

סוג הבחינה: א. בגרות לבתי ספר על-יסודיים

ב. בגרות לנבחנים חיצוניים

מועד הבחינה: קיץ תשס"ה, 2005

מספר השאלון: 603,899205

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

פרק ראשון – בפרק זה ארבע שאלות, — (25×2) — 50 נקודות
ומהן יש לענות על שתיים.

פרק שני – בפרק זה שאלות בחמישה מסלולים שונים. — (25×2) — 50 נקודות
ענה על שאלות רק במסלול שלמדת,
לפי ההוראות בקבוצת השאלות במסלול זה.

סה"כ — 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב עילית בפרק הראשון

כתוב בשפה אחת בלבד – פסקל אֲ C.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – פסקל אֲ C.

(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת,

אחד מחמשת המסלולים: מערכות מחשב ואסמבלר, תורת המחשב,

מודלים חישוביים, חישוב מקבילי ומבוזר, תכנות מונחה עצמים.

כתוב במחברת הבחינה בלבד. בעמודים נפרדים, כל מה שברצונך לכתוב כטיוטה (ראשי פרקים, חישובים וכדומה).
רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

ב ה צ ל ח ה !

ה ש א ל ו ת

בשאלון זה שני פרקים: פרק ראשון ופרק שני.
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

פרק ראשון (50 נקודות)

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. לפניך קטע תכנית בפסקל וב-C:

פסקל

```

const N = 8;
type typeArray = array [1.. N] of integer;

function firstCheck (x , num : integer) : boolean;
    { הפונקציה מקבלת ספרה x ומספר שלם num גדול מאפס.
    { הפונקציה מחזירה...
begin
    if num = 0
        then firstCheck := false
    else if (num mod 10 = x)
        then firstCheck := true
    else firstCheck := firstCheck (x , num div 10);
end;

function secondCheck (a : typeArray ; k : integer) : integer;
    { הפונקציה מקבלת מערך a בגודל N, המכיל מספרים שלמים גדולים מאפס
    { ומספר שלם k בתחום 1 עד N (כולל).
    { הפונקציה מחזירה...
begin
    if (k = N)
        then secondCheck := 0
    else if (firstCheck (a[k] mod 10 , a[k+1]))
        then secondCheck := 1 + secondCheck (a , k+1)
    else secondCheck := secondCheck (a , k+1);
end;
/המשך בעמוד 3/
    
```

C

```

typedef N = 8;
int firstCheck (int x , int num)
//          הפונקציה מקבלת ספרה x ומספר שלם num גדול מאפס.
//          הפונקציה מחזירה ...
{
    if (num == 0)
        return 0;
    else
        if (num %10 == x)
            return 1;
        else
            return firstCheck(x , num/10);
};

int secondCheck (int a[] , int k)
//          הפונקציה מקבלת מערך a בגודל N, המכיל מספרים שלמים גדולים מאפס
//          ומספר שלם k בתחום 0 עד N-1 (כולל).
//          הפונקציה מחזירה ...
{
    if (k == (N-1))
        return 0;
    else
        if (firstCheck (a[k] %10 , a[k+1]) == 1)
            return 1+secondCheck (a , k+1);
        else
            return secondCheck (a , k+1);
};

```

א. מה מבצעת הפונקציה firstCheck ?

ב. נתון מערך חד-ממדי a בגודל 8.

תחילת המערך →

400	200	100	25	351	33	34	4
-----	-----	-----	----	-----	----	----	---

מה יחזיר הזימון secondCheck (a , 4) ?

רשום את המעקב אחר ביצוע הפונקציה secondCheck. אין צורך להראות מעקב

אחר הפונקציה firstCheck.

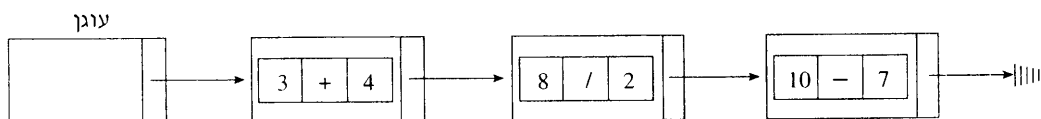
ג. מה מבצעת הפונקציה secondCheck ?

2. רשימה "חשבונית" L היא רשימה ששדה התוכן שלה מייצג ביטוי חשבוני.

הביטוי החשבוני מורכב משלושה חלקים:

- מספר שלם גדול מאפס
- תו אחד מבין ארבעת התווים:
 + המייצג חיבור
 - המייצג חיסור
 * המייצג כפל
 / המייצג חילוק
- מספר שלם גדול מאפס

דוגמה לרשימה "חשבונית" L:



א. הגדר בסביבת העבודה את טיפוס שדה התוכן של איבר ברשימה "חשבונית" L.

ב. ממש בסביבת העבודה תת־תכנית calculate, שתקבל רשימה "חשבונית" L ומקום p ברשימה. p הוא מקום ב־ L שאינו סוף־רשימה ואינו עוגן־רשימה. התת־תכנית תחזיר את התוצאה המתקבלת מהביטוי החשבוני הנמצא באיבר שבמקום p.

ג. ממש בסביבת העבודה תת־תכנית sumExpressions, שתקבל רשימה "חשבונית" L, ותחזיר את הסכום הכולל של תוצאות הביטויים החשבוניים הנמצאים ברשימה. בעבור רשימה ריקה יוחזר 0. עליך להשתמש בתת־תכנית calculate.

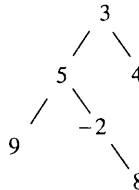
בעבור הרשימה "החשבונית" L בדוגמה הנתונה, התת־תכנית sumExpressions תחזיר 14.

הערה: אין צורך לממש בסביבת העבודה את הפעולות של ממשק רשימה.

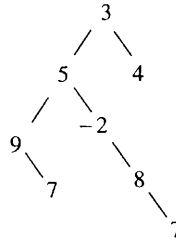
3. לפניך פעולה:

<p>הפעולה מקבלת מספר שלם N ועץ בינארי T לא ריק שערכיו הם מספרים שלמים. לכל עלה בעץ T שערכו גדול מ-N, הפעולה מוסיפה בן ימני שערכו N. הנחה: העץ T מאותחל.</p>	<p>הוסף_עלים (T, N)</p>
---	--------------------------------

דוגמה: נתון העץ T



לאחר זימון הפעולה **הוסף_עלים (T, 7)**, העץ T הוא:



כתוב אלגוריתם שיממש את הפעולה **הוסף_עלים (T, N)**.

4.

- במכון "ארוביקה" יש שיעורי התעמלות אֶרֹבִּית. השיעורים מתקיימים בימים א-ה בשעות 7:00-22:00. כל שיעור מתחיל בשעה עגולה ונמשך שעה אחת. בכל שעה פועל במכון שיעור אחד בלבד, ויכולים להשתתף בו לכל היותר 25 אנשים. אדם יכול להירשם לשיעור אחד בלבד. בהרשמה הוא מציין את היום בשבוע ואת השעה של השיעור שאליו הוא מבקש להירשם, ומוסר את שמו ואת מספר הטלפון הנייד שלו. ההרשמה לשיעור, ביום ובשעה המבוקשים, היא למשך שנה שלמה. אם אדם מעוניין להירשם לשיעור מסוים ואין מקום בשיעור, הוא נרשם כמתין ומוסר את הפרטים האלה: השם, מספר הטלפון הנייד, היום והשעה של השיעור המבוקש.
- א. יצג את המידע הדרוש לניהול הרישום לשיעור במכון "ארוביקה".
 - ב. כתוב אלגוריתם **משתתפים_בשיעור** שידפיס את השמות ואת מספרי הטלפון של כל המשתתפים בשיעור מסוים. רשום את הפרמטרים שהאלגוריתם מקבל ותיעוד.
 - ג. כתוב אלגוריתם **מקום_פנוי** שיקבל יום ושעה של שיעור שיש בו מקום פנוי, וישבץ אליו את הראשון מבין הממתנים שזה השיעור שביקש. האלגוריתם ימחק את רישומו של האדם כמתין. רשום את הפרמטרים שהאלגוריתם מקבל ותיעוד.

/המשך בעמוד 6/