ת.ז.: _____ מס' מחברת: _____

מבחן בקורס CS 1001.py

סמסטר ב' התשע"א, מועד א'

תאריך: 12.7.2011

מרצה: פרופ' בני שור

מתרגל: רני הוד

מומלץ לקרוא את כל ההנחיות והשאלות בתחילת המבחן, לפני תחילת כתיבת התשובות.

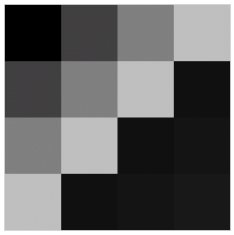
- משך הבחינה שלוש שעות.
- חומר עזר מותר: שני דפי A4, כתובים משני הצדדים.
- במבחן חמש שאלות שלחלקן סעיפי משנה. כדי לקבל ציון 100 בבחינה יש לענות נכונה על כל השאלות. ניקוד כל סעיף מצוין לידו. אין בהכרח קשר בין ניקוד הסעיף ובין קושיו.
  - את התשובות לשאלה הסגורה יש לרשום בעמוד הראשון במחברת בצורה ברורה. לכל סעיף יש לרשום את מספרו בצירוף אות בודדת (א', ב', ג' או ד') ואין להוסיף הסבר.
  - על התשובה לשאלות הפתוחות להופיע במחברת, כל אחת בדף נפרד. יש לענות תשובות ברורות ותמציתיות. תשובות מסורבלות או לא ניתנות לקריאה ע"י הבודקים יזכו לניקוד חלקי בלבד.
- על סעיף של שאלה <u>פתוחה</u> ניתן לענות *"אינני יודע/ת"* כתשובה; על סעיף זה יינתנו 20% מהנקודות. במקרה זה <u>אין להוסיף שום הסבר</u>.
- בפתרון סעיף בשאלה מותר להשתמש בתוצאות הסעיפים הקודמים, גם אם לא פתרתם אותם.

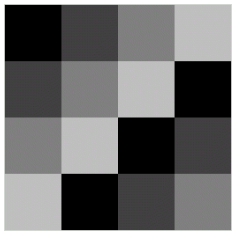
בהצלחה!

שאלה 1 (10 נק' לכל סעיף; סה"כ 20 נק')

1. <u>תזכורת</u>: בכיתה ייצגנו תמונה ע"י מטריצה שבכל תא שלה ערך מ-`range(256)` המייצג את עוצמת האור של הפיקסל המתאים.
להלן תמליל קצר והתמונות[1] שמוצגות בעקבותיו.

```
Python 3.1.2 (r312:79147, Apr 15 2010, 15:35:48)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from matrix import Matrix
>>> A=Matrix(4); B=Matrix(4) # 4-by-4, initially zero
>>> for i in range(4):
        for j in range(4):
                A[i,j]=(i*64+j*64) % 256
                B[i,j]=(i*64+j*64)
>>> A.display(zoom=64); B.display(zoom=64)
```





בני תוהה מה קורה כשפיקסל מקבל ערך $m > 255$; בחרו את האפשרות הנכונה.

(א) מוצג במקומו הפיקסל שערכו $(m \,\%\, 256)$.

(ב) מוצג במקומו הפיקסל שערכו 255.

(ג) מוצג במקומו הפיקסל שערכו 0.

(ד) תשובות א'–ג' אינן נכונות.

[1] כל אחת מהתמונות היא 4×4 פיקסלים והן מוצגות בהגדלה של פי 64. לא נתון לכם איזו מהתמונות היא `A` ואיזו היא `B`.

2. להלן תמליל קצר ובו וריאציה על האלגוריתם binary_gcd עם שתי דוגמאות הרצה.

```
Python 3.1.2 (r312:79147, Apr 15 2010, 15:35:48)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> def quart_gcd(a, b):
        a = abs(a)
        b = abs(b)
        k = 0
        while a != 0 and b != 0:
                if a%4 == b%4 == 0:
                        k += 1
                        a //= 4
                        b //= 4
                elif a%4 != 0 and b%4 == 0:
                        b //= 4
                elif a%4 == 0 and b%4 != 0:
                        a //= 4
                else: # both a%4, b%4 != 0
                        a,b = abs(a-b), min(a,b)
        return 4**k * (a+b)

>>> quart_gcd(3, 15)
3
>>> quart_gcd(28,32)
4
```

בחרו את האפשרות הנכונה.

(א) לכל זוג שלמים a, b המקיימים gcd(a,b) == 1 מתקיים quart_gcd(a,b) == 1.

(ב) לכל זוג שלמים a, b המקיימים quart_gcd(a,b) == 1 מתקיים gcd(a,b) == 1.

(ג) לכל זוג שלמים a, b מתקיים gcd(a,b) == quart_gcd(a,b).

(ד) תשובות א'–ג' אינן נכונות.

שאלה 2 (20 נק')

יהופץ וצפניה משחקים במשחק הבא: צפניה בוחר מספר סודי x מתוך range(N) (עבור N ידוע מראש) ויהופץ צריך לנחש אותו ע"י שאילת שאלות, כל אחת מהצורה "האם המספר הסודי הוא k?". אם יהופץ ניחש נכונה אז המשחק מסתיים בקול תרועה ואחרת צפניה עונה "גדול" אם $x > k$ ו"קטן" אם $x < k$. צפניה עבר השתלמות בשפת הנבאחו ולכן תשובותיו הן "אנילצו" ו"אלציסי", אך יהופץ (וגם אתם...) לא יודעים האם "אנילצו" זה "גדול" או "קטן".

כיתבו פונקציה בשם guessing_game שמקבלת כקלט את המספר הטבעי N ומחזירה את המספר הסודי. על הפונקציה לשאול כמה שפחות שאלות (במקרה הגרוע ביותר). שאילת שאלה ממומשת ע"י קריאה לפונקציה binabidilkid שמקבלת כקלט את k ומחזירה אחת מבין המחרוזות 'aoo', 'altsisi', 'aniltso'.[2]

הסבירו ראשית את האלגוריתם במילים, כיתבו אותו בקוד פייתון המבצע קריאות לפונקציה הנתונה ונתחו את מספר השאלות במקרה הגרוע ביותר.

[2] בשפת הנבאחו, aoo פירושו 'כן'; הפונקציה מחזירה ערך זה במקרה של ניחוש קולע.

שאלה 3 (5 נק' לסעיפים 1,3 ו־10 נק' לסעיף 2; סה"כ 20 נק')

דחיסת Ziv–Lempel, בגרסה שראינו בכיתה, נעשתה באופן חמדני: התאמה באורך $k \geq 3$ מדווחת מיידית.

```
Python 3.1.2 (r312:79147, Apr 15 2010, 15:35:48)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> def maxmatch(T,p,w=2**12-1,max_length=2**5-1):
        assert isinstance(T,str)
        n=len(T)
        maxmatch=offset=0
        for m in range(1,min(p+1,w)):
                for k in range(min(max_length,n-p)):
                        if T[p-m+k] != T[p+k]: break
                else:
                        k += 1
                if maxmatch<k:
                        maxmatch=k
                        offset=m
        return offset,maxmatch

>>> def lz77_compress(text,w=2**12-1,max_length=2**5-1):
        result=[]
        n=len(text)
        p=0
        while p<n:
                if ord(text[p])>=128: continue
                m,k = maxmatch(text,p,w,max_length)
                if k<3:
                        result.append(text[p])
                        p+=1
                else:
                        result.append([m,k])
                        p+=k
        return result
```

בשאלה זו נרצה להראות שהחמדנות אינה אופטימלית. נזכיר כי אצלנו אורכו של קידוד תו ascii בודד במחרוזת הדחוסה הוא $1+7=8$ ביטים ושל קידוד חזרה הוא $1+12+5=18$ ביטים.

1. תנו דוגמא בה החמדנות אינה משתלמת. ציינו את הטקסט, את רשימת הזוגות שמוחזרים ע"י maxmatch עבור האלגוריתם החמדן, את אורך הקידוד, מדוע קיים קידוד (לא חמדן) קצר יותר ואת אורכו.
2. הסבירו (במילים) כיצד ניתן לתקן את האלגוריתם כך שנקבל דחיסה שהיא לעיתים משופרת (למשל בדוגמא שהבאתם לעיל) ולעולם לא נחותה מאשר זו המתקבלת ע"י האלגוריתם החמדן.
3. יישמו את השינוי הנדרש בקוד של הפונקציה lz77_compress. אין לשנות את maxmatch.

שאלה 4 (20 נק')

להלן תקציר חוקי הניקוד במשחק הבאולינג. המשחק מורכב מ־10 מסגרות. בתחילת כל מסגרת ניצבים על המסלול 10 פינים, ולשחקן יש שתי זריקות כדי לנסות להפיל את כולם.
מסגרת שבה כל עשרת הפינים הופלו בזריקה הראשונה נקראת strike, ואין זריקה נוספת במסגרת זו; מסגרת שבה כל העשרה הופלו בתום הזריקה השנייה נקראת spare.
הניקוד הבסיסי של זריקה הוא כמות הפינים שהופלו בה. הניקוד של מסגרת strike הוא הניקוד הבסיסי שלה עצמה (10) ועוד הניקוד הבסיסי של שתי הזריקות העוקבות. הניקוד של מסגרת spare הוא הניקוד הבסיסי שלה עצמה (10) ועוד הניקוד הבסיסי של זריקה עוקבת אחת. הניקוד של כל זריקה אחרת הוא הניקוד הבסיסי שלה עצמה. בכל מקרה ניקוד המשחק הוא סכום הניקוד של 10 המסגרות הראשונות, אבל מתווספות מסגרות נוספות אם יש צורך בהן לחישוב נקודות הבונוס עבור strike או spare.
כיתבו פונקציה בשם bowling_score שמקבלת כקלט רשימה ובה כמות הפינים שנפלו בכל זריקה [3] ומחזירה את ניקוד המשחק. הסבירו ראשית במילים כיצד הפונקציה פועלת ולאחר מכן ממשו אותה בקוד פייתון.

דוגמאות:

```
>>> bowling_score([10] * 12)       # the perfect game:
300                                #  each strike is worth 10+10+10
>>> bowling_score([10,0,0] * 4)    # a "wasteful" game:
40                                 #  each strike is worth only 10+0+0
>>> bowling_score([6,4]*10 + [6]) # not a bad game:
160                                #  each spare is worth 4+6
>>> bowling_score([1,4,4,5,6,4,5,5,10,0,1,7,3,6,4,10,2,8,6])
133
```

שאלה 5 (10 נק' לכל סעיף; סה"כ 20 נק')

1. כיתבו גנרטור בשם merge_sorted שמקבל כקלט זוג איטרטורים, העוברים כ"א על סדרה ממוינת (לא יורדת) אינסופית לא חסומה, ומחזיר איטרטור שעובר על האיחוד הממוין עם חזרות של הסדרות.

דוגמא:

```
>>> def arithmetic(a,d):
        while True:
                yield a
                a += d

>>> it = merge_sorted(arithmetic(5,5), arithmetic(0,3))
>>> list(islice(it, 12))
[0, 3, 5, 6, 9, 10, 12, 15, 15, 18, 20, 21]
```

2. כיתבו גנרטור בשם merge_sorted_unique שמקבל כקלט זוג איטרטורים, העוברים כ"א על סדרה ממוינת (לא יורדת) אינסופית לא חסומה, ומחזיר איטרטור שעובר על האיחוד הממוין ללא חזרות של הסדרות.

דוגמא:

```
>>> it = merge_sorted_unique(arithmetic(5,5), arithmetic(0,3))
>>> list(islice(it, 12))
[0, 3, 5, 6, 9, 10, 12, 15, 18, 20, 21]
```

[3] ניתן להניח כי הרשימה מתארת משחק חוקי בעל מספר מסגרות מתאים, ואין צורך לבדוק זאת.