

מבחן בקורס "מבוא כללי לתכנות", מקבץ הסייבר

סמסטר א' 2014

מועד א

מרצה : אמיר רובינשטיין

משך הבחינה : שעה וחצי

חומר עזר מותר : כל חומר עזר, למעט אלקטרוני (מחשב, מחשבון) וביולוגי (חברים)

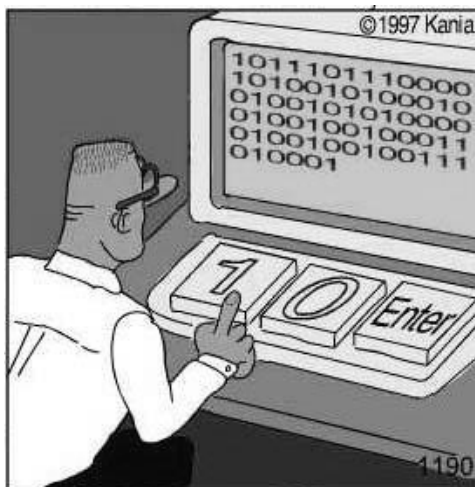
- יש לכתוב את כל התשובות בטופס בחינה זה. מחברות הטיוטה לא ייאספו.
- יש לכתוב את כל התשובות במקום המוקצב ובכתב קריא. חריגות משמעותיות מהמקום המוקצב, תשובות הכתובות בכתב קטן / לא ברור או תשובות שדורשות מאמצים רבים להבנתן עלולות לגרור הורדת ציון.
- במבחן 7 עמודים ו- 7 שאלות – בידקו שכולם בדיכס.
- מומלץ לא "להיתקע" על אף שאלה בודדת, אלא להמשיך לשאלות אחרות ולחזור לשאלה אח"כ.
- קיראו היטב את השאלות. הקפידו לענות בדיוק על מה שנשאלתם, ולנמק אם נדרשתם.

נא לרשום מספר תעודת זהות (ללא שם):

בהצלחה !

שאלה 1 (10 נק')

להלן בדיחה קצרה:



Real programmers code in binary.

מתכנתים אמיתיים כותבים קוד בבינארית.

הסבירו בקצרה את הבדיחה, תוך שימוש נכון במושגים "שפת תכנות", "שפת מכונה", "שפה בינארית", ו-
interpreter.

תשובה:

שאלה 2 (10 נק')

נתון מספר בינארי כלשהו בן 2 סיביות (bits). נסמן מספר זה ב- a . מוסיפים 3 אפסים מימין ל- a ומוסיפים 1 משמאל ל- a . למה שווה המספר החדש? הקיפו את התשובה הנכונה. אין צורך להסביר.

1. $8a+31$

2. $a+32$

3. $8a+32$

4. $31a+8$

5. אף אחת מהתשובות הקודמות אינה נכונה

שאלה 3 (15 נק')

להלן פונקציה בשפת Python:

```
def calc(a, b):  
    res = 0  
    while b>0:  
        res+=a  
        b-=1  
    return res
```

א. מה תחזיר הפונקציה עבור הקלט $a=1, b=0$? _____

ב. מה תחזיר הפונקציה עבור הקלט $a=1, b=5$? _____

ג. סמנו את התשובה הנכונה (אין צורך להסביר). עבור a ו- b שלמים חיוביים או אפס:

1. הפונקציה מחשבת את $a*b$

2. הפונקציה מחשבת את $b**a$

3. הפונקציה מחשבת את $a+b-1$

4. הפונקציה מחשבת את $b-a$

שאלה 4 (20 נק')

בהינתן רשימה של מספרים, נגדיר "פסגה" כמספר שגדול (ממש) גם משכנו השמאלי וגם משכנו הימני. אם לא קיים אחד מהשכנים הללו (בקצוות הרשימה), איבר ייקרא פסגה אם הוא גדול (ממש) מהשכן שקיים לו (שמאלי או ימני, בהתאם לקצה).

בשאלה זו עליכם להשלים הפונקציה peaks הבאה, המקבלת רשימה lst, ומחזירה את כמות הפסגות שיש בה. להלן דוגמאות הרצה:

```
>>> peaks([1,2,1,2,1])
2
>>> peaks([1,1,1,2,1])
1
>>> peaks([1,2,3,4,5])
1
>>> peaks([3,2,3,4,5])
2
>>> peaks([3,5,3,4,5])
2
```

השלימו את הפונקציה:

```
def peaks(lst):
```

שאלה 5 (20 נק')

כזכור, חיפוש בינארי מניח שהרשימה בה מתבצע החיפוש ממוינת. לנוחיותכם מופיע בהמשך הקוד לחיפוש בינארי. נניח שמבצעים חיפוש בינארי על רשימה שאיננה ממוינת. נקרא למספר שמחפשים ברשימה x . לכל אחד מהמצבים הבאים: סמנו בעיגול האם הוא אפשרי או לא. אם לדעתכם הוא אפשרי, תנו דוגמה לרשימה לא ממוינת עם 5 מספרים, ומספר נוסף x שכאשר יינתנו לפונקציה לחיפוש בינארי יתרחש המצב המתואר (אין צורך להסביר במקרה זה). אם לדעתכם המצב איננו אפשרי, הסבירו מדוע.

מצב 1: הפונקציה מחזירה False עבור x שנמצא ברשימה.

אפשרי / לא אפשרי

מצב 2: הפונקציה מחזירה True עבור x שלא נמצא ברשימה.

אפשרי / לא אפשרי

מצב 3: הפונקציה מחזירה False עבור x שלא נמצא ברשימה.

אפשרי / לא אפשרי

```
def binary_search(my_list, x):
    ''' search for x in my_list, which MUST BE SORTED !! '''
    left=0
    right=len(my_list)-1

    while left<=right:
        mid = (left+right)//2
        if my_list[mid]==x:
            return True
        elif my_list[mid] < x: #go to right half
            left = mid+1
        else:                  #go to left half
            right = mid-1
    return False #if we got here the search failed
```

שאלה 6 (10 נק')

להלן בעיית הכרעה: **בהינתן מחרוזת, האם היא פלינדרום?**

תזכורת: פלינדרום היא מחרוזת שנקראת באותו אופן משמאל ומימין. למשל: "ABBA".

להלן האלגוריתם שראינו בכיתה לבדיקה האם מחרוזת נתונה היא פלינדרום:

```
def is_palindrome(st):  
    n = len(st)  
    for i in range(n//2): #check if n//2 is enough  
        if st[i] != st[n-1-i]:  
            return False #not a palindrome  
    return True #if we for here, st IS a palindrome
```

ליד כל טענה סמנו בעיגול האם הטענה נכונה או לא נכונה, והסבירו בקצרה. אנו מניחים כמובן כי $P \neq NP$.

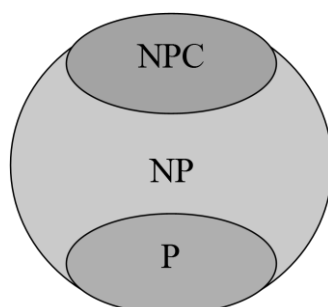
א. הבעיה שייכת ל- P (מחלקת הבעיות שיש להן פתרון בזמן פולינומי). נכון / לא נכון

נימוק:

ב. הבעיה שייכת ל- NP (מחלקת הבעיות שיש להן אלגוריתם אימות לפתרון חוקי נתון בזמן פולינומי). נכון / לא נכון

נימוק:

לנוחיותכם, האיור הבא שנלמד והוסבר בכיתה:



שאלה 7 (15 נק')

להלן הפונקציה Caesar_decrypt_try_all_auto שראינו בכיתה, לפענוח צופן קיסר:

```
1 def Caesar_decrypt_try_all_auto(encryption, common, th):
2     """ find offsets in which encryption has >=th words from dictionary """
3     for offset in range(1,128):
4         match = 0
5         possible_text = Caesar_decrypt(encryption, offset)
6         for w in common:
7             match += possible_text.count(w)
8         if match >= th:
9             print(offset, possible_text)
```

מחליפים את שורה 7 בשתי השורות הבאות (כמו קודם, בתוך לולאת ה-for):

```
7.1         if w in possible_text
7.2             match += 1
```

הסבירו מה שונה בפונקציה לעומת הגרסה המקורית שלה לפני השינוי, וצינו יתרון אחד וחסרון אחד בשינוי זה.

תשובה:

סוף!