתך את מספר הסעיף וציין לידו את האות המייצגת את התשובה הנכונה.

1. הביטוי החסר (1) הוא:
   - א. n*sizeof(char)
   - ב. sizeof(a)
   - ג. n*sizeof(char*)
   - ד. n*sizeof(int*)

2. הביטוי החסר (2) הוא:
   - א. calloc
   - ב. realloc
   - ג. alloc
   - ד. malloc

3. הביטוי החסר (3) הוא:
   - א. a[i]
   - ב. strlen(a)
   - ג. strlen(a[i])
   - ד. strlen(a[n]) – i

4. הביטוי החסר (4) הוא:
   - א. minStr (a[i]);
   - ב. minStr (a[i][j]);
   - ג. minStr (s[i]);
   - ד. minStr (s[i][j]);

המשך בעמוד 30

5. הביטוי החסר (5) הוא:

א. return a[i];

ב. return s[i];

ג. return s;

ד. return a;

שאלה 4 (15 נקודות)

א. לפניך תכנית בשפת C :

```
include <stdio.h>
#define f1(x) x*x
#define f2(x) *x
#define f3(x,y) x/y
#define f4(x, y, z) f3(y,z) + f1(x)
void main()
{
      int x, y=2, z=3, t=4, m=5;
      int v[2];
      int *ptr1, *ptr2;
      printf("1: %d\n", f4(z,t,y));
      ptr1 = &t;
      ptr2 = &m;
      printf("2: %d\n", f2(ptr1)f2(m));
      v[0] = f2(ptr2);
      v[1] = z;
```

המשך בעמוד 31

```
        while(f2(v))
        {
                v[1]+=2;   v[0]--;
        }
        printf("3: %d\n", v[1]);
        x = (f3(m,z) == f3(z,y)) ? 2 : 4;
        printf("4: %d\n", x);
        v[0] = y;
        v[1] = 0;
        switch(v[0])
        {
        case 1: v[1] += 1;
        case 2: v[1] += 2;
        case 3: v[1] += 3;
        default: v[1]++;
        }
        printf("5: %d\n", v[1]);
        v[0] = m;
        x=0;
        while(!(--v[0]%2))
        {
                v[0]--;
                x++;
                if(x==10) break;
        }
        printf("6: %d\n", v[0]);
        getchar();
}
```

רשום במחברתך רק את הפלט המדויק של התכנית הנתונה.

המשך בעמוד 32

ב. לפניך תכנית בשפת C :

```
#include <stdio.h>
void main()
{
unsigned int in, k=0;
printf("Enter number \n");
scanf("%x" , &in);
while(in)
   {
   k ++;
   in = in & (in - 1);
   }
printf("%u", k);
}
```

I. רשום במחברתך רק את הפלט המדויק של התכנית הנתונה בעבור הקלט 15 .

II. כתוב במחברתך את המספר שליד התשובה הנכונה.

התכנית מחשבת ומדפיסה תמיד את:

1. ערכו של in/5
2. מספר האחדות שיש **בַּייצוג הבינארי** של המספר הנקלט בבסיס עשרוני.
3. מקומו של הביט השמאלי ביותר הדלוק (שערכו 1) **בַּייצוג הבינארי** של הערך הנמצא במשתנה in בתום ביצוע הלולאה.
4. מספר הסיביות הדלוקות (שערכן 1) במקומות האי־זוגיים **בַּייצוג הבינארי** של הערך הנמצא במשתנה in בתום ביצוע הלולאה.

המשך בעמוד 33

חלק ב': שפת סף (50 נקודות)

פרק שלישי (20 נקודות)

ענה על שאלה 5 – שאלת חובה.

שאלה 5

לפניך תכנית בשפת אסמבלי הכוללת שגרה **רקורסיבית** בשם FUNC1 אשר מקבלת באמצעות מחסנית את NUM , שהוא מספר עשרוני שלם חיובי הגדול מאפס.

```
 DATA SEGMENT
      NUM DW 1053
 DATA ENDS
 SSEG SEGMENT STACK 'STACK'
      DB 100H DUP(?)
 SSEG ENDS
 CODE SEGMENT
  ASSUME CS:CODE,DS:DATA,SS:SSEG
  START:MOV AX,DATA
      MOV DS,AX
      PUSH NUM
      MOV CX,0              ;(0)
      CALL FUNC1
      MOV AH,4CH
      INT 21H
FUNC1 PROC
      PUSH BP
```

המשך בעמוד 34

```
        MOV BP,SP
        MOV AX,[BP+4[
        CMP AX,0
        JZ SOF
        MOV BX,10
        XOR DX,DX           ; (1)
        DIV BX
        INC CX              ; (2)
        PUSH AX
        CALL FUNC1
                            ; (3)
SOF:    POP BP
        RET 2
FUNC1 ENDP
CODE ENDS
END START
```

א. מה יהיה תוכנו של האוגר CX , בבסיס עשרוני, לאחר ביצוע התכנית הנתונה?

ב. אם נחליף בתכנית הנתונה את השורה **NUM DW 1053** בשורה **NUM DW 25612** , מה יהיה תוכנו של האוגר CX , בבסיס עשרוני, לאחר ביצוע התכנית?

ג. אם נחליף בתכנית הנתונה את השורה **NUM DW 1053** בשורה **NUM DW -64483** , האם תוכנו של האוגר CX , בבסיס עשרוני, יהיה שונה מהערך שהוחזר באוגר CX בסעיף א'? ענה "כן" או "לא".

המשך בעמוד 35

ד. להלן ארבעה היגדים שאחד מהם מתאר את הפעולה שמבצעת התכנית הנתונה. רשום במחברתך את מספרו של ההיגד הנכון.

- **היגד 1 :** השגרה הרקורסיבית מחזירה את הערך 0 אם המספר NUM הוא בבסיס בינארי, ומחזירה את הערך 1 אם המספר NUM הוא בבסיס עשרוני.
- **היגד 2 :** השגרה הרקורסיבית מחזירה את הערך של ספרת האחדות בבסיס עשרוני.
- **היגד 3 :** השגרה הרקורסיבית מחזירה את סכום הספרות שב־NUM בבסיס עשרוני.
- **היגד 4 :** השגרה הרקורסיבית מחזירה את מספר הספרות שב־NUM בבסיס עשרוני.

ה. אם נסיר בתכנית הנתונה את השורה המסומנת ב־(1) – האם שינוי זה עלול להשפיע על הערך המוחזר באוגר CX ? ענה "כן" או "לא".

ו. בסעיף זה הנח כי ערכו של המספר שמשוכן במשתנה NUM הוא 2559 לכל היותר.

אם נחליף בתכנית הנתונה את השורה **DIV BX** בשורה **DIV BL** - האם ייתכן שתוקרן אחת מהודעות המערכת DIVIDE BY ZERO או DIVIDE BY OVERFLOW ?
ענה "כן" או "לא".

ז. אם נחליף בתכנית הנתונה את השורה CALL FUNC1 בשורה JMP FUNC1 –
האם שינוי זה ישפיע על הערך המוחזר באוגר CX ?
ענה "כן" או "לא".

ח. אם נסיר מהתכנית הנתונה את השורה המסומנת ב־(1) ואותה נרשום בשורה המסומנת ב־(3) – האם שינוי זה ישפיע על הערך המוחזר באוגר CX ? ענה "כן" או "לא".

ט. אם נוסיף בתכנית הנתונה את השורה החדשה שלהלן: MOV SP , 50H אחרי השורה המסומנת ב־(0) – האם ביצועי התכנית ישתנו בהכרח ? ענה "כן" או "לא".

י. אם נחליף בתכנית הנתונה את השורה PUSH AX בשורה PUSH DX –
האם שינוי זה ישפיע על הערך המוחזר באוגר CX ? ענה "כן" או "לא".

המשך בעמוד 36

פרק רביעי (30 נקודות)

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

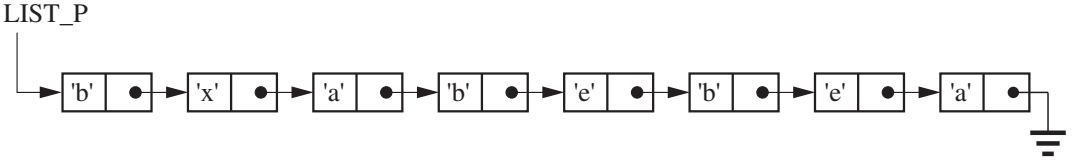
שאלה 6

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

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

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

נוסף על כך נתון כי המשתנה LIST_P הוא מצביע לצומת הראשון ברשימה, כלומר LIST_P מכיל את הכתובת של הצומת הראשון ברשימה (ראה איור א').

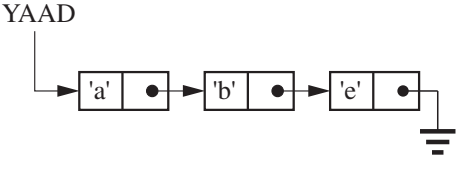


איור א' לשאלה 6

לפניך קטע תכנית בשפת אסמבלי אשר סורק את הרשימה הנתונה ובונה רשימה חדשה הממוינת בסדר עולה לפי האלף־בית האנגלי. רשימה זו תכיל את האיברים שנמצאים ברשימה הנתונה יותר מפעם אחת, וכל איבר כזה יופיע ברשימה החדשה פעם אחת בלבד.

המשתנה YAAD יצביע על הצומת הראשון ברשימה החדשה, כלומר יכיל את הכתובת של הצומת הראשון ברשימה (ראה איור ב').

בעבור הרשימה שבאיור א' קטע התכנית יבנה את הרשימה הבאה:



איור ב' לשאלה 6

שים לב: הרשימה החדשה ממוינת לפי האלף־בית האנגלי.

המשך בעמוד 37

לביצוע משימה זו קטע התכנית משתמש במחסנית.

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

- NUM – ערך של תו המאוחסן באוגר AX
- YAAD

השגרה יוצרת צומת חדש ומציבה בשדה ה־info של צומת זה את התו NUM ומוסיפה את הצומת הזה לסוף הרשימה החדשה אשר לראשה מצביע YAAD .

קטע התכנית משתמש במערך מונים INDEX בגודל 26 בתים כך שבאינדקס 0 תימצא השכיחות (מספר המופעים) של התו 'a' , באינדקס 1 תימצא השכיחות (מספר המופעים) של התו 'b' וכן הלאה.

בקטע התכנית הנתון חסרים **שבעה** ביטויים המסומנים במספרים בין סוגריים עגולים.
רשום במחברתך את מספרי הביטויים החסרים (1) – (7) בלבד, בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
 INDEX DB 26 DUP(0)
       MOV BX,LIST_P
       XOR CX,CX
       XOR AX,AX
AGAIN: MOV AL,[BX]
       SUB AL, _____(1)_____
       MOV SI,AX
       _____(2)_____
       CMP WORD PTR[BX+1],0
       JE NEXT
       _____(3)_____
       _____(4)_____
```

המשך בעמוד 38

```
NEXT:   MOV CX,26
        XOR DI,DI
GO:     CMP INDEX[DI],0
        JE CONT
        CMP INDEX[DI],1
        JE CONT
        MOV AX,DI
        ______(5)______
        PUSH AX
        PUSH YAAD
        ______(6)______
        JC CONT
        POP YAAD
        ADD SP,2
CONT:   ______(7)______
        LOOP GO
EXIT:
```

שאלה 7

א. נתון כי באוגר AX נמצא מספר שלם, חיובי וגדול מאפס.

לפניך קטע תכנית בשפת אסמבלי אשר מציב באוגר AX את הערך של AX*10 .

קטע התכנית אינו משתמש בפקודת MUL אלא בפקודות הזזה וחיבור.

הנח כי ערכו של AX*10 אינו עולה על 65535 .

בקטע התכנית הנתון חסר ביטוי **אחד** המסומן במספר בין סוגריים עגולים.
רשום במחברתך את מספר הביטוי החסר (1) בלבד, וכתוב לידו את הביטוי החסר שהוא מייצג.

```
MOV  BX,AX
```

המשך בעמוד 39

```
MOV CL,3
SHL   AX,CL
SHL BX,1
_____(1)_____
```

ב. לפניך תכנית בשפת אסמבלי שמוגדר בה מערך בשם STR1 .

כל תא במערך זה הוא בגודל בית (8 סיביות) ומכיל תו בתחום התווים '0'...'9' .

בתא האחרון של המערך נמצא זקיף שהוא המספר 0 .

התכנית תחזיר במשתנה NUM10 מספר עשרוני המיוצג באמצעות המערך STR1 .

דוגמה – בעבור המערך STR1 שלהלן:

'5'	'7'	'2'	'8'	0

התכנית תחזיר ב־NUM10 את המספר העשרוני 5728 .

להזכירך:

$$'5' - '0' = 5$$
$$'7' - '0' = 7$$

וכך הלאה.

תהליך חישוב הערך שישוכן ב- NUM10 בעבור המערך שבדוגמה הנו:

$$(((('5'-'0') \cdot 10 + ('7'-'0')) \cdot 10 + ('2'-'0')) \cdot 10 + ('8'-'0') =$$

$$= ((5 \cdot 10 + 7) \cdot 10 + 2) \cdot 10 + 8 = 5728$$

הנח כי הערך שישוכן ב־NUM10 אינו עולה על 65535 .

המשך בעמוד 40

בקטע התכנית הנתון חסרים **שמונה** ביטויים המסומנים במספרים בין סוגריים עגולים.
רשום במחברתך את מספרי הביטויים החסרים (1) – (8) בלבד, בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
  SSEG SEGMENT STACK 'STACK'
  DB 100H DUP()
  SSEG ENDS
  CODE SEGMENT
  ASSUME   CS:CODE
  STR1 DB '12345',0
  NUM10 DW 0

START: LEA SI,STR1
       XOR CX,CX
       MOV CL,CS:[SI]
       MOV NUM10,CX
       AND NUM10, _____(1)_____
NEXT:  _____(2)_____
       MOV CL,CS:[SI]
       _____(3)_____ CL,CL
       JZ SOF
       PUSH NUM10
       PUSH CX
       CALL MUL_BY_SHIFT_ADD
       _____(4)_____
       JMP NEXT
```

המשך בעמוד 41

```
SOF:   MOV AH,4CH
       INT 21H
MUL_BY_SHIFT_ADD:
       PUSH BP
       MOV BP,SP
       MOV AX,[BP+6]
       MOV BX,[BP+4]
       AND BL, ______(1)______
       ______(5)______
       SHL AX,CL
       MOV DX,[BP+6]
       ______(6)______
       ______(7)______
       ADD AX,BX
       ______(8)______
       POP BP
       RET 2
CODE ENDS
END START
```

המשך בעמוד 42

שאלה 8

לפניך קטע תכנית בשפת אסמבלי שבו מוגדרים המערך ARRAY – ב־CODE SEGMENT ומערך נוסף – במחסנית.

כל תא בכל אחד משני המערכים הוא בגודל מילה (16 סיביות), ומכיל תו בתחום התווים 'a'...'z' .

שני המערכים מסתיימים בזקיף 0 , כלומר, גם בתא האחרון של המערך ARRAY וגם בתחתית המחסנית נמצא הערך 0 .

```
SSEG SEGMENT STACK 'STACK
DB 100H DUP (0)
SSEG ENDS
CODE SEGMENT
ASSUME CS:CODE
ARRAY      DW 'a','x','b','e','a','y','b',0
COMMON     DB  26 DUP(0);
HOW_MANY   DB   0
```

פקודות כלשהן;

```
       XOR SI,SI           ; INDEX TO ARRAY
       XOR AX,AX
NEXT:  MOV AX,ARRAY[SI]
       OR AX,AX
       JE  CHECK_STACK
       SUB AL,'A'
       MOV BX,AX
       CMP COMMON[BX],0
       JNE CON
       MOV COMMON[BX],1
CON:   INC SI
       INC SI
```

המשך בעמוד 43

```
        INC SI
        JMP NEXT
CHECK_STACK:
        XOR AX,AX
AGAIN:  POP AX
        OR  AX,AX
        JE  HOW_MANY_COMMON
        SUB AL,'A'
        MOV SI,AX
        CMP COMMON[SI],0
        JE NOT_COMMON
        INC COMMON[SI]
NOT_COMMON:
        INC SI
        JMP AGAIN
HOW_MANY_COMMON:
        LEA BX,COMMON
        MOV CX,26
AG:     CMP BYTE PTR CS:[BX],1
        JBE GO
        INC HOW_MANY
GO:     INC BX
        LOOP AG
 SOF:       MOV AH,4CH
            INT 21H
            CODE ENDS
END START
```

המשך בעמוד 44

א. נתונים שני המערכים האלה:

המערך ARRAY :

ARRAY	'a'	'x'	'b'	'e'	'a'	'y'	'b'	0

והמערך הנוסף המשוכן במחסנית כדלהלן:

	כתובת	**תוכן**
	82	0'b'
	84	0'm'
	86	0'a'
	88	0'a'
	90	0'e'
	92	0'z'
	94	0'y'
	96	0'a'
	98	0
SP →	100	?

איור לשאלה 8

מהו הערך שיימצא במשתנה HOW_MANY לאחר הרצת קטע התכנית הנתון על שני המערכים האלה?

ב. אם המערך שמשוכן במחסנית מכיל את הערך 0 בלבד (כלומר המערך ריק), אז מהו הערך שיימצא במשתנה HOW_MANY לאחר הרצת קטע התכנית הנתון?

ג. אם נחליף בקטע התכנית הנתון את השורה JBE GO בשורה JLE GO - האם ביצועי קטע התכנית ישתנו?

ענה "כן" או "לא".

ד. אם נחליף בקטע התכנית הנתון את השורה LOOP AG בשורה LOOPE AG - האם ביצועי קטע התכנית ישתנו?

ענה "כן" או "לא".

המשך בעמוד 45

ה. עתה נניח שבכל אחד מן המערכים **בפני עצמו** אין איברים כפולים, כלומר האיברים שבתוך ARRAY שונים זה מזה, והאיברים שבמחסנית שונים זה מזה.

להלן חמישה היגדים שאחד מהם מתאר את הפעולה שמבצע קטע התכנית הנתון. רשום במחברתך את מספרו של ההיגד הנכון.

היגד 1: קטע התכנית מונה את מספר האיברים שמופיעים במערך ARRAY ולא מופיעים במחסנית.

היגד 2: קטע התכנית מונה את מספר האיברים שמופיעים במחסנית ולא מופיעים במערך ARRAY .

היגד 3: קטע התכנית מונה את מספר האיברים שמופיעים במחסנית או במערך ARRAY .

היגד 4: קטע התכנית מונה את מספר האיברים שמופיעים במחסנית או במערך ARRAY , אך אינם שייכים לשני המערכים גם יחד.

היגד 5: קטע התכנית מונה את מספר האיברים המשותפים שמופיעים בשני המערכים גם יחד.

בהצלחה!

נספח: מילון מונחים (2 עמודים)

לשאלון 714001, אביב תשע"ו

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

המשך בעמוד 2

המונח	תרגום המונח		
	ערבית	רוסית	אנגלית
שפת סף:			
אוגר	مخزن (ريغستر)	Регистр	register
אלכסון ראשי	مائل رئيسيّ	Главная диагональ	main diagonal
בסיס הקסאדצימלי	قاعدة الست عشريّة	Шестнадцатеричная система счисления, основание 16	hexadecimal base
מחסנית	باغة	Стек	stack
מספר בבסיס עשרוני	عدد عشريّ	десятичное число	decimal base number
מספר מסומן	عدد مُعَلَّم	обозначенное число, число со знаком	marked number
סיביות	البتات	биты	bits
פלינדרום	بولينديروم	Палиндром	palindrome
שגרה	رتابة	Функция, рутина	routine
שגרה רקורסיבית	إجراء تراجعيّ	Рекурсивная функция	recursive routine
תווית	نصّ	Метка	string