**מדינת ישראל**
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תש"ף, 2020
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.
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־2 עמודי נספח.**

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 (25 נקודות)**

מעוניינים ליצור ספר לימוד דיגיטלי המאפשר כמה פעולות:

**א.** קריאה רציפה בספר, החל מעמוד מסוים.

**ב.** קריאה סדרתית בספר, על־פי נושא מסוים.

**שים לב:** הדפים המשויכים לנושא אינם בהכרח עוקבים.

**ג.** דפדוף בספר - בקריאה רציפה ובקריאה סדרתית לפי נושא ניתן לבצע דפדוף לדף הקודם ולדף הבא.

**ד.** הצגת האינדקס – מפתח מילים חשובות שנבחרו מבין כל המילים המופיעות בספר, וכן מספרי הדפים שבהם הן מופיעות.

**הנחות יסוד:**

**1.** כל דף עוסק בנושא אחד בלבד (משויך לנושא אחד בלבד), אבל לנושא כלשהו יכולים להיות משויכים דפים רבים.

**2.** כל נושא חייב להיות משויך לפחות לדף אחד שעוסק בו.

**3.** כל דף חייב להיות משויך לנושא שבו הוא עוסק.

**4.** המילים החשובות בדף (מילות המפתח), שייכללו באינדקס, תחומות בסימן $ משני הצדדים. הנח כי הסימן $ אינו מופיע כחלק מיתר הטקסט בספר.

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

עתה נפרט את טיפוס הנתונים **צומת ברשימה מקושרת**:

בעבור רשימה מקושרת השייכת לנושא – הצומת ייצג דף השייך לנושא.

בעבור רשימה מקושרת השייכת למילה באינדקס - הצומת ייצג דף שבו מופיעה המילה.

המשך בעמוד 3

להלן מבנה של תא **ברשימה המקושרת** בשפת C :

```
typedef struct listType
{
   int ind;                  // מיקום הדף במערך הדפים
   struct listType *next;    // שדה קישור
}listRec, *listPtr;
```

עתה נפרט את טיפוס הנתונים **מערך דינמי של הדפים**.

להלן מבנה של תא **במערך הדינמי של הדפים** בשפת C :

```
typedef struct pageType
{
   int pageNum;          // מספר הדף
   int topicID;          // מספר הנושא
   char content[3000];    // תכולת הדף, כולל הכותרת
}pageRec, *pagePtr;
```

עתה נפרט את טיפוס הנתונים **מערך דינמי של הנושאים**.

להלן מבנה של תא **במערך הדינמי של הנושאים** בשפת C :

```
typedef struct topicType
{
   int topicID;      // מספר הנושא
   char title[40];   // שם הנושא
   listPtr tlist;    // מצביע לרשימת הדפים העוסקים בנושא הזה
}topicRec, *topicPtr;
```

כל תא במערך זה מייצג נושא, ומכיל, בין היתר, מצביע לראש **רשימה מעגלית מקושרת**, המייצגת את הדפים העוסקים בנושא הזה. רשימה זו מכילה את המיקומים של הדפים במערך הדפים.

המשך בעמוד 4

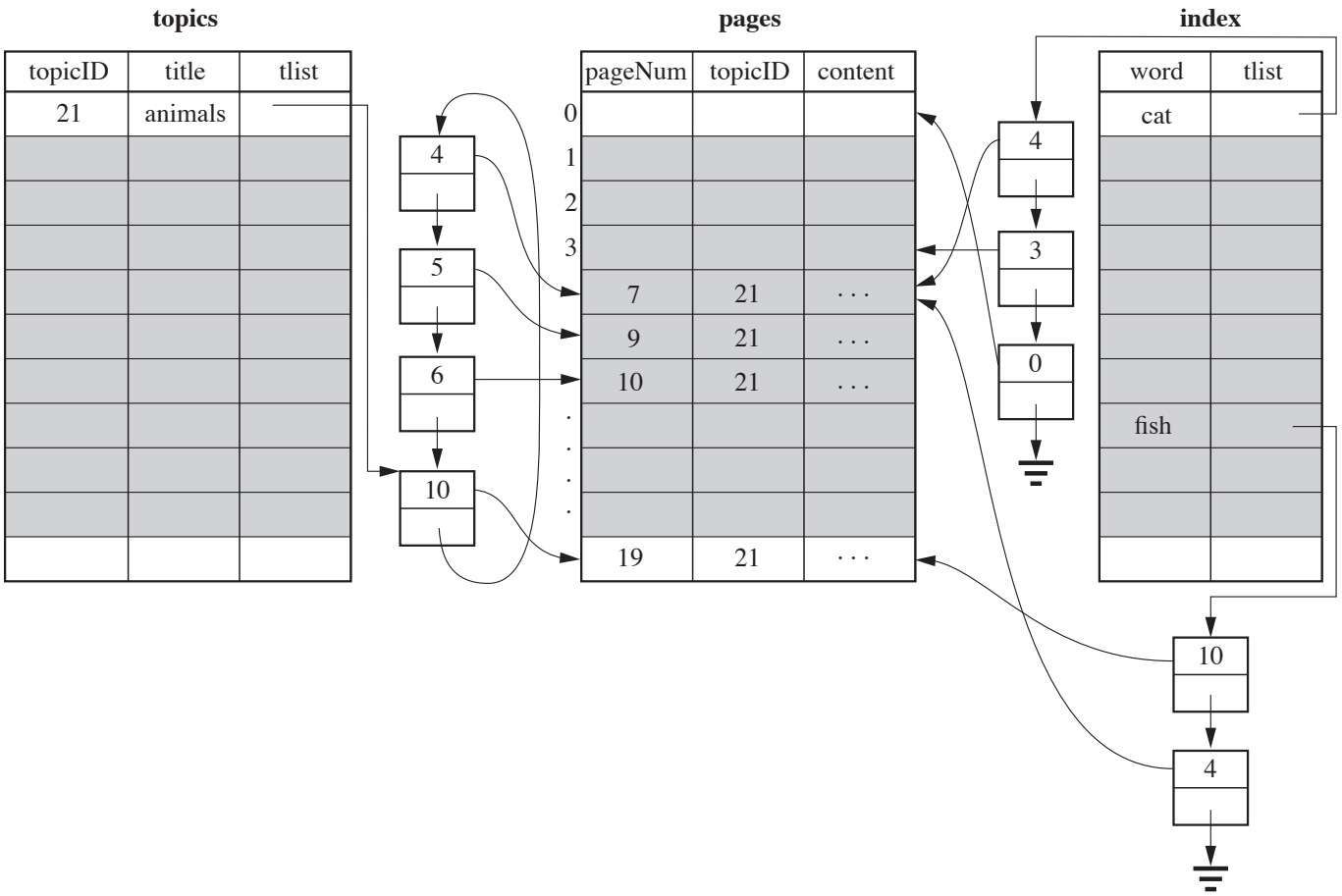
עתה נפרט את טיפוס הנתונים **המערך הדינמי – אינדקס**.

להלן מבנה של תא **במערך הדינמי אינדקס** בשפת C :

```
typedef struct indexType
{
   char word[20];    // מילת מפתח
   listPtr tlist;    // שדה קישור לרשימת הדפים הכוללים את מילת המפתח הזו
}indexRec, *indexPtr;
```

כל תא במערך זה מייצג מילה באינדקס, ומכיל, בין היתר, מצביע לראש רשימה מקושרת, המכילה את מספרי הדפים שבהם מופיעה המילה הזו. רשימה זו מכילה את המיקומים של הדפים במערך הדפים.

באיור לשאלה 1 מתוארים מבני הנתונים התומכים בפעולות המערכת הממוחשבת.




**איור לשאלה 1**

המשך בעמוד 5

**שלבים בבניית הספר הדיגיטלי:**

**א.** קולטים את רשומות כל הנושאים בספר לתוך המערך הדינמי של הנושאים (topics) .

**ב.** קולטים את הדפים (לא בהכרח לפי הסדר) לתוך המערך הדינמי של הדפים (pages) . בעבור כל דף שנוסף, מעתיקים את מילות המפתח (התחומות על־ידי הסימן $ משני צידיהן), ומכניסים אותן למערך הדינמי – אינדקס.

**ג.** ממיינים את מערך הדפים (pages) .

**ד.** לכל נושא ברשימת הנושאים יוצרים רשימה מקושרת **מעגלית** של הדפים השייכים לאותו נושא.

**ה.** לכל מילה באינדקס יוצרים רשימה מקושרת של הדפים שבהם מופיעה המילה הזו.

**ו.** ממיינים את מערך האינדקס (index) על־פי סדר אלפביתי.

**להלן הגדרות התקפות לכל הסעיפים שיבואו בהמשך:**

```
typedef enum {FAILURE,SUCCESS,INVALID_INPUT,RECORD_NOT_FOUND,DUPLICATE_ID}statusType;
typedef enum {FALSE,TRUE} boolean;
typedef enum {FOLLOW,BRANCH} modeType;  // קריאה רציפה, ו־BRANCH-קריאה סדרתית על־פי נושא FOLLOW-
```

המשך בעמוד 6

נתונה פונקצייה שכותרתה היא:

```
void init(void);
```

פונקצייה זו מאתחלת את המערכת הממוחשבת, ומבצעת את הפעולות המתוארות בשלבים לבניית הספר הדיגיטלי, המופיעים בעמוד 5 , כאשר חלק מן הפעולות ימומשו בהמשך.

**נתונה ספריית פונקציות המכילה, בין היתר, את הפונקציות האלה:**

| | |
|---|---|
| `int findTopic(int topicID);` | פונקצייה זו מקבלת את topicID , שהוא מספר של נושא כלשהו, ומחזירה את המיקום שלו במערך הנושאים. אם הוא לא נמצא – הפונקצייה מחזירה 1– . |
| `int findPage(int pageNum);` | פונקצייה זו מקבלת את pageNum , שהוא מספר דף כלשהו, ומחזירה את המיקום שלו במערך הדפים. אם הוא לא נמצא – הפונקצייה מחזירה 1– . |
| `int findWord(char word[]);` | פונקצייה זו מקבלת את word , שהיא מילת מפתח כלשהי, ומחזירה את המיקום שלה במערך האינדקס. אם היא לא נמצאת – הפונקצייה מחזירה 1– . |
| `void printPage(int pageNum);` | פונקצייה זו מקבלת את pageNum , שהוא מספר דף כלשהו, ומדפיסה את תוכנו. |

**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.**

**להלן הגדרות של משתנים גלובליים:**

```
topicPtr topics = NULL;          // מצביע למערך הדינמי של הנושאים
pagePtr pages = NULL;            // מצביע למערך הדינמי של הדפים
indexPtr indexes = NULL;         // מצביע למערך הדינמי של האינדקס
int lastTopic = 0;               // כמות הנושאים במערך הנושאים
int lastPage = 0;                // כמות הדפים במערך הדפים
int lastIndex = 0;               // כמות מילות המפתח במערך האינדקס
int currentPage = 0;             // מיקום הדף הנוכחי במערך הדפים
int currentTopic = 0;            // מספר הנושא הנוכחי
modeType mode = FOLLOW;          // אפשרות הקריאה (רציפה, או סדרתית על־פי נושא) כאשר הערך ההתחלתי הוא קריאה רציפה
```

המשך בעמוד 7

**ענה על הסעיפים שלהלן:**

**(5 נק')** **א.** לפניך פונקצייה שכותרתה:

```
void insertIndex(char word[],int page)
```

פונקצייה זו מקבלת את הפרמטרים האלה:

word – מילת מפתח,

page – המיקום של הדף המכיל את המילה word במערך הדפים.

הפונקצייה מבצעת את הפעולות שלהלן:

מאתרת את המילה במערך האינדקס. אם המילה לא קיימת במערך האינדקס, היא מוסיפה אותה למערך.

הפונקצייה מוסיפה את מספר הדף לרשימה המקושרת של הדפים המכילים את מילת המפתח הזו.

**הערות:**

1. אין צורך לבדוק את תקינותן של ההקצאות הדינמיות בתוכנית.
2. ברשימה מקושרת האיברים מוכנסים לראש הרשימה.

המשך בעמוד 8

בפונקצייה שלהלן 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 insertIndex(char word[], int page)
{
     int ip;
     listPtr lst;
     ip = ________(1)_________;
     if (ip == -1)
     {
          if (lastIndex == 0)
          {
              indexes = malloc(sizeof(indexRec));
          }
          else
          {
              indexes = realloc(indexes , ________(2)_________);
          }
          strcpy(indexes[lastIndex].word, word);
          indexes[lastIndex].tlist = NULL;
      }
      lst = malloc(sizeof(listRec));
      lst->ind = ________(3)_________;
      if(ip != -1)
      {
         lst->next = indexes[ip].tlist;
         indexes[ip].tlist = lst;
      }
      else
      {
         lst->next = ________(4)_________;
         ________(5)_________ = lst;
        lastIndex++;
      }
}
```

המשך בעמוד 9

**ב.** (5 נק') לפניך פונקצייה **רקורסיבית** שכותרתה:

```
void extractWords(char content[],int page)
```

פונקצייה זו מקבלת את הפרמטרים האלה:

content – תוכן הדף,

page – מיקום הדף במערך הדפים.

הפונקצייה מאתרת את מילות המפתח בדף, ומטפלת בהכנסתן למערך האינדקס, כלומר מוסיפה את הדף לרשימת הדפים הכוללים את מילת המפתח הזו.

**שים לב:** הפונקצייה נעזרת באחת הפונקציות שהוזכרו לעיל.

הפונקצייה נעזרת בפונקציית המערכת char * strchr(char * s, char ch) המשמשת לחיפוש תו במחרוזת. פונקצייה זו מקבלת מחרוזת s ותו ch , ומחזירה מצביע למיקום התו במחרוזת. אם התו לא קיים במחרוזת – הפונקצייה מחזירה NULL .

המשך בעמוד 10

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 extractWords(char content[], int page)
{
   char *loc1, *loc2, *loc;
   char word[20];
   int i;
   int ch = '$';
   if(strlen(content) == 0) return ;
   loc1 = _______(1)________;
   if (loc1 == NULL) return;
   loc2 = strchr(_______(2)________, ch);
   loc = loc1+1;
   i = 0;
   while (_______(3)________)
   {
      word[i++] = _______(4)________;
      loc = loc+1;
   }
   word[i] ='\0';
   _______(5)________;
   extractWords(loc2+1, page);
}
```

המשך בעמוד 11

**ג.** (5 נק') לפניך פונקצייה שכותרתה:

```
statusType insertPage(int pageNum, int topicID, char content[])
```

פונקצייה זו מקבלת את הפרמטרים האלה:

pageNum – מספר הדף,

topicID – מספר הנושא,

content[] – תוכן הדף.

פונקצייה זו מטפלת בהכנסת דף למערך הדפים.

הפונקצייה בודקת אם הנושא שמספרו topicID וכן אם הדף שמספרו pageNum קיימים במערכת.

אם הנושא לא קיים – הפונקצייה מחזירה את הערך RECORD_NOT_FOUND .

אם הדף כבר קיים – הפונקצייה מחזירה את הערך DUPLICATE_ID .

אם הדף הוכנס בהצלחה למערך הדפים - הפונקצייה מחזירה את הערך SUCCESS .

המשך בעמוד 12

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.

```
statusType insertPage(int pageNum, int topicID,char content[])
{
     statusType status = SUCCESS;
     int tp ,page;
     tp = ________(1)_________;
     page = findPage(pageNum);
     if (tp == -1) return RECORD_NOT_FOUND;
     if (page != -1) return DUPLICATE_ID;
     if (lastPage == 0)
     {
        pages = malloc(sizeof(pageRec));
     }
     else
     {
        pages = realloc(pages, (________(2)_________)*sizeof(pageRec));
     }
      pages[lastPage].pageNum = pageNum;
      pages[lastPage].topicID = topicID;
      ________(3)_________;
      ________(4)_________;
      return status ;
}
```

המשך בעמוד 13

**ד.** לפניך פונקצייה שכותרתה: **(6 נק')**

```
void attachment(void)
```

פונקצייה זו סורקת את כל דפי הספר.

עבור כל דף היא בודקת אם הנושא שבו הדף עוסק מופיע במערך הנושאים.

אם מספר הנושא המופיע בדף קיים במערך הנושאים, הפונקצייה:

1. מוסיפה את מספר הדף לרשימה המקושרת **המעגלית** של הדפים העוסקים בנושא זה.
2. מחלצת מכל דף את מילות המפתח, ומטפלת בהכנסתן לאינדקס.

אם מספר הנושא המופיע בדף אינו קיים במערך הנושאים – הפונקצייה תדפיס הודעה מתאימה.

המשך בעמוד 14

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 attachment(void)
{
int i, t;
listPtr lst;
for(i= 0; i < lastPage; i++)
 {
      t = ________(1)_________;
      if (t != -1)
      {
            lst = malloc(sizeof(listRec));
            lst->ind = i;
            if(topics[t].tlist == NULL)
            {
                  lst->next = lst;
                  ________(2)_________;
            }
            else
            {
                  lst->next = topics[t]. ________(3)_________;
                  ________(4)_________ = lst ;
                  topics[t].tlist = lst;
            }
            extractWords( ________(5)_________);
      }
      else printf("page: %d topic: %d NOT FOUND", i, pages[i].topicID);
 }
}
```

המשך בעמוד 15

**(4 נק')** **ה.** לפניך פונקצייה שכותרתה:

```
void nextPage(void)
```

פונקצייה זו בודקת באיזה ממצבי הקריאה נמצאת המערכת.

אם המערכת נמצאת במצב FOLLOW (קריאה רציפה), הפונקצייה תדפיס את הדף העוקב.

אם היא נמצאת במצב BRANCH (קריאה סדרתית על־פי נושא), הפונקצייה תדפיס את הדף הבא אחרי הדף הזה בתוך אותו הנושא.

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 nextPage(void)
{
   int page;
   int tp;
   listPtr lst;
   if (mode == FOLLOW)
   {
      page = ________(1)_________;
      printPage(page);
   }
   else
   {
      tp = ________(2)_________;
      lst = ________(3)_________;
      while (lst && lst->next)
      {
            if(lst->ind == currentPage)
           {
             page = ________(4)_________;
             printPage(page);
             break;
           }
           ________(5)_________;
      }
    }
 }
```

המשך בעמוד 16

**שאלה 2 – שאלת חובה (10 נקודות)**

נתון מערך **דו־ממדי** שבו כל תא מכיל אות.

המטרה היא לאתר את כל מופעיה של מילה כלשהי במערך (תפזורת).

כמקובל בכללי התפזורת הרגילים, המילה יכולה להיקרא במאוזן, במאונך או באלכסון, ויכולה להיקרא משני הכיוונים בכל אחת מן האפשרויות.

תחילה נחפש את האות הראשונה במילה. נניח שמצאנו אותה במקום i,j . כעת, לכל איבר במקום i,j יש לכל היותר שמונה שכנים – שמונה כיוונים שונים שבהם יכולה להימצא האות השנייה במילה.

לדוגמה:

| | | |
|---|---|---|
| i–1, j–1 | i–1, j | i–1, j+1 |
| i, j–1 | i, j | i, j+1 |
| i+1, j–1 | i+1, j | i+1, j+1 |

נשתמש בשני מערכים x[k] ו־y[k] כאשר $0 \le k \le 7$ , x[k] מציין את התוספת של i ואילו y[k] – את התוספת של j (תוספת אפשרית היא 0, 1 או 1–) באופן הזה:

```
                0    1    2    3    4    5    6    7
int x[] = {   -1,  -1,  -1,   0,   0,   1,   1   1  };

                0    1    2    3    4    5    6    7
int y[] = {   -1,   0,   1,  -1,   1,  -1,   0,   1  };
```

כך, ניתן לייצג את שמונה האפשרויות עבור כיוון ההתקדמות באופן הזה:

| | | |
|---|---|---|
| i+x[0], j+y[0] | i+x[1], j+y[1] | i+x[2], j+y[2] |
| i+x[3], j+y[3] | i, j | i+x[4], j+y[4] |
| i+x[5], j+y[5] | i+x[6], j+y[6] | i+x[7], j+y[7] |

להלן הגדרות בעבור המשתנים הגלובליים:

```
#define R 3

#define C 14

typedef enum {FALSE,TRUE} bool;

int x[] = { -1, -1, -1, 0, 0, 1, 1, 1 };

int y[] = { -1, 0, 1, -1, 1, -1, 0, 1 };
```

המשך בעמוד 17

**(7.5 נק') א.** לפניך פונקצייה שכותרתה:

```
bool search2D(char grid[R][C], int row, int col, char word[])
```

הפונקצייה מקבלת את הפרמטרים שלהלן:

grid – מערך דו־ממדי שמכיל אותיות בתפזורת,

row – מספר השורה אשר ממנה מתחילים את חיפוש המילה,

col – מספר העמודה אשר ממנה מתחילים את חיפוש המילה,

word – המילה שאותה מחפשים.

הפונקצייה תחזיר TRUE אם מצאנו את המילה באחד מכיווני החיפוש של התא שממנו התחלנו את החיפוש. אחרת, הפונקצייה תחזיר FALSE .

המשך בעמוד 18

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.

```
bool search2D(char grid[R][C], int row, int col, char word[])
{
    int len, dir, k, rd, cd;
    if (grid[row][col] != word[0])
        return FALSE;
    len = strlen(word);
    for (dir = 0; dir < 8; dir++)
    {
        rd = row + x[dir];
        cd = col + y[dir];
        for (k = 1; _____(1)_____; k++)
        {
            if (rd >= R || rd < 0 || cd >= C || cd < 0)
                break;
            if (_____(2)_____)
                break;
            rd += _____(3)_____;
            cd += _____(4)_____;
        }
        if (_____(5)_____)
            return TRUE;
    }
    return FALSE;
}
```

המשך בעמוד 19

**ב.** (2.5 נק') לפניך פונקצייה שכותרתה:

```
void patternSearch(char grid[R][C], char word[])
```

ולאחריה התוכנית הראשית.

כתוב במחברתך מה תדפיס התוכנית הראשית.

```
void patternSearch(char grid[R][C], char word[])
{
    int row, col;
    for (row = 0; row < R; row++)
       for (col = 0; col < C; col++)
          if (search2D(grid, row, col, world))
              printf( "\n pattern found at row: %d \n" , row);
}


int main ()
{
    char grid[R][C] = {
                        "GDEKSFORGEEKS",
                        "GEEKSQUIZGEEK",
                        "IDEQAPRACTICE"
                       };
    patternSearch(grid, "GEE");
    return 0;
}
```

המשך בעמוד 20

## <u>פרק שני</u> (15 נקודות)

**ענה על <u>אחת</u> מבין השאלות 3–4 (לכל שאלה – 15 נקודות).**

### שאלה 3 (15 נקודות)

**(8 נק')** **א.** נתון מערך דו־ממדי mat[M][N] , שמכיל את אחד הערכים 0 או 1 בכל אחד מן התאים שבו, לדוגמה:

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

כל אחת משורות המערך ממוינת בסדר עולה.

יש למצוא את השורה שמכילה את רצף האחדות המקסימלי.

הנח כי $N > 2$ , וכן הנח כי יש רק שורה אחת המכילה רצף מקסימלי של אחדות.

לפניך שתי פונקציות שמבצעות את הפעולה ביעילות.

כותרת הפונקציה **הראשונה** (פונקצייה **רקורסיבית**) היא:

```
int getFirstOccurence(int arr[], int low, int high)
```

פונקצייה זו מקבלת את הפרמטרים האלה:

arr [ ] – מערך **חד־ממדי** המייצג שורה במטריצה,

low – מיקום האיבר שממנו מתחיל החיפוש במערך,

high – מיקום האיבר שבו מסתיים החיפוש במערך.

הפונקצייה מחזירה את המיקום שבו מופיע ה־1 הראשון בשורה, ומשתמשת באלגוריתם לחיפוש בינארי.

אם השורה מכילה רק אפסים, הפונקצייה מחזירה 1– .

כותרת הפונקצייה **השנייה** היא:

```
int findRowMaxOne(int mat[][N])
```

והיא מקבלת את המטריצה mat[][N] , ומחזירה את השורה שבה מופיע רצף האחדות המקסימלי.

הפונקצייה נעזרת בפונקצייה getFirstOccurence שלעיל.

המשך בעמוד 21

**בשתי הפונקציות** יחד 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 getFirstOccurence(int arr[], int low, int high)
{
  if(high >= low)
  {
    int mid = _____(1)_____;
    if (arr[mid-1] == 0 && arr[mid] == 1)
          return mid;
    else if (arr[mid] == 0)
          return _____(2)_____(arr, _____(3)_____, high);
    else
          return _____(4)_____(arr, low, _____(5)_____);
  }
  return -1;
}
```

```
int findRowMaxOne(int mat[][N])
{
    int max_row = 0, max = -1;
    int i, ind, count = -1;
    for (i = 0; i < M; i++)
    {
       if (mat[i][0] == 1)
       {
          max = N;
          max_row = i;
          break;
       }
       ind = _____(6)_____;
       if (ind != -1) count = N - ind;
       if (count > max)
       {
          max = _____(7)_____;
          max_row = _____(8)_____;
       }
    }
    return max_row;
}
```

המשך בעמוד 22

**ב.** **(3 נק')** להלן תוכנית בשפת C :

```
#include <stdio.h>
int main( )
{
  static int a[] = {10, 20, 30, 40, 50};
  static int *p[] = {a, a+3, a+4, a+1, a+2};
  int **ptr = p;
  ptr++;
  printf("%d %d", ptr - p, **ptr);
  return 0;
}
```

מהו הפלט של התוכנית? כתוב את מספר התשובה הנכונה במחברתך.

1. 1 40
2. 1 50
3. 2 10
4. 2 30

המשך בעמוד 23

**ג.** **(4 נק')** להלן תוכנית בשפת C :

```
#include <stdio.h>
int fun(char *str1)
{
  char *str2 = str1;
  while(*++str1);
  return (str1-str2);
}

int main()
{
  char *str = "GeeksQuiz";
  printf("%d", fun(str));
  return 0;
}
```

מהו הפלט של התוכנית? כתוב את מספר התשובה הנכונה במחברתך.

1. 10
2. 9
3. 8
4. מספר אקראי

המשך בעמוד 24

**שאלה 4 (15 נקודות)**

**א.** **(8 נק')** נתון מערך [ ]arr המכיל מספרים שלמים. המטרה היא לסדר את איברי המערך כך שכל איבר שהאינדקס שלו במערך הוא אי־זוגי יהיה גדול מכל שכן שנמצא לצידו (מימין או משמאל).

לפניך פונקצייה שכותרתה:

```
void rearrangeArray(int arr[], int n)
```

פונקצייה זו מקבלת את הפרמטרים האלה:

arr[ ] – מערך חד־מימדי,

n – גודל המערך.

הפונקצייה תסדר את איברי המערך בהתאם לדרישה המובאת בפתיח השאלה.

הפונקצייה נעזרת בפונקצייה שכותרתה:

`void swap(int arr[], int i, int j)` , המחליפה את האיבר שבמקום ה־i במערך באיבר שבמקום ה־j .

בפונקצייה `rearrangeArray` 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 rearrangeArray(int arr[], int n)
{
    int i;
    for (i = 1; i < n; __________(1)__________)
    {
        if (arr[i - 1] > arr[i])
        {
            __________(2)__________;
        }
        if (i + 1 < n && __________(3)__________)
        {
            __________(4)__________;
        }
    }
}
```

המשך בעמוד 25

**ב.** (3 נק') להלן תוכנית בשפת C:

```
#include <stdio.h>
void f(int *p, int *q)
{
   p = q;
  *p = 2;
}

int main()
{
  int i = 0, j = 1;
  f(&i, &j);
  printf("%d %d\n", i, j);
  getchar();
  return 0;
}
```

מהו הפלט של התוכנית? כתוב את מספר התשובה הנכונה במחברתך.

1. 0 1
2. 0 2
3. 2 1
4. 2 2

המשך בעמוד 26

**ג.** להלן תוכנית בשפת C : **(4 נק')**

```
#include <stdio.h>
int fun(char *p)
{
    int current = 1, i = 1;
    if (p == NULL || *p =='\0') return 0;
    while (*(p+current))
    {
        if (p[current] != p[current-1])
        {
            p[i] = p[current];
            i++;
        }
        current++;
    }
    *(p+i)='\0';
    return i;
}

int main()
{
    char str[] = "geeksskeeg";
    fun(str);
    puts(str);
    return 0;
}
```

המשך בעמוד 27

מה יהיה הפלט של התוכנית? כתוב את מספר התשובה הנכונה במחברתך.

1. gekskeg
2. geeksskeeg
3. geeks
4. geekss

המשך בעמוד 28

# חלק ב': שפת סף (50 נקודות)

## פרק שלישי (20 נקודות)

**ענה על שאלה 5 - שאלת חובה.**

**שאלה 5 (20 נקודות)**

לפניך תוכנית בשפת אסמבלי, הכוללת שגרה רקורסיבית בשם REC .

התוכנית סורקת מערך של מילים (words) אשר כל אחד מן הבתים (bytes) שלהן שונה מאפס.

הנח כי איברי המערך מייצגים מספרים **בבסיס עשרוני**.

```
SSEG SEGMENT STACK 'STACK'
     DB 100 DUP()
SSEG ENDS
DSEG SEGMENT
     NUM DW 500, 10000
     DW 0
     P  DW NUM
     N  DB 7       ;(*)
DESG ENDS
CODE SEGMENT
      ASSUME CS:CODE, DS:DESG
START:MOV  AX, DSEG
      MOV  DS, AX
      MOV  DL, N
      XOR  SI, SI    ;(**)
      PUSH P
      CALL REC
```

המשך בעמוד 29

```
EXIT: MOV  AH, 4CH
      INT  21H
REC:  PUSH BP
      MOV  BP, SP
      MOV  BX, [BP+4]
      CMP  WORD PTR [BX+SI], 0
      JE   SOF
      MOV  AX, [BX+SI]
      CMP  AH, DL
      JAE  CON
      DIV  DL
      XCHG AX, [BX+SI]
CON:  INC  SI
      INC  SI
      PUSH P
      CALL REC
SOF:  POP  BP
      RET  2    ;(***)
CODE ENDS
END START
```

המשך בעמוד 30

**(4 נק')** **א.** עקוב אחר פעולתה של השגרה REC בעבור המערך NUM המכיל את שתי המילים שלהלן, הנתונות בבסיס עשרוני:

| 500 | 10000 |
|---|---|

כתוב **במחברתך** מה יהיה תוכנו בבסיס עשרוני של כל בית (byte) בכל אחת מן המילים שבמערך, בהגיע התוכנית לתווית EXIT .

**(4 נק')** **ב.** מחליפים **בקטע הקוד הנתון** את שורת הפקודה המסומנת ב־(*) בשורת הפקודה: N DB 3 .

עקוב אחר פעולתה של השגרה REC בעבור המערך NUM המכיל את שלוש המילים שלהלן, הנתונות בבסיס עשרוני:

| 29 | 2020 | 933 |
|---|---|---|

כתוב **במחברתך** מה יהיה תוכנו בבסיס עשרוני של כל בית (byte) בכל אחת מן המילים שבמערך, בהגיע התוכנית לתווית EXIT .

**(4 נק')** **ג.** מה מבצעת התוכנית בעבור המערך NUM והמשתנה N ? ענה בקצרה ובאופן כללי, מבלי להתייחס להוראה ספציפית.

**(4 נק')** **ד.** הצע פקודה אחרת שתחליף את השורה המסומנת ב־(**) כך שביצועי התוכנית לא ישתנו.

**(4 נק')** **ה.** אם נחליף **בקטע הקוד הנתון** את שורת הפקודה המסומנת ב־(***) בשורת הפקודה: RET ,
האם ישתנו ביצועי התוכנית? ענה "כן" או "לא", ונמק בקצרה.

המשך בעמוד 31

## **פרק רביעי (30 נקודות)**

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

### **שאלה 6 (15 נקודות)**

להלן קטע קוד:

```
        MOV  BX,   3000H
        MOV  CX,   7FH
AGAIN:  MOV  AL,   CL
        AND  AL,   4DH
        CMP  AL,   01001101B   ;(*)
        JNE  NEXT
        MOV  [BX], CL
        INC  BX
NEXT:   LOOP AGAIN     ;(**)
```

התוכנית יוצרת מערך החל מהכתובת 3000H .

המשך בעמוד 32

**(5 נק')** **א.** בעבור המערך שנוצר על־ידי קטע הקוד שלעיל - הצג את ערכו של תוכן התא בכל אחת מן הכתובות, בשלוש שיטות הייצוג שלהלן: עשרוני, בינארי והקסדצימלי, החל מהכתובת 3000H .

העתק **למחברתך** את הטבלה שלהלן, והשלם אותה כנדרש.

| address | dec | bin | hex |
|---|---|---|---|
| 3000H | | | |
| 3001H | | | |
| ... | | | |
| ... | | | |
| ... | | | |
| ... | | | |
| ... | | | |
| ... | | | |

**(5 נק')** **ב.** על פי קטע הקוד שלעיל - מתבצעת חזרה אל התווית AGAIN 127 פעמים.

ניתן לקצר את ריצת הלולאה על־ידי החלפת השורה המסומנת ב־(**) בשלוש ההוראות שלהלן:

```
NEXT: DEC  CX
      CMP  CL , ________(1)________
      JAE  ________(2)________
```

בהוראות הללו 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.

**(5 נק')** **ג.** אם נחליף **בקטע הקוד הנתון** את שורת הפקודה המסומנת ב־(*) בשורת הפקודה: TEST AL, 01001101B , האם ישתנו ביצועי התוכנית? ענה "כן" או "לא", ונמק בקצרה.

המשך בעמוד 33

**שאלה 7 (15 נקודות)**

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

- Info – שדה מידע שגודלו 8 סיביות, המכיל ערך מספרי.
- Next – מצביע אל הצומת הבא ברשימה. גודלו של המצביע הוא מילה אחת (16 סיביות), וערכו של המצביע Next בצומת האחרון הוא 1– , לציון סוף הרשימה.

נוסף על כך, נתון כי המשתנה LIST הוא מצביע לצומת הראשון ברשימה, כלומר LIST מכיל את הכתובת של הצומת הראשון ברשימה המקושרת.

התוכנית שלהלן מקבלת מצביע LIST לרשימה מקושרת, ומוציאה ממנה את הצומת המכיל מספר שלילי **בשדה ה־Info** שלו.

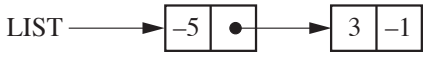
**הנחות**:

**1.** ברשימה יש צומת **יחיד** ששדה ה־Info שלו מכיל מספר שלילי.

**2.** ברשימה יש לפחות צומת אחד ששדה ה־Info שלו הוא חיובי.

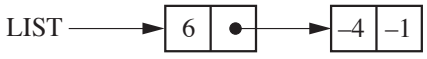
**דוגמאות**:

**1.** עבור הרשימה הזו:




לאחר הרצת התוכנית תתקבל הרשימה שלהלן:

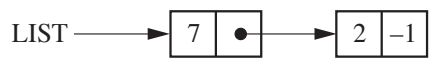

**2.** עבור הרשימה הזו:




לאחר הרצת התוכנית תתקבל הרשימה שלהלן:


**3.** עבור הרשימה הזו:


לאחר הרצת התוכנית תתקבל הרשימה שלהלן:




המשך בעמוד 34

לפניך קטע קוד המבצע את הנדרש.

בקטע הקוד 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.

```
      MOV  BX, LIST
      MOV  DI, BX
      MOV  AL, [BX]
      CMP  AL, 0
      _______(1)_______ NEXT
      MOV  BX, [BX + 1]
      MOV  _______(2)_______, BX
      JMP  END_RESHIMA
NEXT: MOV  DI, BX
      MOV  BX, _______(3)_______
      MOV  AL, [BX]
      CMP  AL, 0
      JGE  _______(4)_______
      CMP  [WORD PTR (BX + 1)], -1
      JNE  REMOVE_INTERNAL_ITEM
      MOV  [WORD PTR DI], -1
      JMP  END_RESHIMA
REMOVE_INTERNAL_ITEM:
      MOV  BX, [BX+1]
      MOV  _______(5)_______, BX
END_RESHIMA:
      MOV  AH, 4CH
      INT  21H
```

המשך בעמוד 35

**שאלה 8 (15 נקודות)**

להלן הגדרה של סגמנט נתונים:

```
SDATA SEGMENT
      X1  DB '123', 1, 2, 3, '-1-2-3'
      X2  DB  200
      X3  DW  X2
SDATA ENDS
```

**(3 נק')** **א.** מהן הכתובות הלוגיות (ההיסט) בהתייחס לתחילת הסגמנט SDATA , שבהן ממוקמים תאי המערכים X1, X2, X3 ?

**(2 נק')** **ב.** מה יהיה תוכנו של האוגר DL , בייצוג בינארי, לאחר הרצת הפקודות שלהלן:

```
MOV DX, X3
NEG DL
```

נתון כי האוגר DS מכיל את המספר 3E25H .

הנח כי DS מגדיר סגמנט נתונים בגודל מלא (64 kb) .

**(2 נק')** **ג.** מהי הכתובת הפיזית, בייצוג הקסדצימלי, של תחילת סגמנט הנתונים שאותו מגדיר האוגר DS ?

**(2 נק')** **ד.** מהי הכתובת הפיזית, בייצוג הקסדצימלי, של סוף סגמנט הנתונים שאותו מגדיר האוגר DS ?
(כתובתו של ה־byte האחרון בסגמנט הנתונים.)

**(2 נק')** **ה.** אם נחלק את סגמנט הנתונים לארבעה חלקים שווים – מה יהיה גודלו של כל רבע סגמנט בייצוג הקסדצימלי?

**(2 נק')** **ו.** מה תהיה הכתובת הפיזית, בייצוג הקסדצימלי, של תחילת כל רבע סגמנט?

**(2 נק')** **ז.** מה תהיה הכתובת הלוגית (ההיסט) , בייצוג הקסדצימלי, של תחילת הרבע הרביעי?

**בהצלחה!**

# נספח: מילון מונחים (2 עמודים)

## לשאלון 714001, אביב תש"ף

| 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 גלובלי | 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

נספח: מילון מונחים
סמל 714001, אביב תש"ף

| המונח | תרגום המונח | | |
|---|---|---|---|
| | ערבית | רוסית | אנגלית |
| **חלק ב' – שפת סף:** | | | |
| אוגר | مخزن (ريغستر) | регистр | register |
| אלכסון ראשי | مائل رئيسيّ | главная диагональ | main diagonal |
| בסיס הקסדצימלי | 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 (بوليندروم) | палиндром | palindrome |
| קומפילציה (הידור) | ترجمة الأوامر | компиляция | compilation |
| שגרה | رتابة/ روتين | функция, рутина | routine |
| שגרה רקורסיבית | إجراء تراجعيّ | рекурсивная функция | recursive routine |