

NAME:  
EMAIL:  
SIGNATURE:  
CIRCLE COURSE SECTION: MW 1-3 TTh 11-1 TTh 4-6

**Lehman College, CUNY**  
**CIS 166 Final Exam, Version 1, Spring 2014**

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")  
days = s[:-1].split("s")  
print("There are", num, "fun days in a week")  
print("Two of them are",days[0], days[-1])  
result = ""  
for i in range(len(days[0])):  
    if i > 2:  
        result = result + days[0][i]  
print("My favorite", result, "is Saturday.")
```

2. Define a Python function named `calculate_tax` which accepts one parameter, `income`, and returns the income tax. Income is taxed according to the following rule: the first \$200,000 is taxed at 25% and any remaining income is taxed at 50%. For example, `calculate_tax(100000)` should return  $100,000 \times 0.25 = 25,000$ , while `calculate_tax(300000)` should return  $200,000 \times 0.25 + 100,000 \times 0.5 = 100,000$ .

3. Complete the following program. That is, write the functions `getInputs()`, `countWord()`, `average()`, and `printSummary()`:

```
def main():
    fname, word = getInputs()    #get the file name and word to be searched
    infile = open(fname, "r")    #open the file for reading
    resultList = list()         #initialize result list to empty list

    for line in infile:
        num = countWord(line, word) #return the number of
                                    #times word occurs in line
        resultList.append(num)

    a = average(resultList)      #compute the average number of
                                #times word appears per line
    printSummary(word, a)       #print the average (including explanation)
```

4. Given the following function definitions:

```
def bar(n):  
    if n <= 8:  
        return 1  
    else:  
        return 0  
  
def foo(l):  
    n = bar(l[-1])  
    return l[n]
```

(a) What does `foo([1,2,3,4])` return?

(b) What does `foo([1024,512,256,128])` return?

5. Given the following code:

```
file = open("numbers.txt")
total = 0
for line in file.readlines():
    for strnum in line.split(","):
        num = int(strnum)
        if num % 2 == 0:
            total = total + num
        print(total)
```

(a) What will the output be for this `numbers.txt`?

**numbers.txt:**

1,2,3,4,5,6

(b) What will the output be for this `numbers.txt`?

**numbers.txt:**

123456

6. Draw what will be displayed in the graphics window when the following program is executed. Remember to indicate the final position and direction of the turtle at the end of program. (The turtle always points to the right of the screen at the start of the program.)

```
from turtle import *

def mystery(t, n, d):
    for i in range(n):
        if d == 'r':
            t.right(360/n)
        else:
            t.left(360/n)
        t.forward(50)
```

**Graphics Displayed:**

```
def draw(t, n):
    t.forward(100)
    mystery(t, n, 'r')
    mystery(t, n, 'l')
```

```
t = Turtle()
draw(t, 4)
```

7. Write a **program** that reads in a text file, `infile.txt`, and prints out the lines containing the phrase:  
`The Amazing Spider Man` (that is, the line must contain all four words in this order):

8. Write the python code for the algorithms below:

```
(a) find(st)
    set index to 0
    set location to -1
    set found to false
    while not found
        if st[index] equals ','
            set location to index
            set found to true
        increment index
    return location
```

```
(b) getSmaller(ls)
    for each item in ls
        if current item is less than first item in ls
            switch first item and current item in ls
```

9. Given the following input file `twitter-followers.dat`, write a program that reads in the input file and prints the name of the celebrity followed by the word **Amazing** for each celebrity with over 50 million followers, the word **Sweet** for celebrities with followers between 40 and 50 millions and the word **Good** for all others.

**twitter-followers.dat**

(Celebrity, Followers in Millions)

Katy Perry, 52  
Justin Bieber, 51  
Barack Obama, 42  
Lady Gaga, 41  
Taylor Swift, 40  
Britney Spears, 37  
Rihanna, 35  
Justin Timberlake, 32  
Ellen DeGeneres, 28

10. Given the following code:

```
def getArray():
    input = [
        "Oppenheimer,Robert",
        "Fermi,Enrico",
        "Feynman,Richard",
        "Teller,Edward",
        "Frisch,Otto",
        ",Zazzles"
    ]
    return input
def main():
    array = getArray()
    last = ""
    for line in array:
        last = line.split(",")[1]
    print(last)
main()
```

And given following CSV file labeled `cats.txt`:

```
Oppenheimer,Robert
Fermi,Enrico
Feynman,Richard
Teller,Edward
Frisch,Otto
,Zazzles
```

- (a) To make the program easier to update, rewrite the `getArray()` function so that it retrieves the input from the CSV file. The updated program should print the same output when run.

- (b) What is the output of the program?

**Useful String Methods:** (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

**Useful Turtle Methods:** (from <http://docs.python.org/3.0/library/turtle.html>)

Function	Meaning
<code>forward(d)</code>	Move turtle forward <code>d</code> steps
<code>backward(d)</code>	Move turtle backward <code>d</code> steps
<code>right(angle)</code>	Turn turtle <code>angle</code> degrees to the right
<code>left(angle)</code>	Turn turtle <code>angle</code> degrees to the left
<code>up()</code>	Pull the pen up no drawing when moving
<code>down()</code>	Pull the pen down drawing when moving

NAME:  
EMAIL:  
SIGNATURE:  
CIRCLE COURSE SECTION:    MW 1-3    TTh 11-1    TTh 4-6

**Lehman College, CUNY**  
**CIS 166 Final Exam, Version 2, Spring 2014**

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
s = "marchxoctoberxjanuaryxaugustx"  
num = s.count("x")  
items = s[:-1].split("x")  
result = ""  
for item in items:  
    print( item.capitalize() )  
    result = result + item[0].upper()  
print( (result[0:2] + "NTHS!! ") * 3, end="")
```

2. Define a Python function named `calculate_tax` which accepts one parameter, `income`, and returns the income tax. Income is taxed according to the following rule: the first \$100,000 is taxed at 25% and any remaining income is taxed at 50%. For example, `calculate_tax(80000)` should return  $80,000 \times 0.25 = 20,000$ , while `calculate_tax(200000)` should return  $100,000 \times 0.25 + 100,000 \times 0.5 = 75,000$ .

3. Complete the following program—that is, write the functions `getInputs()`, `countAs()`, `average(l)`, and `printSummary(a)`:

```
def main():
    fname = getInputs()          #get the file name
    infile = open(fname, "r")    #open the file for reading
    resultList = list()         #initialize result list to empty list

    for line in infile:
        num = countAs(line)     #return the number of 'a' and 'A' in line
        resultList.append(num)

    a = average(resultList)     #compute the average number of
                                #times 'a' or 'A' appears per line
    printSummary(a)            #print the average (including explanation)
```

4. Given the following function definitions:

```
def bar(n):  
    if n >= 32:  
        return 2  
    else:  
        return 1  
  
def foo(l):  
    n = bar(l[2])  
    return l[n]
```

(a) What does `foo([1,2,3,4])` return?

(b) What does `foo([1024,512,256,128])` return?

5. Given the following code:

```
file = open("numbers.txt")
total = 0
for line in file.readlines():
    for strnum in line.split(","):
        num = int(strnum)
        if num % 2 == 0:
            total = total + num
        print(total)
```

(a) What will the output be for this `numbers.txt`?

**numbers.txt:**

10,11,12,13,14

(b) What will the output be for this `numbers.txt`?

**numbers.txt:**

1011121314

6. Draw what would be displayed in the graphics window when the following program is executed. Remember to indicate the final position and direction of the turtle at the end of program. (The turtle always points to the right of the screen at the start of the program.)

```
from turtle import *

def mystery(t, n, d):
    for i in range(n):
        if d == 'r':
            t.right(360/n)
        else:
            t.left(360/n)
        t.forward(50)
```

**Graphics Displayed:**

```
def draw(t, n):
    t.forward(100)
    mystery(t, n, 'r')
    mystery(t, n, 'l')
```

```
t = Turtle()
draw(t, 3)
```

7. Write a **program** that reads in a text file, `infile.txt`, and replace each line with the word **Awesome** (that is, every line of the `infile.txt` should be **Awesome**), then prints out the total number of lines in the file.

8. Write the python code for the algorithms below:

```
(a) find(st)
    set index to (length of st) - 1
    set location to -1
    set found to false
    while not found
        if st[index] equals ','
            set location to index
            set found to True
        decrement index
    return location
```

```
(b) getBigger(ls)
    for each item in ls
        if current item is greater than first item in ls
            switch first item and current item in ls
```

9. Given the following input file `facebook-fans.dat`, write a program that reads in the input file and prints out name of the celebrity followed by the word **Amazing** for each celebrity with over 85 million fans, the word **Sweet** for celebrities with fans between 70 and 85 million and the word **Good** for all others.

**facebook-fans.dat** (Celebrity, Fans in Millions)

```
Shakira, 87
Rihanna, 86
Eminem, 84
Cristiano Ronaldo, 76
Vin Diesel, 70
Katy Perry, 65
Will Smith, 64
Justin Bieber, 64
Lady Gaga, 63
Linkin Park, 60
```

10. Given the following code:

```
def getArray():
    input = [
        "Stanley,Paul",
        "Simmons,Gene",
        "Singer,Eric",
        "Thayer,Tommy"
    ]
    return input
def main():
    array = getArray()
    last = ""
    for line in array:
        last = line.split(",")[1]
    print(last)
main()
```

And given the following CSV file called `kiss.txt`:

```
Stanley,Paul
Simmons,Gene
Singer,Eric
Thayer,Tommy
```

- (a) To make the program easier to update, rewrite the `getArray()` function so that it retrieves the input from the CSV file. The updated program should print the same output when run.

- (b) What is the output of the program?

**Useful String Methods:** (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

**Useful Turtle Methods:** (from <http://docs.python.org/3.0/library/turtle.html>)

Function	Meaning
<code>forward(d)</code>	Move turtle forward <code>d</code> steps
<code>backward(d)</code>	Move turtle backward <code>d</code> steps
<code>right(angle)</code>	Turn turtle <code>angle</code> degrees to the right
<code>left(angle)</code>	Turn turtle <code>angle</code> degrees to the left
<code>up()</code>	Pull the pen up no drawing when moving
<code>down()</code>	Pull the pen down drawing when moving

NAME:  
EMAIL:  
SIGNATURE:  
CIRCLE COURSE SECTION: MW 1-3 TTh 11-1 TTh 4-6

**Lehman College, CUNY**  
**CIS 166 Final Exam, Version 3, Spring 2014**

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
s = "history.biology.french.trigonometry.science."  
num = s.count(".")  
subjects = s[:-1].split(".")  
print("There are", num, "important subjects in school.")  
for item in subjects[:-1]:  
    print("Don't know much about", item)  
print("But I do know that I love computer " + subjects[4])
```

2. Define a Python function named `calculate_tax` which accepts one parameter, `income`, and returns the income tax. Income is taxed according to the following rule: the first \$50,000 is taxed at 10% and any remaining income is taxed at 20%. For example, `calculate_tax(40000)` should return  $40,000 \times 0.1 = 4,000$ , while `calculate_tax(100000)` should return  $50,000 \times 0.1 + 50,000 \times 0.2 = 15,000$ .

3. Complete the following program that is, write the functions `getInputs()`, `countSpaces()`, `minMax()`, and `printSummary()`:

```
def main():
    fname = getInputs()          #get the file name
    infile = open(fname, "r")    #open the file for reading
    resultList = list()         #initialize result list to empty list

    for line in infile:
        num = countSpaces(line) #return the number of spaces in line
        resultList.append(num)

    m,M = minMax(resultList)     #compute the minimum and maximum spaces per line
    printSummary(m,M)           #print the min and max spaces (including explanation)
```

4. Given the following function definitions:

```
def bar(n):  
    if n < 8:  
        return -1  
    else:  
        return n//2
```

```
def foo(l):  
    n = bar(l[3])  
    return 2*n
```

(a) What does `foo([1,2,3,4])` return?

(b) What does `foo([1024,512,256,128])` return?

5. Given the following code:

```
file = open("numbers.txt")
total = 0
for line in file.readlines():
    for strnum in line.split(","):
        num = int(strnum)
        if num % 2 == 0:
            print(num)
            total = total + num
print(total)
```

(a) What will the output be for this `numbers.txt`?

**numbers.txt:**

1,2,3,4,5,6

(b) What will the output be for this `numbers.txt`?

**numbers.txt:**

123456

6. Draw what would be displayed in the graphics window when the following program is executed. Remember to indicate the final position and direction of the turtle at the end of program. (The turtle always points to the right of the screen at the start of the program.)

```
from turtle import *

def mystery(t, n, d):
    for i in range(n):
        if d == 'r':
            t.right(360/n)
        else:
            t.left(360/n)
        t.forward(50)
```

**Graphics Displayed:**

```
def draw(t, n):
    t.backward(100)
    mystery(t, n, 'l')
    mystery(t, n, 'r')
```

```
t = Turtle()
draw(t, 3)
```

7. Write a **program** that reads in a text file, `infile.txt`, and prints out each line surrounded by `'--'`.

8. Write the python code for the algorithms below:

```
(a) find(st)
    set index to 0
    set location to -1
    set firstFound to false
    set notFound to true
    while notFound and index < length st
        if st[index] equals ',' and firstFound is false
            set firstFound to true
        otherwise, if st[index] equals ','
            set location to index
            set notFound to false
        increment index
    return location
```

```
(b) getBigger(ls)
    for each item in ls
        if current item is greater than last item in ls
            switch last item and current item in ls
```

9. Given the following input file `best-selling-albums.dat`, write a program that reads in the input file and prints out the name of the album followed by the word **Amazing** for each album whose sales were over 50 millions, the word **Sweet** for sales between 45 and 50 millions and the word **Good** for all others.

**best-selling-albums.dat** (Album Name, Copies Sold in Millions)

Thriller, 65  
The Dark Side of the Moon, 45  
Eagles Greatest Hits, 42  
Back in Black, 40  
Saturday Night Fever, 40  
Rumours, 40  
The Bodyguard, 40

10. Given the following code:

```
def getArray():
    input = [
        "Targaryen, Daenerys",
        "Baelish, Petyr",
        "Arryn, Lysa",
        "Clegane, Sandor",
        "Stark, Bran"
    ]
    return input
def main():
    array = getArray()
    last = ""
    for line in array:
        last = line.split(",")[1]
    print(last)
main()
```

And given the following CSV file labeled `stillalive.txt`:

```
Targaryen, Daenerys
Baelish, Petyr
Arryn, Lysa
Clegane, Sandor
Stark, Bran
```

- (a) To make the program easier to update, rewrite the `getArray()` function so that it retrieves the input from the CSV file. The updated program should print the same output when run.

- (b) What is the output of the program?

**Useful String Methods:** (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

**Useful Turtle Methods:** (from <http://docs.python.org/3.0/library/turtle.html>)

Function	Meaning
<code>forward(d)</code>	Move turtle forward <code>d</code> steps
<code>backward(d)</code>	Move turtle backward <code>d</code> steps
<code>right(angle)</code>	Turn turtle <code>angle</code> degrees to the right
<code>left(angle)</code>	Turn turtle <code>angle</code> degrees to the left
<code>up()</code>	Pull the pen up no drawing when moving
<code>down()</code>	Pull the pen down drawing when moving

NAME:  
EMAIL:  
SIGNATURE:  
CIRCLE COURSE SECTION: MW 1-3 TTh 11-1 TTh 4-6

**Lehman College, CUNY**  
**CIS 166 Final Exam, Version 4, Spring 2014**

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
s = "omelettesporridgescerealspancakes"  
num = s.count("s")  
breakfast = s[:-1].split("s")  
print("You have a choice of", num, "options:")  
for item in breakfast:  
    print(item.capitalize())  
print("\nBut I need " + breakfast[0][1] + breakfast[1][1] + breakfast[2][2:4] + "!!!")
```

2. Define a Python function named `calculate_tax` which accepts one parameter, `income`, and returns the income tax. Income is taxed according to the following rule: the first \$500,000 is taxed at 50% and any remaining income is taxed at 75%. For example, `calculate_tax(400000)` should return  $400,000 \times 0.5 = 200,000$ , while `calculate_tax(600000)` should return  $500,000 \times 0.5 + 100,000 \times 0.75 = 325,000$ .

3. Complete the following program that is, write the functions `getInputs()`, `countSpaces()`, `calculate()`, and `printSummary()`:

```
def main():
    fname = getInputs()          #get the file name
    infile = open(fname, "r")    #open the file for reading
    resultList = list()          #initialize result list to empty list

    for line in infile:
        num = countSpaces(line) #return the number of spaces in line
        resultList.append(num)

    n = calculate(resultList)    #compute number of lines with more than 5 spaces
    printSummary(n)              #print the number of long lines (including explanation)
```

4. Given the following function definitions:

```
def bar(n):  
    if n >= 8:  
        return 8  
    else:  
        return n*2  
  
def foo(l):  
    n = bar(l[1])  
    return n//2
```

(a) What does `foo([1,2,3,4])` return?

(b) What does `foo([1024,512,256,128])` return?

5. Given the following code:

```
file = open("numbers.txt")
total = 0
for line in file.readlines():
    for strnum in line.split(","):
        num = int(strnum)
        if num % 2 == 0:
            print(num)
            total = total + num
print(total)
```

(a) What will the output be for this `numbers.txt`?

**numbers.txt:**

5,6,7,8,9

(b) What will the output be for this `numbers.txt`?

**numbers.txt:**

5  
6  
7  
8  
9

6. Draw what would be displayed in the graphics window when the following program is executed. Remember to indicate the final position and direction of the turtle at the end of program. (The turtle always points to the right of the screen at the start of the program.)

```
from turtle import *

def mystery(t, n, d):
    for i in range(n):
        if d == 'r':
            t.right(360/n)
        else:
            t.left(360/n)
        t.forward(50)
```

**Graphics Displayed:**

```
def draw(t, n):
    t.backward(100)
    mystery(t, n, 'l')
    mystery(t, n, 'r')
```

```
t = Turtle()
draw(t, 4)
```

7. Write a **program** that reads in a text file, `infile.txt`, and prints out each line uppercase except for first character on each line. For example, "Hello World" should be printed out as "hELLO WORLD".

8. Write the python code for the algorithms below:

(a) find(st)

```
set index to (length of st) - 1
set location to -1
set firstFound to false
set notFound to true
while notFound and index > -1
    if st[index] equals ',' and firstFound is false
        set firstFound to true
    otherwise, if st[index] equals ','
        set location to index
        set notFound to false
    decrement index
return location
```

(b) getSmaller(ls)

```
for each item in ls
    if current item is smaller than last item in ls
        switch last item and current item in ls
```

9. Given the following input file `pitchers-era.dat`, write a program that reads in the input file and prints out the name of the pitcher followed by the word **Awesome** for each ERA between 1.00 and 1.99, the word **Great** for each ERA between 2.00 and 2.99 and the word **Good Job** for all others.

`pitchers-era.dat` (Name, ERA)

```
Ed Walsh, 1.82
Mariano Rivera, 2.21
Babe Ruth, 2.28
Sandy Koufax, 2.76
Juan Marichal, 2.89
Pedro Martinez, 2.93
Roger Clemens, 3.12
Greg Maddux, 3.16
```

10. Given the following code:

```
def getArray():
    input = [
        "Baggins,Frodo",
        "Gamgee, Samwise",
        "Greenleaf, Legolas",
        "Baggins, Bilbo"
    ]
    return input
def main():
    array = getArray()
    last = ""
    for line in array:
        last = line.split(",")[1]
    print(last)
main()
```

And given the following CSV file labeled `lotr.txt`:

```
Baggins,Frodo
Gamgee, Samwise
Greenleaf, Legolas
Baggins, Bilbo
```

- (a) To make the program easier to update, rewrite the `getArray()` function so that it retrieves the input from the CSV file. The updated program should print the same output when run.

- (b) What is the output of the program?

**Useful String Methods:** (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

**Useful Turtle Methods:** (from <http://docs.python.org/3.0/library/turtle.html>)

Function	Meaning
<code>forward(d)</code>	Move turtle forward <code>d</code> steps
<code>backward(d)</code>	Move turtle backward <code>d</code> steps
<code>right(angle)</code>	Turn turtle <code>angle</code> degrees to the right
<code>left(angle)</code>	Turn turtle <code>angle</code> degrees to the left
<code>up()</code>	Pull the pen up no drawing when moving
<code>down()</code>	Pull the pen down drawing when moving