מדינת ישראל
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תשע"ז, 2017
סמל השאלון: 714001
נספחים: א. דף תשובות
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: **תכנות מערכות בשפת C** ו**שפת סף** ובהם שמונה שאלות. עליך לענות על **שש** שאלות, על־פי ההנחיות שבכל פרק.
בסך־הכול – 100 נקודות.

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

ד. 1. את התשובות לשאלות 1 ו־2 יש לרשום **אך ורק** על־גבי דף התשובות שבנספח א'.
את התשובות ליתר השאלות יש לרשום במחברת הבחינה.

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

בשאלון זה 44 עמודים ו־3 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 (30 נקודות)

ענה על שתי השאלות 1–2.

שאלה 1 – שאלת חובה (20 נקודות)

I. לפניך תכנית בשפת C שהפונקצייה main שלה מכילה הגדרה של משתנה בשם str .
במשתנה הזה תשוכן מחרוזת שאינה ריקה, המכילה אותיות גדולות בלבד, ללא רווחים ביניהן, ולכל אות יהיו תשעה מופעים לכל היותר במחרוזת.

בתכנית מוגדרת גם פונקצייה בשם func , אשר מקבלת כפרמטר את המחרוזת str , ומחזירה מחרוזת חדשה באופן הזה:
במחרוזת החדשה תופיע כל אות שהופיעה במחרוזת str , ומיד לאחריה – מספר המופעים של אותה אות במחרוזת str . לדוגמה: בעבור המחרוזת "ABCCBAXXX", המחרוזת החדשה שתוחזר תהיה "A2B2C2X3" .

המשך בעמוד 3

```
#include <stdio.h>
#include <stdlib.h>
char* func(const char* str)
{
	char i, size = 0, *p = NULL;
	char englishLetters[26] = {0};
	while (*str)
	{
		englishLetters[ _____(1)_____]++;
		str++;
	}
	for (i = 0; i < 26; i++)
	{
		if (englishLetters[i])
		{
			size =_____(2)_____;
			p = realloc(p,size*sizeof(char));
			if (!p){printf("Not enough memory!"); exit(1);}
			p[_____(3)_____] = _____(4)_____;
			p[_____(5)_____] = _____(6)_____;
		}
	 }
	p =_____(7)_____;
	if (!p){ printf ("Not enough memory!"); exit(2);}
	p[_____(8)_____] = _____(9)_____;
	_____(10)_____;
}
```

המשך בעמוד 4

```
void main()
{
	char str[] = "ABCCBAXXX";
	char* p = func(str);
	puts(p);
	free(p);                                    //שחרור המחרוזת הדינמית
}
```

בקטע התכנית הזה חסרים **עשרה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **א'–י'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

א. הביטוי החסר (1) הוא:

1. *str–'A'
2. *str–1
3. *str+'A'
4. *str

ב. הביטוי החסר (2) הוא:

1. size+2
2. size+1
3. size+i
4. size*2

המשך בעמוד 5

ג. הביטוי החסר (3) הוא:

1. size*2+1
2. size–2
3. size–1
4. size/2

ד. הביטוי החסר (4) הוא:

1. size
2. (char)('A'+i)
3. size+2
4. i

ה. הביטוי החסר (5) הוא:

1. size+1
2. size
3. size–1
4. i

ו. הביטוי החסר (6) הוא:

1. englishLetters[i+size]
2. englishLetters[i]
3. (char)(englishLetters[i]+'0')
4. englishLetters[i]+'A'

המשך בעמוד 6

ז. הביטוי החסר (7) הוא:

1. realloc(p, size*sizeof(char)+size)
2. realloc(p, – –size*sizeof(char))
3. realloc(p, size*sizeof(char))
4. realloc(p, ++size*sizeof(char))

ח. הביטוי החסר (8) הוא:

1. size+1
2. size
3. size*2
4. size–1

ט. הביטוי החסר (9) הוא:

1. –2
2. i
3. 0
4. size

י. הביטוי החסר (10) הוא:

1. return englishLetters
2. return str
3. return p
4. return &p

המשך בעמוד 7

II. משרד התיירות עורך בקרת איכות בשישה בתי־מלון הממוספרים מ־0 עד 5 (כולל).

בכל בית־מלון נבדקים חמישה סוגי שירותים, הממוספרים מ־0 עד 4 (כולל), ולכל שירות יש **משקל**, שאינו בהכרח שווה.

סוגי השירותים הם: שירות קבלה, שירות חדרים, בריכה, שירות הסעדה ומשחקייה, שיסומנו באותיות A–E בהתאמה.

לכל בית־מלון ניתן **דירוג** עבור כל אחד מחמשת סוגי השירותים שהוא מציע לאורחיו.

לכל בית־מלון מחושב **דירוגו הסופי** (שהוא ציון משוקלל המחושב על־פי הדירוגים שקיבל בעבור השירותים שהוא מציע לאורחיו).

לכל סוג של שירות מחושב **דירוגו הממוצע** בין בתי־המלון.

נסמן את **משקל** השירות ואת **דירוג** בית־המלון, כדלהלן:

$\mathbf{W_j}$ – **המשקל** הניתן לשירות מסוג j , כאשר $0 \le j \le 4$ וגם $\sum_{j=0}^{4} W_j = 100$

$\mathbf{M_{ij}}$ – **הדירוג** שקיבל בית־המלון ה־i , בשירות מסוג j , כאשר $0 \le i \le 5$, $0 \le j \le 4$ וגם $0 \le M_{ij} \le 10$

וכעת נציג את דרך חישובם של **הדירוג הסופי** של בית־המלון ה־i , ושל **הדירוג הממוצע** של השירות ה־j :

$\mathbf{S_i}$ – **הדירוג הסופי** של בית־המלון ה־i , והוא יחושב באמצעות הנוסחה שלהלן:

$$S_i = \left(\sum_{j=0}^{4} M_{ij} * W_j \right) / 100$$

$\mathbf{P_j}$ – **הדירוג הממוצע** של השירות מסוג j בין כל בתי־המלון, והוא יחושב באמצעות הנוסחה שלהלן:

$$P_j = \left(\sum_{i=0}^{5} M_{ij} \right) / 6$$

המשך בעמוד 8

תוצאות הבקרה נרשמות בבסיס נתונים ממוחשב, הכולל את המערכים וטיפוס הנתונים, שהגדרותיהם מובאות להלן:

```
#define NUM_OF_HOTELS  6
#define NUM_OF_SERVICES  5

typedef struct serviceType    //טיפוס סוג השירות
{
   char name[12];              //שם סוג השירות
   float avgPosition;         //דירוגו הממוצע של סוג השירות
}service,*servicePtr;
```

- `service services[NUM_OF_SERVICES]`

מערך השירותים.

- `char *names[NUM_OF_HOTELS] = {"DAN","SHERATON","HILTON","DANYEL","AKADIA","CARLTON"}`

מערך המכיל את שמות בתי־המלון.

- `char *serviceNames[NUM_OF_SERVICES]= {"A","B","C","D","E"}`

מערך המכיל את שמות סוגי השירותים.

- `int factors[NUM_OF_SERVICES] = {10,40,15,15,20};`

מערך המכיל את משקלי השירותים באחוזים.

לדוגמה: אם factors[2]=15 אז המשקל הניתן לשירות מספר 2 הוא 15 אחוז.

המשך בעמוד 9

- `int rateArr[NUM_OF_HOTELS][NUM_OF_SERVICES]`

  מערך דו־ממדי שבו משכנים בעבור כל בית־מלון את הדירוג שקיבל בעבור כל אחד מבין סוגי השירותים הניתנים בו.

  לדוגמה: אם rateArr[2][4] = 8 אז בית־המלון שמספרו 2 קיבל בעבור השירות שמספרו 4 את הדירוג 8 .

- `int firstRate[NUM_OF_SERVICES]`

  כאשר firstRate[j] - מכיל את מספר בית־המלון שקיבל את הדירוג הגבוה ביותר עבור סוג השירות שמספרו j , לכל $0 \leq j \leq 4$.

- `float hotelRates[NUM_OF_HOTELS]`

  מערך "הדירוג הסופי של בתי־המלון"

  כאשר hotelRates[i] יכיל את הדירוג הסופי של בית־המלון ה־i , לכל $0 \leq i \leq 5$.

המשך בעמוד 10

ענה על הסעיפים י"א–י"ג שלהלן:

לפניך קטע קוד המדפיס את פרטי כל אחד מבין חמשת סוגי השירותים, כאשר השירותים ממוינים בסדר עולה לפי שדה "הדירוג הממוצע" שלהם.

בקטע הקוד מזמנים פונקצייה בשם intcmp אשר מוצגת לאחריו.

בקטע הקוד הנ"ל חסרים **שלושה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **י"א–י"ג** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
/* Init services*/
for (i = 0; i < NUM_OF_SERVICES; i++)
{
      strcpy(services[i].name , serviceNames[i]);
      services[i].avgPosition = 0;
}
/*averages per service */
for (j = 0; j < NUM_OF_SERVICES; j++)
{
  for (i = 0; i < NUM_OF_HOTELS; i++)
      services[j].avgPosition += (float)rateArr[i][j];
   _________ (1)_________;
}
qsort(_____________(2)_____________);
      printf("\n Category  \t Average ");
for (j = 0; j < NUM_OF_SERVICES ; j++)
   printf("\n%s \t \t%5.2f", services[j].name,services[j].avgPosition);
```

המשך בעמוד 11

```
int intcmp(_____________(3)_____________)
{
  servicePtr p = v1;
  servicePtr q = v2;
  return (int)(p->avgPosition - q->avgPosition) ;
}
```

י"א. הביטוי החסר (1) הוא:

1. services[j].avgPosition /= NUM_OF_HOTELS
2. services[j].avgPosition += services [j].average
3. services[j].avgPosition += NUM_OF_HOTELS
4. avgPosition /= NUM_OF_HOTELS

י"ב. הביטוי החסר (2) הוא:

1. services, NUM_OF_SERVICES, 5, intcmp
2. services, NUM_OF_SERVICES, sizeof(services [0]), intcmp
3. &services, NUM_OF_SERVICES, 5, intcmp
4. services, NUM_OF_HOTELS, sizeof(services [0]), intcmp

י"ג. הביטוי החסר (3) הוא:

1. char *v1, char *v2
2. const void **v1, const void **v2
3. const void *v1, const void *v2
4. servicePtr *v1, servicePtr *v2

המשך בעמוד 12

לפניך קטע קוד המחשב את דירוגו הסופי של כל בית־מלון, מאתר את בית־המלון **שהדירוג הסופי** שלו הוא הגבוה ביותר, ומדפיס את שם בית־המלון הזה ואת הדירוג הסופי שלו.

הנחה: קיים רק בית־מלון אחד שדירוגו הסופי הוא הגבוה ביותר.
כל תא במערך hotelRates מאותחל לאפס.

בקטע התכנית הזה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **י"ד–י"ז** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
for(i = 0; i < NUM_OF_HOTELS; i++)
 {
     for (j = 0; j < NUM_OF_SERVICES; j++)
             hotelRates[i] += _____(1)_____;
     hotelRates[i] /= 100;
 }
bestHotel = 0;
for(i = 1; i < NUM_OF_HOTELS; i++)
      bestHotel = (_____(2)_____)? i : bestHotel;
printf("\n\n The best hotel is %s with grade %5.2f",
                _____(3)_____ ,_____(4)_____);
```

י"ד. הביטוי החסר (1) הוא:

1. (float) rateArr [i][j] – factors [j]
2. (float) rateArr [i][j] / factors [j]
3. (float) rateArr [i][j] + factors [j]
4. (float)rateArr[i][j] * factors[j]

המשך בעמוד 13

ט"ו. הביטוי החסר (2) הוא:

1. hotelRates[i] < hotelRates[bestHotel]
2. hotelRates[i] > hotelRates[bestHotel–1]
3. hotelRates[i] > hotelRates[bestHotel]
4. hotelRates[i] > hotelRates[bestHotel+1]

ט"ז. הביטוי החסר (3) הוא:

1. bestHotel
2. names[bestHotel]
3. hotelRates[bestHotel]
4. names[0]

י"ז. הביטוי החסר (4) הוא:

1. hotelRates[bestHotel]
2. names[bestHotel]
3. bestHotel
4. hotelRates[NUM_OF_HOTELS]

לפניך קטע קוד המדפיס עבור כל שירות את שמו ואת שם בית־המלון שקיבל בו את הדירוג הגבוה ביותר.

קטע הקוד בונה את המערך firstRate .

הנחה: עבור כל שירות, קיים רק בית מלון אחד שקיבל בו את הציון הגבוה ביותר.

בקטע התכנית הזה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **י"ח–כ"א** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

המשך בעמוד 14

```
for(j=0; j < NUM_OF_SERVICES; j++)
  {
    bestQ =0;
    for (i=0; i< _____________(1)_______; i++)
            bestQ = (rateArr[i][j] > _______(2)____)? i : bestQ;
    __________(3)_____________;
  }

  for (i=0; i < NUM_OF_SERVICES; i++)
  printf("\n\n Best quailty %s has the Hotel %s",
             services[i].name, _________(4)_________);
  getchar();
  return 0 ;
```

י"ח. הביטוי החסר (1) הוא:

1. j
2. NUM_OF_SERVICES
3. bestQ
4. NUM_OF_HOTELS

י"ט. הביטוי החסר (2) הוא:

1. firstRate[j]
2. rateArr[i–1][j]
3. rateArr[bestQ][j]
4. rateArr[i][j–1]

המשך בעמוד 15

כ. הביטוי החסר (3) הוא:

1. bestQ = firstRate[i]
2. firstRate[j] = bestQ
3. bestQ = firstRate[j]
4. firstRate[i] = bestQ

כ"א. הביטוי החסר (4) הוא:

1. names[firstRate[i]]
2. names[i]
3. firstRate[i]
4. names[bestQ]

המשך בעמוד 16

שאלה 2 – שאלת חובה (10 נקודות)

I. לפניך הגדרה חדשה:

$b = (b_0, b_1, \ldots, b_i, b_{i+1}, b_{i+2}, \ldots, b_{n-1})$ הוא מערך הממוין בסדר עולה, שאין בו איברים כפולים, כלומר: $b_0 < b_1 < \ldots b_i < b_{i+1} < b_{i+2} < \ldots < b_{n-1}$.

מערך a ייקרא **"מערך ממוין מוסט"** כאשר המערך a מתקבל מן המערך הממוין b , לאחר שכל איבר במערך b הוסט ממקומו i פעמים שמאלה בצורה מעגלית, כאשר $0 < i < n-1$, כלומר: $a = (b_i, b_{i+1}, \ldots, b_{n-1}, b_0, b_1, b_2, \ldots, b_{i-1})$.

דוגמה:

עבור המערך הממוין הזה:

	0	1	2	3	4
b =	1	2	3	4	5

ה"מערך הממוין המוסט" בעבור i=2 , ייראה כך:

	0	1	2	3	4
a =	3	4	5	1	2

נוסף על כך, להלן פונקצייה המממשת את החיפוש הבינארי. תוכל להיעזר בה.

```
int binarySearch(int a[], int n, int x)
{
   int l, h, m;
   l = 0;
   h = n-1;
   while(l <= h)
   {
      m = (l+h) / 2;
      if (a[m] == x) return m;
      if (a[m] < x) l = m+1;
      else h = m-1;
   }
   return -1;
}
```

המשך בעמוד 17

לפניך פונקצייה שכותרתה:

```
int findShifted(int arr[], int n, int x)
```

פונקצייה זו מקבלת שלושה פרמטרים:

arr – מערך ממוין מוסט בגודל n , המכיל מספרים השונים זה מזה

n – גודל המערך

x – מספר נוסף כלשהו

אם הערך x קיים במערך arr אזי הפונקצייה תחזיר את מיקומו במערך.

אחרת – היא תחזיר את הערך (1–) .

בפונקצייה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **א'–ה'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
int findShifted(int arr[], int n, int x)
{
   int low = 0,high = n-1,mid = 0;
   int index;
          // arr[mid] < arr[mid-1] מציאת מקום במערך כך שיתקיים:
   while (low <= high)
   {
           mid = (low+high) / 2;
           if(arr[n-1] <= arr[mid])
               low = mid+1;
           else if (arr[0] >= arr[mid])
                      high = mid;
           if (_______(1)_____) break ;
   }
   index =_____________(2)___________;
   if(_________(3)______) return index;
   index = binarySearch(_____________(4)_________);
   return(index == -1)? -1 :_________(5)_______;
}
```

המשך בעמוד 18

א. הביטוי החסר (1) הוא:

1. arr[mid] == x
2. low == 0 && mid == high
3. low == mid
4. arr[mid] < arr[mid–1]

ב. הביטוי החסר (2) הוא:

1. low
2. mid
3. binarySearch(arr+mid, n–mid+1, x)
4. binarySearch(arr, mid, x)

ג. הביטוי החסר (3) הוא:

1. low <= high
2. arr[mid] == x
3. index != –1
4. arr[index] == x

ד. הביטוי החסר (4) הוא:

1. arr, n, x
2. arr+index, n–index, x
3. arr+mid, n–mid, x
4. arr, mid, x

המשך בעמוד 19

ה. הביטוי החסר (5) הוא:

1. index
2. index+mid
3. (arr[mid] < x)? index : mid+index
4. (arr[0] > x)? mid+index : index

II. לפניך פונקצייה שכותרתה:

```
void sortPartialSorted(int a[], int n, const int k)
```

פונקצייה זו מקבלת שלושה פרמטרים:

a – מערך בגודל n , המכיל מספרים שלמים

n – גודל המערך

k – מיקום כלשהו על־פני המערך

ידוע כי כל המספרים במערך, החל מן המיקום ה־k ואילך, ממוינים בסדר עולה, וכי שאר המספרים במערך, החל מן המיקום ה־0 ועד למיקום ה־k-1 , לא בהכרח ממוינים.

הפונקצייה תמיין את המערך a בסדר עולה.

שים לב :

- יש לסרוק את המערך פעם אחת בלבד.
- המספרים שנמצאים עד המיקום ה־k-1 , כולל אותו, יכולים להיות גדולים מן המספרים שנמצאים **לאחר** המיקום ה־k-1.

דוגמה:

בעבור המערך a הזה:

0	1	2	3	4	5	6	7	8	9
17	1	23	1	4	6	9	20	25	30

וכן בעבור n = 10 ו־ k = 3 ,
הפונקצייה תחזיר את המערך שלהלן:

0	1	2	3	4	5	6	7	8	9
1	1	4	6	9	17	20	23	25	30

המשך בעמוד 20

בארבע הפונקציות הבאות **יחד** חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **ו'–י'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

הערה: ביטוי להשלמה יכול לכלול כמה ביטויים להשלמה, כגון: "a, right".

```
void sortPartialSorted(int a[], int n, const int k)
{
   int i ;
   int *helper = malloc(sizeof(int)* k);
   for (i=0; i<k; i++) helper[i] = a[i];
   my_sort(helper, k);
   merge(___(1)___ , _____(2)____ , helper, k, a);
}

void my_sort(int a[], int n)
{
   int *tmp_array = malloc(sizeof(int) * n);
   internal_sort(a, n, tmp_array);
   free(tmp_array);
}

void internal_sort(int a[], int n, int helper_array[])
{
   int i;
   int left = n/2, right = n-left;
   if (n < 2) return;
   internal_sort(____(3)_____, helper_array);
```

המשך בעמוד 21

```
   internal_sort(_______(4)_______, helper_array);
   merge(_______(3)_____, _____(4)_____, helper_array);
   for(i = 0; i < n; i++) a[i] = helper_array[i];
}
void merge(int a[], int na, int b[], int nb, int c[])    //  פעולת מיזוג
{
   int ia, ib, ic;
   for(ia = ib = ic = 0; (___________(5)_____); ic++)
   {
            if(a[ia] < b[ib])
            {
                     c[ic] = a[ia];
                     ia++;
            }
            else
            {
                     c[ic] = b[ib];
                     ib++;
            }
   }
   for(; ia < na; ia++, ic++) c[ic] = a[ia];
   for(; ib < nb; ib++, ic++) c[ic] = b[ib];
}
```

המשך בעמוד 22

ו. הביטוי החסר (1) הוא:

1. a
2. *a[k]
3. a+k+1
4. a+k

ז. הביטוי החסר (2) הוא:

1. n
2. 1
3. k
4. n–k

ח. הביטוי החסר (3) הוא:

1. a+left, right–left
2. a+right, left
3. a, left
4. *a[left], right

ט. הביטוי החסר (4) הוא:

1. *a[left], right
2. a+left, right–left
3. a, right
4. a+right, left

המשך בעמוד 23

ג. הביטוי החסר (5) הוא:

1. ic < na+nb
2. (ia < na) && (ib < nb)
3. (ic < na) || (ic < nb)
4. (ic != na) && (ic != nb)

<u>פרק שני</u> (20 נקודות)

ענה על <u>אחת</u> מבין השאלות 3-4 (לכל שאלה – 20 נקודות).

שאלה 3

א. רשום במחברתך את הפלט המדויק של התכנית שלהלן:

```
#include <stdio.h>
char *c[] = {"ENTER", "NEW", "POINT", "FIRST"};
char **cp[] = {c+3, c+2, c+1, c};
char ***cpp=cp;
void main()
{
      printf("%s\n", **++cpp);
      printf("%s\n", *--*++cpp+3);
      printf("%s\n", *cpp[-2]+3);
      printf("%s\n", cpp[-1][-1]+1);
      getchar();
}
```

המשך בעמוד 24

ב. רשום במחברתך את הפלט המדויק של קטע הקוד שלהלן.

```
c=0x6^0x5;
printf("%d\n",c);
```

ג. לפניך הגדרה של צומת ברשימה מקושרת חד־כיוונית בשפת C :

```
typedef  struct  nodeType
{
   int  num;
   struct   nodeType   *next;
} nodeRec,  *nodePtr;
```

נתונות שתי רשימות מקושרות חד־כיווניות, לא בהכרח זהות באורכן, שלראשן מצביעים LIST1 ו־LIST2 .

לפניך פונקצייה **רקורסיבית** אשר יוצרת משתי הרשימות הנתונות רשימה חדשה, באופן הזה:

האיבר הראשון ברשימה החדשה יהיה האיבר הראשון ברשימה **LIST1** .

האיבר השני ברשימה החדשה יהיה האיבר הראשון ברשימה **LIST2** .

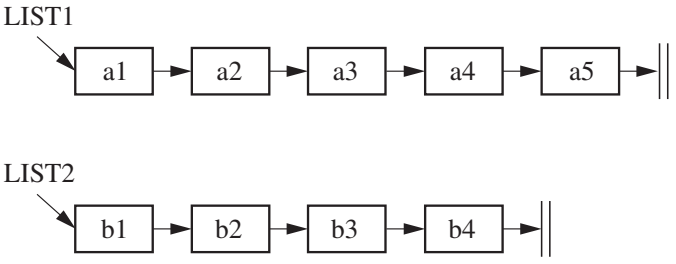
האיבר השלישי ברשימה החדשה יהיה האיבר השני ברשימה **LIST1** .

האיבר הרביעי ברשימה החדשה יהיה האיבר השני ברשימה **LIST2** .
וכן הלאה.

אם אחת משתי הרשימות קצרה יותר באורכה וסריקתה מסתיימת קודם, הפונקצייה תשרשר את זנב הרשימה הארוכה יותר שנותרה לסוף הרשימה החדשה.

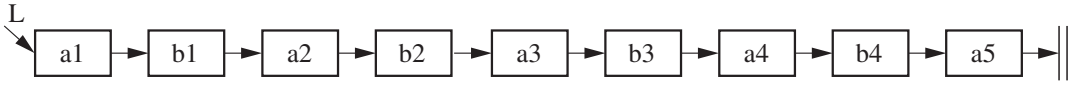
המשך בעמוד 25

דוגמה: בעבור שתי הרשימות באיור א שלהלן:



איור א' לשאלה 3

הפונקצייה תיצור ותחזיר את הרשימה הזו:



איור ב' לשאלה 3

המשך בעמוד 26

בפונקצייה 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.

```
nodePtr mergeLists(nodePtr List1, nodePtr List2)
 {
        nodePtr p,q,L = NULL;
        if(List1 == NULL && List2==NULL) return NULL;
        if( List1 == NULL)
        {
                L = mergeLists(__________(1)__________);
                p = malloc(sizeof(nodeRec));
                p->num = List2->num;
                p->next = L;
                ____(2)___= p;
        }
        else if (List2 == NULL)
             {
                L = mergeLists(_________(3)_______);
                q = malloc(sizeof(nodeRec));
                q->num = List1->num;
                q->next = L;
                ____(2)___= q;
             }
   else
```

המשך בעמוד 27

```
    {
                L = mergeLists(_________(4)_______);
                p = malloc(sizeof(nodeRec));
                p->num =_________(5)________;
                p->next = L;
                _____(2)___= p;
                q = malloc(sizeof(nodeRec));
                q->num =_________(6)________;
                q->next = L;
                _____(2)___= q;
    }
    return  L;
}
```

שאלה 4

א. לפניך פונקצייה שכותרתה:

```
void searchAndReplace(char *str, char *search, char *replace)
```

פונקצייה זו מקבלת שלוש מחרוזות: str , search ו־replace .

האורך של המחרוזות search , ו־replace זהה, ונוסף על כך, בכל אחת מן המחרוזות search ו־replace כל התווים שונים זה מזה.

אם התו שבמקום ה־j במחרוזת str מופיע במקום ה־i במחרוזת search , אז ב־str[j] יוצב התו שב־replace[i] .

דוגמה:

בעבור המחרוזות: "Good luck in the exam" , search = "oe" , replace = "yk" –
לאחר זימון הפונקצייה, המחרוזת המשוכנת ב־str תהיה: "Gyyd luck in thk kxam".

המשך בעמוד 28

שים לב:

המחרוזות str ו-search נסרקות פעם אחת **בלבד**.

בפונקצייה 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.

```
#define N 128

typedef enum {FALSE,TRUE} boolean;

void searchAndReplace(char *str, char *search, char *replace)

{

   int dict[N] = {0};

   while (*search)

   {

      dict[*search] =_______(1)______;

      search++;

      replace++;

   }

   while(______(2)_______)

   {

      if(______(3)______)  *str =______(4)______;

      str++ ;

   }

}
```

המשך בעמוד 29

ב. לפניך פונקצייה שכותרתה:

```
int stringStartsWith(char *arr[], int n, char *str)
```

פונקצייה זו מקבלת את arr – מערך מצביעים למחרוזות שגודלו n , אשר המחרוזות בו ממוינות בסדר לקסיקוגרפי, מכילות אותיות קטנות בלבד ואינן ריקות.

נוסף על כך, הפונקצייה מקבלת מחרוזת בשם str , שמכילה גם היא אותיות קטנות בלבד ואינה ריקה. הפונקצייה תחזיר את כמות המחרוזות ב־arr שמתחילות במחרוזת str .
אם ב־arr אין אף מחרוזת כזו אזי הפונקצייה תחזיר את הערך 0 .

דוגמה:
עבור המערך הזה:

```
char *arr[]={"a","aba","and","array","bad","banana","goat",
"good","grade","zebra"}
```

אם str="a" אז הפונקצייה תחזיר את הערך 4 כיוון שארבע המחרוזות "a","aba","and","array" , שנמצאות ב־arr מתחילות ב־"a" .

אם str="go" אז הפונקצייה תחזיר את הערך 2 כיוון ששתי המחרוזות"goat","good" שנמצאות ב־arr מתחילות ב־"go" .

אם str="gg" אז הפונקצייה תחזיר את הערך 0 כיוון שאין אף מחרוזת ב־arr שמתחילה ב־"gg" .

שים לב:

פונקצייה זו נעזרת:

1. ברעיון של החיפוש הבינארי
2. בפונקצייה שכותרתה:

```
int strcomp(char *s, char *init)
```

פונקצייה זו מקבלת שתי מחרוזות: s ו־init . הפונקצייה בודקת אם המחרוזת init מופיעה בתחילת המחרוזת s ומחזירה אחד מבין הערכים האלה:

אפס – אם המחרוזת init מופיעה בתחילת המחרוזת s .

מספר שלילי – אם המחרוזת s קטנה יותר בסדר הלקסיקוגרפי מן המחרוזת init.

מספר חיובי שונה מאפס – אם המחרוזת s גדולה יותר בסדר הלקסיקוגרפי מן המחרוזת init .

המשך בעמוד 30

בפונקצייה 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.

```
int stringStartsWith(char *arr[], int n, char *str)
{
        int first, last, L=0, R = n-1, m;
        while(L <= R)
        {
                m = (L+R)/2;
                if(______(1)_____) R = m-1 ;
                 else L = m+1;
        }
        first =_____(2)____;
        R = n-1;
        while(L <= R)
        {
                m = (L+R)/2;
                if (_______(3)______)  R = m-1;
                 else L = m+1;
        }
        last =_______(4)______;
        return________(5)_______;
}
```

המשך בעמוד 31

ג. רשום במחברתך את הפלט המדויק של התכנית שלהלן:

```
#include <stdio.h>
int a[3][3] = {{1, 2, 3},{1, 2, 3},{7, 8, 9}};
int *pa[] = {*a, *(a+1), a[2]};
int *p = a[0];
void main()
{
      int i = 2;
      printf("%d %d %d\n", a[i][2-i], *a[i], *(*(a+i)+i));
      for(i = 0; i < 3; i++)
            printf("%d  %d\n", *pa[i], p[i]);
      getchar();
}
```

המשך בעמוד 32

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 RECUR , אשר מקבלת באמצעות מחסנית את הארגומנט NUM, שהוא מספר עשרוני שלם חיובי הגדול מאפס, ובאמצעות האוגר DI את הארגומנט DIGIT , שהוא ספרה עשרונית.

הנחת יסוד: ערכו של המספר שמשוכן במשתנה NUM הוא 65535 לכל היותר.

```
SSEG SEGMENT STACK 'STACK'
  DB 100 DUP()
SSEG ENDS
DSEG SEGMENT
NUM DW 3234
DIGIT DW 3
DSEG ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DSEG
RECUR: PUSH BP
       MOV  BP,SP
       XOR  DX,DX
       MOV  AX,[BP+4]
       DIV  SI
```

המשך בעמוד 33

```
        CMP  DX,DI
        JNE  CONT
        OR   CL,BL
CONT:   SHL  BL,1
        OR   AX,AX
        JZ   BACK
        PUSH AX
        CALL RECUR
BACK:   POP  BP
        RET  2
START:  MOV  AX,DSEG
        MOV  DS,AX
        MOV  DI,DIGIT
        MOV  SI,10
        XOR  CL,CL
        MOV  BL,1
        PUSH NUM
        CALL RECUR
SOF:    MOV  AH,4CH
        INT  21H
CODE ENDS
END START
```

המשך בעמוד 34

לפניך תשעה סעיפים **א'–ט'** שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. מה יהיה תוכנו של האוגר CL , בבסיס בינארי, לאחר ביצוע התכנית הנתונה?

ב. אם נחליף בתכנית הנתונה את השורה **NUM DW 3234** בשורה **NUM DW 1234**, ואת השורה **DIGIT DW 3** בשורה **DIGIT DW 5**, מה יהיה תוכנו של האוגר CL, בבסיס בינארי, לאחר ביצוע התכנית?

ג. אם נחליף בתכנית הנתונה את השורה **NUM DW 3234** בשורה **NUM DW 3313** , מה יהיה תוכנו של האוגר CL , בבסיס בינארי, לאחר ביצוע התכנית?

ד. אם נחליף בתכנית הנתונה את השורה **NUM DW 3234** בשורה **NUM DW 0** , ואת השורה **DIGIT DW 3** בשורה **DIGIT DW 0** , מה יהיה תוכנו של האוגר CL , בבסיס בינארי, לאחר ביצוע התכנית?

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

- **היגד 1:** השגרה הרקורסיבית מחזירה באוגר CL את מספר המופעים של הספרה DIGIT במספר NUM , לדוגמה: אם NUM=12 ו־ DIGIT=5 , אז הייצוג הבינארי שמוחזר באוגר CL הוא 0000 0000 , כיוון ש־5 מופיע ב־ NUM 0 פעמים.
- **היגד 2:** השגרה הרקורסיבית מחזירה באוגר CL את תוצאת החישוב: NUM AND DIGIT , לדוגמה: אם NUM=25 ו־ DIGIT=5 , אז הייצוג הבינארי שמוחזר באוגר CL הוא 0000 0001 .
- **היגד 3:** השגרה הרקורסיבית מציבה באוגר CL את הערך 1 בכל המקומות שבהם מופיע DIGIT ב־ NUM , לדוגמה: אם NUM=3646 ו־ DIGIT=6 , אז הייצוג הבינארי שמוחזר באוגר CL הוא 0000 0101 .
- **היגד 4:** השגרה הרקורסיבית מחזירה באוגר CX מספר חדש שנוצר לאחר השמטתו של DIGIT מהמספר NUM , לדוגמה: אם NUM=5154 ו־ DIGIT=5 , אז המספר החדש שיתקבל באוגר CL הוא הייצוג הבינארי של 14 .

ו. האם השגרה הרקורסיבית הנתונה מתייחסת למספר העשרוני שב־NUM כאל מספר לא מסומן? ענה "כן" או "לא".

ז. אם נחליף בתכנית הנתונה את השורה **DIV SI** בשורה **IDIV SI** , האם שינוי זה ישפיע **בהכרח** על הערך המוחזר באוגר AL ? ענה "כן" או "לא".

המשך בעמוד 35

ח. אם נחליף בתכנית הנתונה את השורה **DIGIT DW 3** בשורה **DIGIT DB 3** , האם התכנית תעבור קומפילציה (הידור)? ענה "כן" או "לא".

ט. אם נחליף בתכנית הנתונה את השורה **RET 2** בשתי השורות האלה:

RET

RET

האם ביצועי התכנית ישתנו **בהכרח**? ענה "כן" או "לא".

<u>פרק רביעי</u> (30 נקודות)

ענה על <u>שתיים</u> מבין השאלות 6–8 (לכל שאלה – 15 נקודות).

שאלה 6

לפניך תכנית בשפת אסמבלי שמוגדר בה מערך בשם ARR:

```
SSEG SEGMENT STACK 'STACK'
 DB   100 DUP()
SSEG  ENDS
DSEG  SEGMENT
 ARR DB 200,100,-3,-80,128,-128
DSEG  ENDS
CODE SEGMENT
 ASSUME CS:CODE,DS:DSEG
START:
      MOV  AX,DSEG
      MOV  DS,AX
NEXT: MOV  CX,6-1
      XOR  DL,DL
      LEA  BX,ARR
```

המשך בעמוד 36

```
AG:    MOV  AL,[BX]
       CMP  AL,[BX+1]
       JLE  CONT
       XCHG AL,[BX+1]
       MOV  [BX],AL
       INC  DL
CONT:  INC  BX
       LOOP AG
       OR   DL,DL
       JNE  NEXT
SOF:   XOR  DH,DH
       MOV  CX,6
       XOR  BX,BX
SUM:   ADD  DH,[BX]
       INC  BX
       LOOP SUM
EXIT:  MOV  AH,4CH
       INT  21H
CODE ENDS
END START
```

א. כתוב במחברת הבחינה את אברי המערך ARR בבסיס 10 , משמאל לימין, כאשר התכנית מגיעה לתווית SOF .

ב. מה יהיה תכנו של CH בבסיס 16 כאשר התכנית מגיעה לתווית EXIT ?

ג. בהתייחס אל איברי המערך ARR וכן אל DH כאל מספרים עשרוניים מסומנים, האם ערכו של DH יהיה שווה לסכום אברי המערך ARR בהגיע התכנית לתווית EXIT ? ענה "כן" או "לא".

ד. אם נחליף בתכנית הנתונה את השורה **ARR DB 200,100, – 3, – 80,128, –128** בשורה **ARR DB – 56,100,253, – 80,128, – 128** , האם שינוי זה ישפיע על ביצועי התכנית? ענה "כן" או "לא".

ה. אם נחליף בתכנית הנתונה את השורה **JLE CONT** בשורה **JBE CONT** , האם שינוי זה ישפיע על ביצועי התכנית? ענה "כן" או "לא".

המשך בעמוד 37

שאלה 7

לפניך תכנית בשפת אסמבלי שמוגדרים בה שלושה מערכים: STR1, INDX1, STR2 .

הנח כי כל אחד מן המערכים הללו הוא בגודל של 5 בתים.

אם לכל i , כאשר $0 \le i \le 4$, מתקיים: $STR1[INDX1[i]] = STR2[i]$, אז התכנית מציבה באוגר DX את הערך 1 . אחרת - היא מציבה בו את הערך 0 .

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

```
     SSEG SEGMENT STACK 'STACK'
       DB     100 DUP()
     SSEG    ENDS
     DSEG    SEGMENT
       STR1  DB  "XYWAB"
       INDX1 DB  3,0,1,3,3
       STR2  DB  "YXYAA"
       P     DW STR1,INDX1,STR2
DSEG ENDS
CODE SEGMENT
       ASSUME CS:CODE,DS:DSEG
       COMP:
           PUSH   BP
           MOV    BP,SP
           MOV    BX,[BP+8]
           MOV    SI,[BP+6]
           MOV    DI,[BP+4]
```

המשך בעמוד 38

```
        MOV     DX,1
        MOV     CX,5
AGAIN:
        MOV     AL,[SI]
        _____(1)_____
        CMP     AL,_____(2)_____
        JE      NEXT
        _____(3)_____
        _____(4)_____
NEXT:   INC     SI
        _____(5)_____
        LOOP    AGAIN
EXIT:   POP     BP
        RET     6
START:
        MOV     AX,DSEG
        MOV     DS,AX
        PUSH    P
        PUSH    P+2
        PUSH    P+4
        CALL    COMP
        MOV     AH,4CH
        INT     21H
CODE ENDS
END START
```

המשך בעמוד 39

שאלה 8

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

info – שדה מידע שגודלו 8 סיביות, המכיל תו.

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

המצביע next בצומת האחרון הוא 0 .

א. לפניך שגרה בשם CALLEE .

לאחר הקריאה לשגרה CALLEE ולאחר ביצוע הפקודות:

```
PUSH   BP
MOV    BP,SP
```

המחסנית תיראה כך:

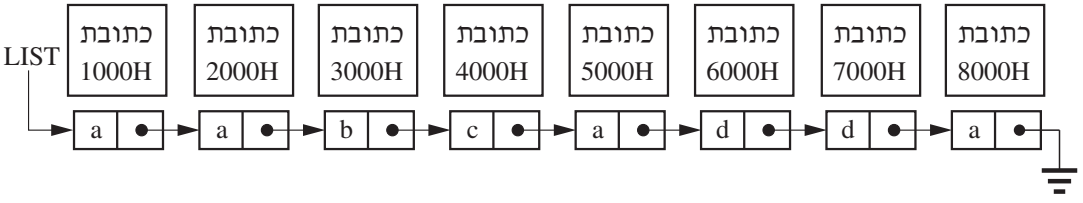
תוכן המחסנית	כתובת
•	•
•	•
•	•
•	•
ערכו הקודם של bp	[bp]
כתובת חזרה	[bp+2]
תו	[bp+4]
כתובת של צומת ברשימה קיימת	[bp+6]

השגרה מחפשת את התו שנמצא ב־[bp+4] , ברשימה נתונה, החל מן הצומת שכתובתו נמצאת ב־[bp+6] . אם החיפוש הסתיים בהצלחה – השגרה מחזירה ב־[bp+6] את כתובתו של הצומת שבו נמצא לראשונה התו המבוקש, אחרת – השגרה מחזירה ב־[bp+6] את הערך NULL .

המשך בעמוד 40

דוגמה:

בעבור הרשימה שלהלן (כאשר המספר בבסיס 16 המופיע מעל כל צומת מציין את כתובתו של הצומת):



איור א' לשאלה 8

השגרה תחזיר ב- [bp+6] את הערך NULL בעבור תמונת המחסנית הזו:

תוכן המחסנית	כתובת
.	.
.	.
.	.
.	.
ערכו הקודם של bp	[bp]
כתובת חזרה	[bp+2]
'b'	[bp+4]
4000H	[bp+6]

והשגרה תחזיר ב- [bp+6] את הערך 5000H בעבור תמונת המחסנית הזו:

תוכן המחסנית	כתובת
.	.
.	.
.	.
.	.
ערכו הקודם של bp	[bp]
כתובת חזרה	[bp+2]
'a'	[bp+4]
3000H	[bp+6]

המשך בעמוד 41

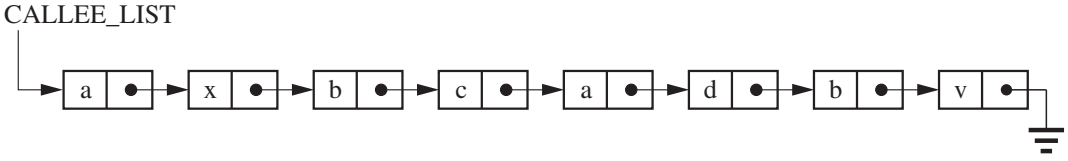
להלן השגרה בשם CALLEE .

בשגרה זו חסרים **ארבעה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחברתך את מספרי הביטויים החסרים (1) – (4) בלבד, בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

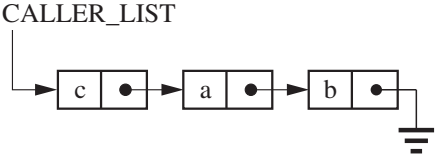
```
CALLEE:
    PUSH   BP
    MOV    BP,SP
    MOV    AL,[BP+4]
    MOV    BX,[BP+6]
COMP:
    _______(1)_____
    JNE    CONT
    _______(2)_____
    JMP    EXIT
CONT:
    CMP    WORD PTR[BX+1],NULL
    _____(3)_____
    MOV    _____(4)_____ ,NULL
    JMP    EXIT
NEXT:MOV   BX,[BX+1]
     JMP   COMP
EXIT:POP   BP
     RET   2
```

המשך בעמוד 42

ב. נתונים שני המשתנים CALLEE_LIST ו־ CALLER_LIST שהם מצביעים לראשן של שתי רשימות בלתי תלויות זו בזו. CALLEE_LIST מכיל את הכתובת של הצומת הראשון ברשימה הראשונה (ראה איור ב'), ו־ CALLER_LIST מכיל את הכתובת של הצומת הראשון ברשימה השנייה (ראה איור ג').



איור ב' לשאלה 8



איור ג' לשאלה 8

להלן הגדרה:

נאמר שרשימה L1 היא **תת־רשימה** של רשימה L2 , אם מתקיימים שני התנאים שלהלן:

1. כל איבר x ברשימה L1 מופיע גם ברשימה L2.
2. סדר האיברים זהה בשתי הרשימות.

לדוגמה: 3, 12, 6, 32, 1 היא תת־רשימה של: 6, **3**, 34, 1, 1, **12**, **6**, 8, **32**, 5, **1**, 13.

שים לב: האיברים שברשימה L1 אינם צריכים להופיע ברצף בתוך הרשימה L2 .

המטרה היא לכתוב שגרה הבודקת אם הרשימה שלראשה מצביע CALLER_LIST היא **תת־רשימה** של הרשימה שלראשה מצביע CALLEE_LIST .

המשך בעמוד 43

לפניך שגרה בשם CALLER אשר מקבלת באמצעות מחסנית שני ארגומנטים כמתואר בתמונת המחסנית הבאה:

תוכן המחסנית	
.	.
.	.
.	.
.	.
ערכו הקודם של bp	[bp]
כתובת חזרה	[bp+2]
כתובתו של הצומת הראשון ברשימה CALLER_LIST	[bp+4]
כתובתו של הצומת הראשון ברשימה CALLEE_LIST	[bp+6]

אם הרשימה שלראשה מצביע CALLER_LIST היא **תת־רשימה** של הרשימה שלראשה מצביע CALLEE_LIST , אז השגרה מחזירה בדגל הנשא את הערך 1 , אחרת - היא מחזירה בו את הערך 0 .

הערה: בעבור שתי הרשימות שבאיורים ב' ו-ג', השגרה CALLER תחזיר בדגל הנשא את הערך 1 , כיוון שהרשימה שלראשה מצביע CALLER_LIST היא **תת־רשימה** של הרשימה שלראשה מצביע CALLEE_LIST .

להלן שגרה בשם CALLER אשר נעזרת בשגרה CALLEE הנתונה בסעיף הקודם.
בשגרה זו חסרים **שישה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום במחברתך את מספרי הביטויים החסרים (1) – (6) בלבד, בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
LIS_P    DW CALLER_LIST,CALLEE_LIST
CALLER:
    PUSH  BP
    MOV   BP,SP
    PUSH  [BP+6]
    MOV   DI,[BP+4]
```

המשך בעמוד 44

```
    MOV   AL,[DI]
    _____(1)_____
AGAIN:
    CALL  CALLEE
    POP   SI
    OR    SI,SI
    JE    NOT_FOUND
    CMP   WORD PTR[DI+1],NULL
    JE    FOUND
    CMP   _____(2)_____,NULL
    JE    NOT_FOUND
    MOV   SI,[SI+1]
    MOV   DI,[DI+1]
    _____(3)____
    _____(4)_____
    JMP   AGAIN
    POP   BP
    RET   4
FOUND:
    _____(5)_____
    JMP   CA_SOF
NOT_FOUND:
    _____(6)_____
CA_SOF: POP       BP
    RET 4
```

בהצלחה!

נספח א': דף תשובות לשאלות 1–2

לשאלון 714001, אביב תשע"ז

מקום למדבקת נבחן

הדבק את מדבקת הנבחן שלך במקום המיועד לכך, והדק את הדף הזה אל מחברת הבחינה שלך.

הקף בעיגול את הספרה המייצגת את התשובה הנכונה לכל סעיף.

שאלה 1

סעיף א	1	2	3	4
סעיף ב	1	2	3	4
סעיף ג	1	2	3	4
סעיף ד	1	2	3	4
סעיף ה	1	2	3	4
סעיף ו	1	2	3	4
סעיף ז	1	2	3	4
סעיף ח	1	2	3	4
סעיף ט	1	2	3	4
סעיף י	1	2	3	4
סעיף י"א	1	2	3	4
סעיף י"ב	1	2	3	4
סעיף י"ג	1	2	3	4
סעיף י"ד	1	2	3	4
סעיף ט"ו	1	2	3	4
סעיף ט"ז	1	2	3	4
סעיף י"ז	1	2	3	4
סעיף י"ח	1	2	3	4
סעיף י"ט	1	2	3	4
סעיף כ	1	2	3	4
סעיף כ"א	1	2	3	4

שאלה 2

סעיף א	1	2	3	4
סעיף ב	1	2	3	4
סעיף ג	1	2	3	4
סעיף ד	1	2	3	4
סעיף ה	1	2	3	4
סעיף ו	1	2	3	4
סעיף ז	1	2	3	4
סעיף ח	1	2	3	4
סעיף ט	1	2	3	4
סעיף י	1	2	3	4

נספח ב': מילון מונחים (2 עמודים)

לשאלון 714001, אביב תשע"ז

המונח	תרגום המונח		
	ערבית	רוסית	אנגלית
חלק א' – תכנות מערכות בשפת C :			
אתחול	ابتداء	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 סיביות	سلسلة أرقام ثنائيّة / سلسلة بتات	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
חלק ב' – 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	أرقام ثنائيّة (بتات)	биты	bits
פלינדרום	سِياقٌ مُتَناظِر (بولينديروم)	Палиндром	palindrome
קומפילציה (הידור)	ترجمة الأوامر	компиляция	compilation
שגרה	رتابة/ روتين	Функция, рутина	routine
שגרה רקורסיבית	إجراء تراجعيّ	Рекурсивная функция	recursive routine