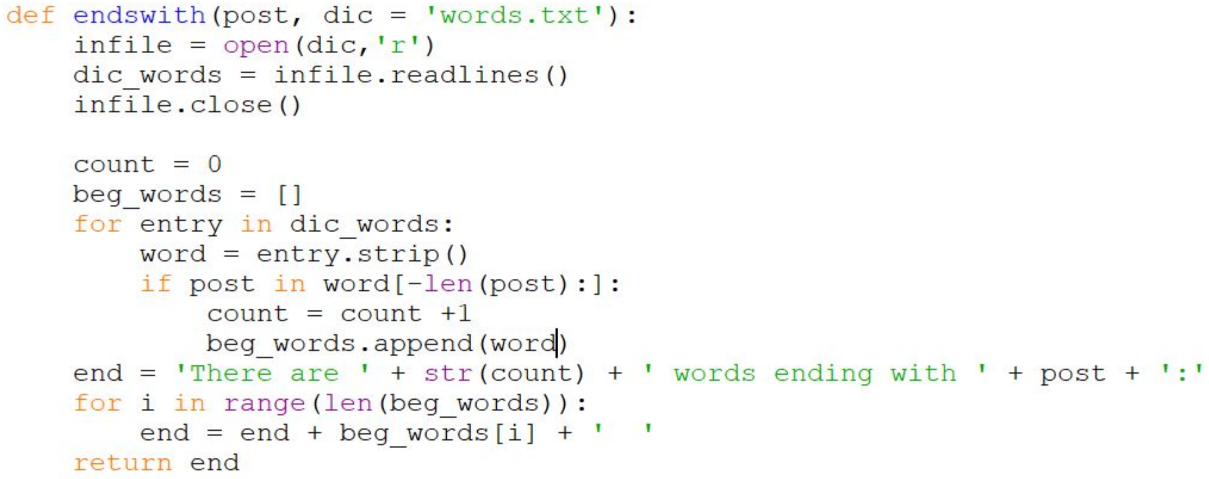
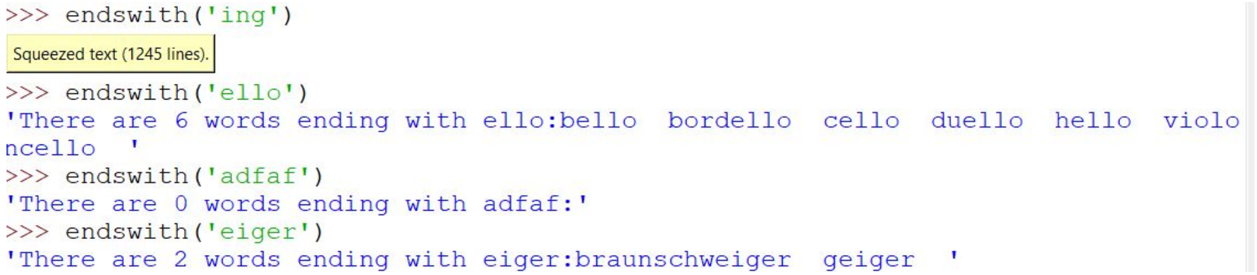
2.

a,



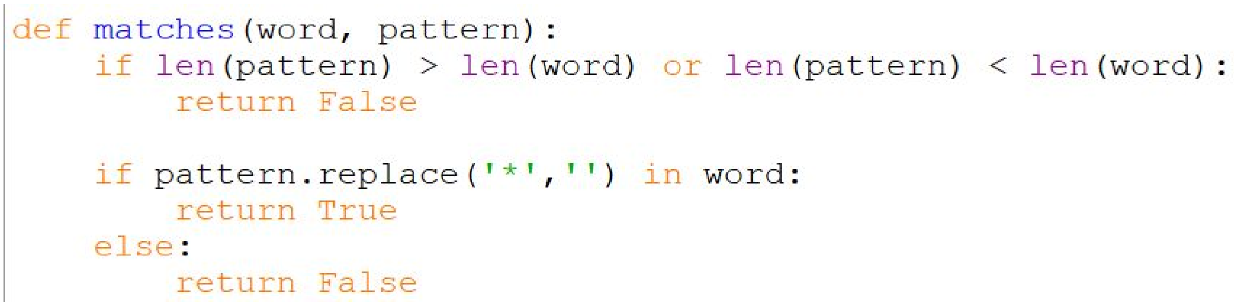
```python
def endswith(post, dic = 'words.txt'):
    infile = open(dic,'r')
    dic_words = infile.readlines()
    infile.close()

    count = 0
    beg_words = []
    for entry in dic_words:
        word = entry.strip()
        if post in word[-len(post):]:
            count = count +1
            beg_words.append(word)
    end = 'There are ' + str(count) + ' words ending with ' + post + ':'
    for i in range(len(beg_words)):
        end = end + beg_words[i] + '  '
    return end
```

Tests:



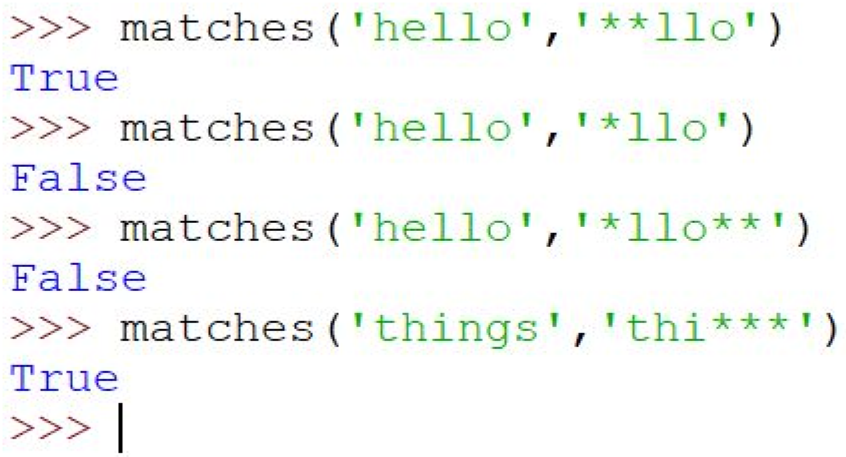
```
>>> endswith('ing')
Squeezed text (1245 lines).
>>> endswith('ello')
'There are 6 words ending with ello:bello  bordello  cello  duello  hello  violo
ncello  '
>>> endswith('adfaf')
'There are 0 words ending with adfaf:'
>>> endswith('eiger')
'There are 2 words ending with eiger:braunschweiger  geiger  '
```

B.



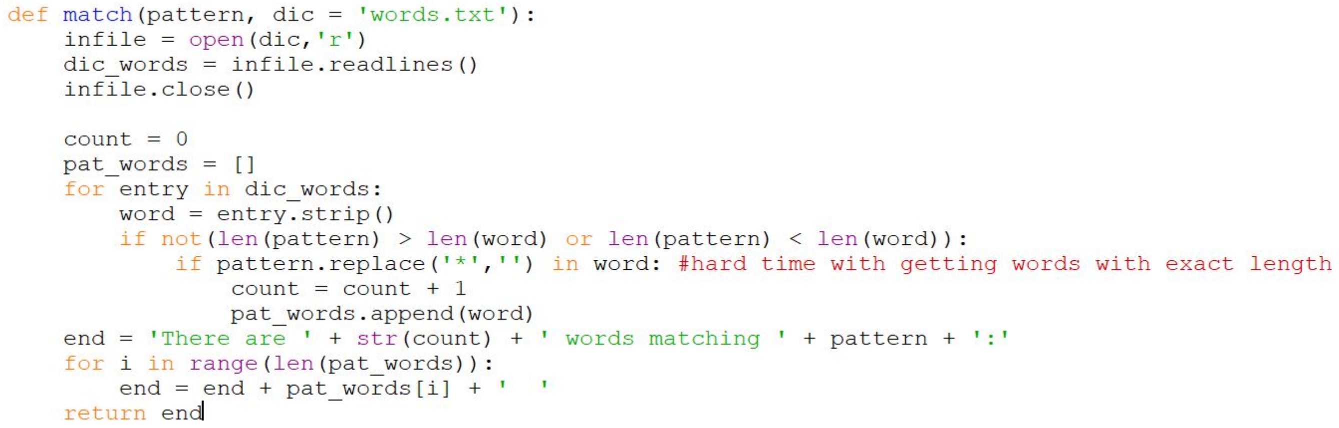
```python
def matches(word, pattern):
    if len(pattern) > len(word) or len(pattern) < len(word):
        return False

    if pattern.replace('*','') in word:
        return True
    else:
        return False
```

Test:



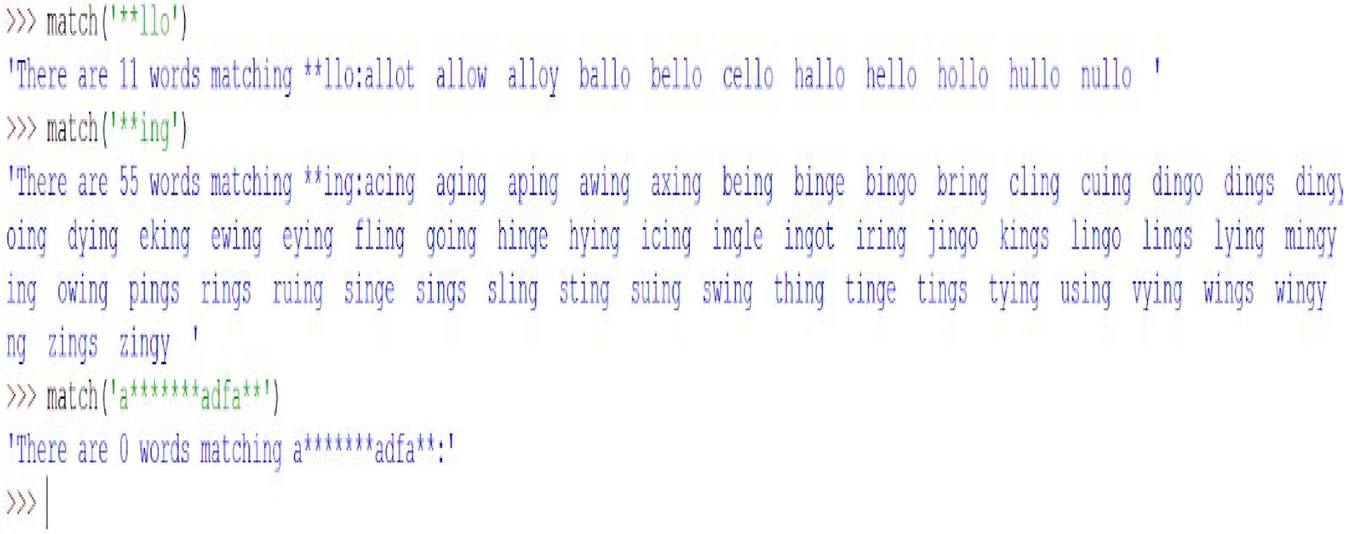
```
>>> matches('hello','**llo')
True
>>> matches('hello','*llo')
False
>>> matches('hello','*llo**')
False
>>> matches('things','thi***')
True
>>> 
```

C.



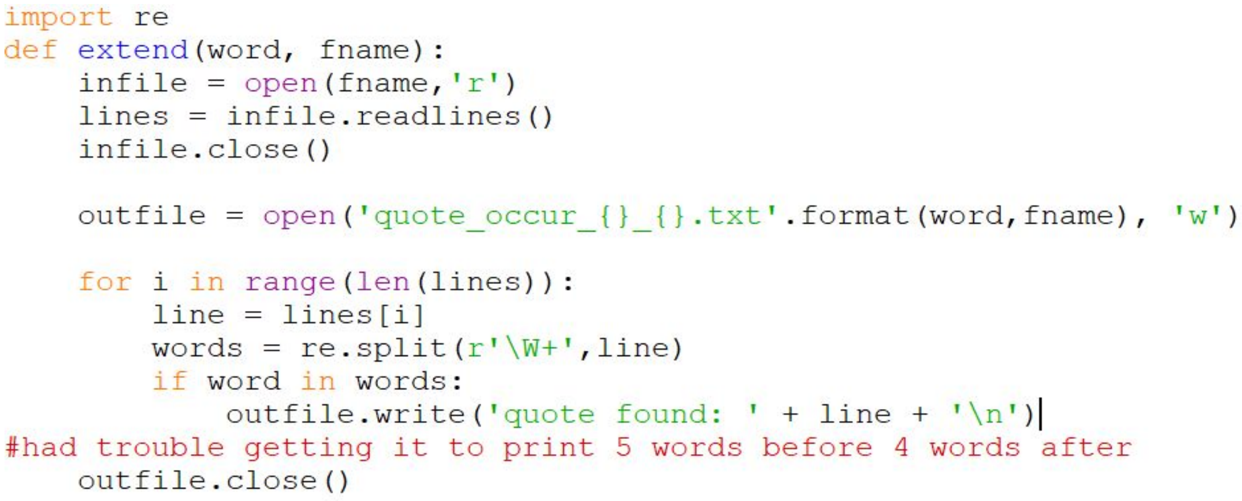
```
def match(pattern, dic = 'words.txt'):
    infile = open(dic,'r')
    dic_words = infile.readlines()
    infile.close()

    count = 0
    pat_words = []
    for entry in dic_words:
        word = entry.strip()
        if not(len(pattern) > len(word) or len(pattern) < len(word)):
            if pattern.replace('*','') in word: #hard time with getting words with exact length
                count = count + 1
                pat_words.append(word)
    end = 'There are ' + str(count) + ' words matching ' + pattern + ':'
    for i in range(len(pat_words)):
        end = end + pat_words[i] + '  '
    return end
```

Tests:

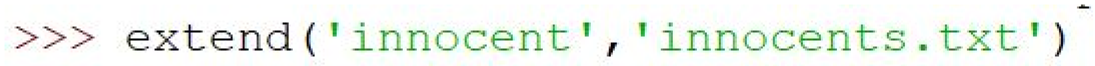


```
>>> match('**llo')
'There are 11 words matching **llo:allot  allow  alloy  ballo  bello  cello  hallo  hello  hollo  hullo  nullo  '
>>> match('**ing')
'There are 55 words matching **ing:acing  aging  aping  awing  axing  being  binge  bingo  bring  cling  cuing  dingo  dings  dingy
oing  dying  eking  ewing  eying  fling  going  hinge  hying  icing  ingle  ingot  iring  jingo  kings  lingo  lings  lying  mingy
ing  owing  pings  rings  ruing  singe  sings  sling  sting  suing  swing  thing  tinge  tings  tying  using  vying  wings  wingy
ng  zings  zingy  '
>>> match('a*******adfa**')
'There are 0 words matching a*******adfa**:'
>>> 
```

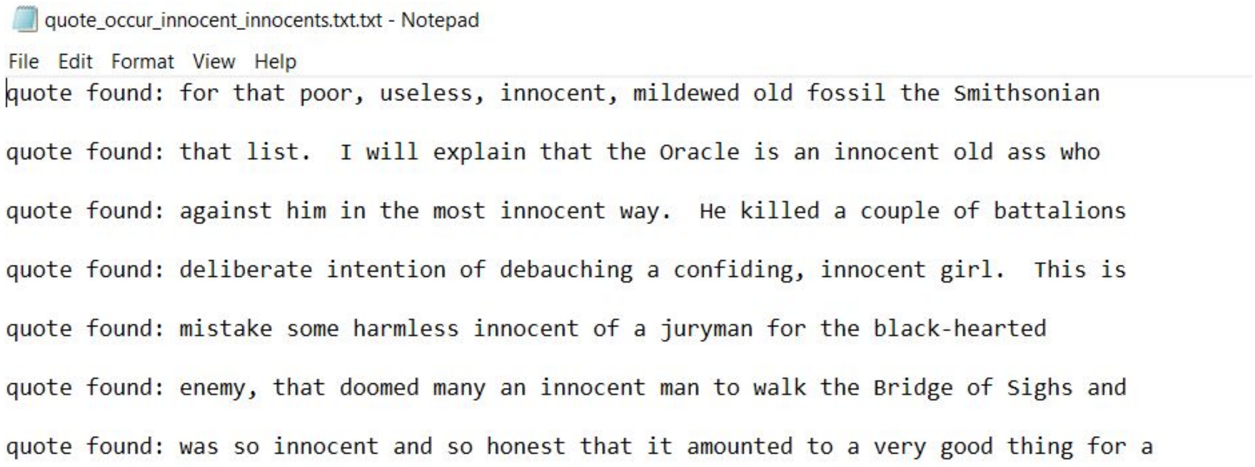
3.



```python
import re
def extend(word, fname):
    infile = open(fname,'r')
    lines = infile.readlines()
    infile.close()

    outfile = open('quote_occur_{}_{}.txt'.format(word,fname), 'w')

    for i in range(len(lines)):
        line = lines[i]
        words = re.split(r'\W+',line)
        if word in words:
            outfile.write('quote found: ' + line + '\n')
#had trouble getting it to print 5 words before 4 words after
    outfile.close()
```



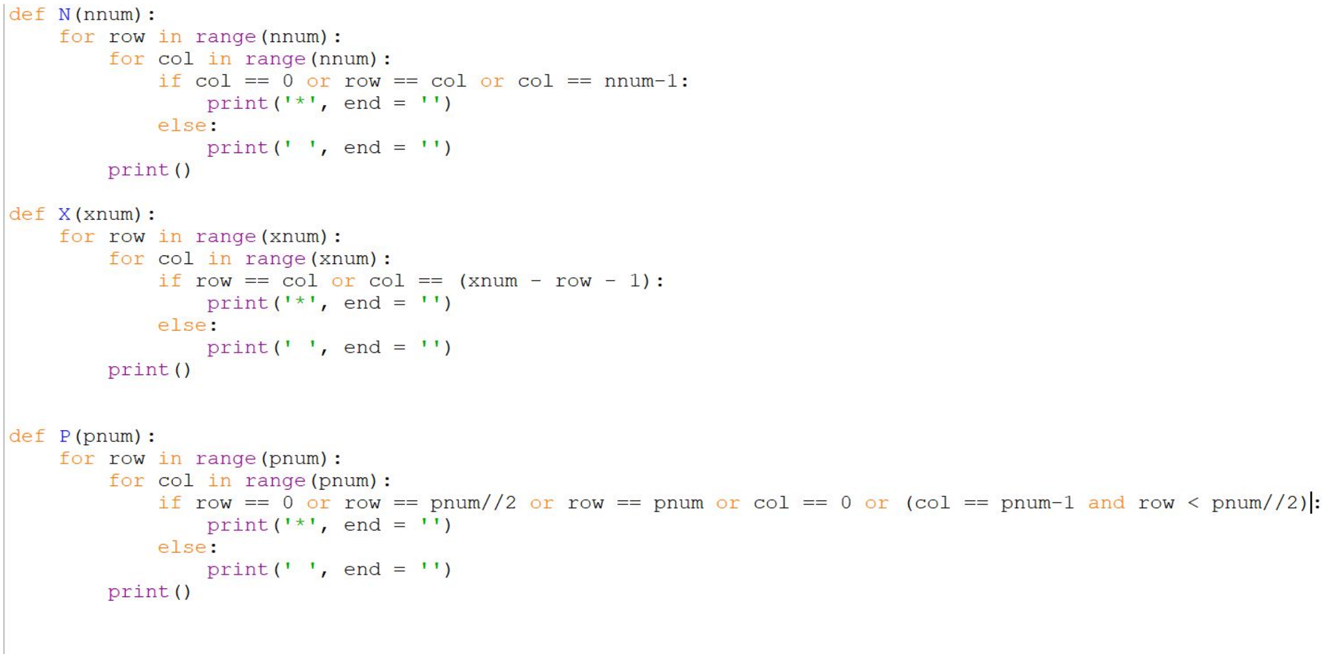
```
>>> extend('innocent','innocents.txt')
```



quote_occur_innocent_innocents.txt.txt - Notepad
File Edit Format View Help

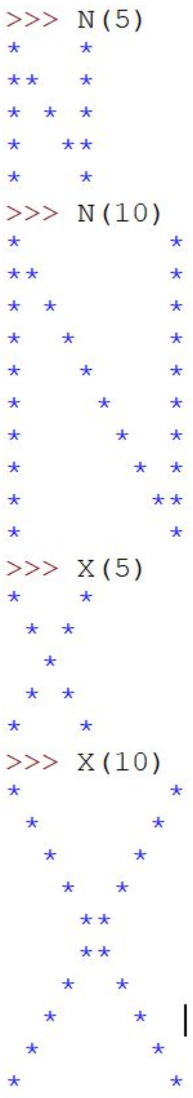
```
quote found: for that poor, useless, innocent, mildewed old fossil the Smithsonian

quote found: that list.  I will explain that the Oracle is an innocent old ass who

quote found: against him in the most innocent way.  He killed a couple of battalions

quote found: deliberate intention of debauching a confiding, innocent girl.  This is

quote found: mistake some harmless innocent of a juryman for the black-hearted

quote found: enemy, that doomed many an innocent man to walk the Bridge of Sighs and

quote found: was so innocent and so honest that it amounted to a very good thing for a
```

4.

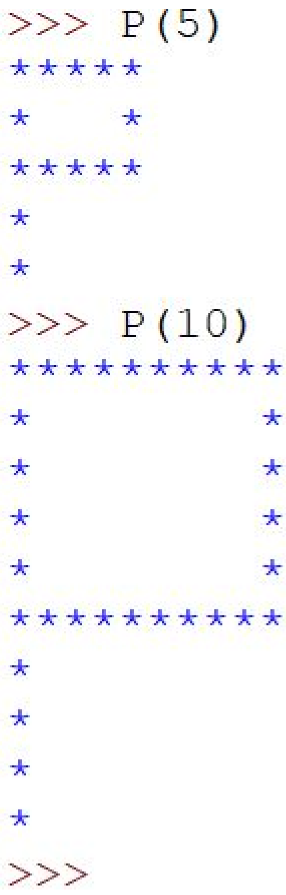


```
def N(nnum):
    for row in range(nnum):
        for col in range(nnum):
            if col == 0 or row == col or col == nnum-1:
                print('*', end = '')
            else:
                print(' ', end = '')
        print()

def X(xnum):
    for row in range(xnum):
        for col in range(xnum):
            if row == col or col == (xnum - row - 1):
                print('*', end = '')
            else:
                print(' ', end = '')
        print()


def P(pnum):
    for row in range(pnum):
        for col in range(pnum):
            if row == 0 or row == pnum//2 or row == pnum or col == 0 or (col == pnum-1 and row < pnum//2):
                print('*', end = '')
            else:
                print(' ', end = '')
        print()
```

Tests:



```
>>> N(5)
*   *
**  *
* * *
*  **
*   *
>>> N(10)
*        *
**       *
* *      *
*  *     *
*   *    *
*    *   *
*     *  *
*      * *
*       **
*        *
>>> X(5)
*   *
 * *
  *
 * *
*   *
>>> X(10)
*        *
 *      *
  *    *
   *  *
    **
    **
   *  *
  *    *
 *      *
*        *
```



```
>>> P(5)
*****
*   *
*****
*
*
>>> P(10)
**********
*        *
*        *
*        *
*        *
**********
*
*
*
*
>>>
```