

```
def count_black_pixels(im):
    n,m = im.dim()
    blacks_count = 0
```



```
return blacks_count
```

שאלה 5 (20 נק')

שאלה זאת עוסקת בפונקציות גנרטור.

תזכורת: קריאה לפונקציית גנרטור fgen מחזירה גנרטור (סוג של איטרטור), gen, וכל קריאה ל next(gen) מחזירה ערך כלשהו.

נאמר שפונקציית הגנרטור fgen מייצרת ערך x אם x מתקבל (מוחזר) אחרי מס' סופי של קריאות ל next(gen).

נאמר ש fgen מייצרת קבוצה (אולי אינסופית) של ערכים אם היא מייצרת כל ערך בקבוצה.

בשאלה זו אין חשיבות לסדר ייצור הערכים ע"י הגנרטור

א. רוני התבקש לכתוב פונקציית גנרטור שמייצרת את כל (אינסוף) הזוגות הסדורים של מספרים טבעיים (i,j) , ללא חשיבות לסדר ייצורם וללא חזרות. שימו לב שהזוגות הם סדורים, כלומר $(2,3)$ שונה מ- $(3,2)$, והזוגות הנדרשים כוללים גם זוגות בהם $i=j$ (למשל הזוג $(2,2)$). הוא כתב את הקוד הבא:

```
def AllPairs():
    i=0
    while True:
        j=0
        while True:
            yield(i,j)
            j=j+1
        i=i+1
```

בקוד שלעיל ישנה בעיה. מהי? (סמנו בעמוד הבא את התשובה הנכונה ונמקו בקצרה).

1. קריאה ל- `AllPairs()` תגרום ללולאה אינסופית והתוכנית לא תעצור
 2. קריאה ל- `next` על הגנרטור שמוחזר מ- `AllPairs()` תגרום ללולאה אינסופית והתוכנית לא תעצור
 3. פונקצית הגנרטור מייצרת חלק מהזוגות הדרושים אך לא את כולם
 4. זוגות מסוימים מיוצרים יותר מפעם אחת
 5. אחרי מס' כלשהו של קריאות רצופות ל- `next`, קריאה נוספת ל- `next` תגרום ללולאה אינסופית והתוכנית לא תעצור
- הסבר קצר:

- ב. השלימו את הקוד של פונקציית הגנרטור הבאה שמייצרת את כל הזוגות הסדורים של מספרים טבעיים (i,j) המקיימים $j < i$.

```
def SomePairs():
    i=0
    while True:
```

- ג. השלימו את הקוד של פונקציית הגנרטור `RevGen`. הקלט שלה הוא פונקציית גנרטור `PairsGen` שמייצרת מס' אינסופי של זוגות. `RevGen` מייצרת את כל הזוגות (i,j) , שעבורם מתקיים שהזוג (j,i) מיוצר ע"י `PairsGen`.

```
def RevGen(PairsGen):
```

ד. כעת נתקן את הקוד של רוני כך ש- `AllPairs()` ייצר את כל הזוגות (i,j) כנדרש. השתמשו בפונקציות שכתבתם בסעיפים קודמים, וכן בשתי פונקציות עזר:

- פונקציית גנרטור בשם `EqPairs` שמייצרת את כל הזוגות של מספרים טבעיים מהצורה (i,i) (כלומר זוגות עבורם שני איברי הזוג שווים). אין צורך לממש פונקציה זו.

- פונקציית גנרטור בשם `UnionGenerators`. פונקציה זו מקבלת כקלט שני גנרטורים **אינסופיים** שמייצרים קבוצות איברים **זרות**. הפונקציה מייצרת את קבוצת האיחוד של קבוצות איברים אלה.

(1) השלימו את הקוד של `UnionGenerators`:

```
def UnionGenerators (gen1, gen2):
    while True:
```

(2) השלימו את הקוד של `AllPairs` תוך שימוש בפונקציות שכתבתם בסעיפים הקודמים ובפונקציות העזר.

```
def AllPairs():
```

סוף!