Name:
Email:
Signature:
Circle course section: MW 11-1 MW 1-3 TTh 6-8

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

Lehman College, CUNY
CIS 166 Final Exam, Version 1, Fall 2014

1. What will the following code print:

```
a = ",Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec,"
b = "Apr 15, 2014"
c = b.split()
print(c)
d = a.split(",")
print(d[1:12])
e = (a.find(c[0]) - 1) / 3
print(e)
f = c[1][:-1]
print(str(int(e)) + "/" + f + "/" + c[2])
```

2. Write a program to print the fine for speeding. The program must read the speed from user input, then compute and print the fine. The fine is \$12 for each mph over 65 and less than or equal to 70, and \$15 for each additional mph over 70.

   For example, if the speed is 68 mph, then the fine would be \$36 = \$12 x 3. If the speed is 72 mph, then the fine would be \$90 = \$12 x 5 + \$15 x 2.

3. Complete the following program, which reads in a file that has multiple grades, each separated by a comma, and prints out the computed average. That is, write the functions `getGrades()` and `calculateAverage()`:

```
def main():
    grades = getGrades()   #get the file name containing the grades
                           #and return the contents of the file
    avg = calculateAverage(grades) #separate the grades into numbers and compute
                           #the average
    print("The calculated average is:", avg)

main()
```

4. Given the following function definitions:

```
def help(g):
    s = 1
    for h in g:
        s = s + h
        print(s)
    return s

def abc(d):
    e = len(d)
    print("e is ", e)
    if e >= 2:
        f = help(d[0:3])
    elif 2 > e >= 1:
        f = help(d[0:1])
    else:
        f = 5
    return f
```

(a) What does `abc([0,1,2,3])` return?

Write output for partial credit:

(b) What does `abc([49])` return?

Write output for partial credit:

5. Given the following code:

```
def main():
    file = open("poetry.txt", 'r')
    count = 0
    for line in file:
        line2 = line[:-1] + "?"
        if count % 2 == 0:
            print(line2)
        else:
            print(len(line[:-1]))
        count = count + 1

main()
```

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

poetry.txt:

```
What a
nice
day.
It is.
```

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

poetry.txt:

```
No rain
but
cloudy.
```

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):

    for i in range(n):
        t.forward(50)
        if i % 2 == 0:
            t.right(90)
        else:
           t.left(90)

def draw(t, n):
    mystery(t, n)

t = Turtle()
draw(t, 5)
```

Graphics Displayed:

7. Write a program that reads in a file called **infile.txt**. It should count and print out three things: the number of lines in the file, the number of times that the lower-case letter *e* appears in the file, and the average number of times that the lower-case letter *e* appears per line.

8. Write the Python code for the algorithms below:

   (a) `total(ls)`

```
total(ls)
    Set total to 0
    for each item in the list ls
        Add item to total
    print total
```

   (b) `search(ls, key, first, last)`

```
search(ls, key, first, last)
    while first is less than last
        Set mid to first + last / 2
        if ls[mid] is less than key
            Set first to mid + 1
        else
            Set last to mid
    if last equals first and ls[first] equals key
        return first
    else
        return -1
```

9. Given the following input file `mathproblems.dat`, write a `program` that reads in the input file, executes the `operation` in the middle on the numbers to the left and right. Print the `result` of each line.

   mathproblems.dat (Number, operand, Number)

   ```
   4, +, 2
   5, -, 3
   20, *, 2
   10, /, 5
   ```

10. Write a program which does the following:
    (a) Takes in a CSV file, where each line of the file contains:
    `<Last Name>,<First Name>,<City>,<State>,<Zip>`
    (b) Asks for the user to input a particular State to search by
    (c) Searches for all entries in the CSV which contains the given state
    (d) Prints to screen all names belonging to that state

    For example, given a CSV file labeled `employees.txt`:

```
Oppenheimer,Robert,Bronx,NY,10467
Fermi,Enrico,Manhattan,NY,10001
Feynman,Richard,Brooklyn,NY,12255
Teller,Edward,Knoxville,TN,12345
Frisch,Otto,Phoenix,AZ,54321
```

    If the user enters the state ”NY”, the resulting output of the program would be:

```
The following people live in NY:
Robert Oppenheimer
Enrico Fermi
Richard Feynman
```

Useful String Methods: (from p 140 of textbook)

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

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

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

Name:
Email:
Signature:
Circle course section: MW 11-1 MW 1-3 TTh 6-8

Lehman College, CUNY
CIS 166 Final Exam, Version 2, Fall 2014

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
a = ",Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec,"
b = "Mar 15, 2014"
c = b.split()
print(c)
d = a.split(",")
print(d[1:12])
e = a.find(c[0]) / 3
print(e)
f = c[1][:-1]
print(str(int(e)) + "/" + f + "/" + c[2])
```

2. Write a program to print the fine for speeding. The program must read the speed from user input, then compute and print the fine. The fine is \$10 for each mph over 55 and less than or equal to 65, and \$15 for each additional mph over 65.

   For example, if the speed is 58 mph, then the fine would be \$30 = \$10 x 3. If the speed is 67 mph, then the fine would be \$130 = \$10 x 10 + \$15 x 2.

3. Complete the following program, which reads in a file that has multiple grades, each separated by a semi-colon, and prints out the computed average. That is, write the functions `retrieveGrades()` and `computeAverage()`:

```
def main():
    grades = retrieveGrades()  #get the file name containing the grades
                                #and return the contents of the file
    avg = computeAverage(grades) #separate the grades into numbers and compute
                               #the average
    print("The calculated average is:", avg)

main()
```

4. Given the following function definitions:

```
def help(g):
    s = 0
    for h in g:
        s = s + 2
        print(s)
    return s

def abc(d):
    e = len(d) - 1
    print("e is", e)
    if e >= 3:
        f = help(d[0:2])
    elif 2 >= e >= 1:
        f = help(d[0:1])
    else:
        f = 10
    return f
```

(a) What does `abc([7,8,9])` return?

Write output for partial credit:

(b) What does `abc([77])` return?

Write output for partial credit:

5. Given the following code:

```
def main():
    file = open("song.txt", 'r')
    count = 0
    for line in file:
        line2 = line[:-1] + str(count);
        if count % 2 == 0:
            print(line2)
        else:
            print(line[:-1])
        count = count + 1

main()
```

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

song.txt:

```
Hi ho
it's off
to program
I go.
```

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

song.txt:

```
Sitting on
the
dock.
```

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):

    for i in range(n):
        t.forward(50)
        if i % 2 == 0:
            t.left(90)
        else:
           t.right(90)

def draw(t, n):
    mystery(t, n)

t = Turtle()
draw(t, 5)
```

Graphics Displayed:

7. Write a program that reads in a file, **infile.txt**. For each line in the file it should print out "- * -" to an output file, **outfile.txt**. Finally, it should print the total number of lines in the input file to the screen.

8. Write the Python code for the algorithms below:

(a) count(ls)

```
count(ls)
    Set count to 0
    for each item in the list ls
        If item is positive
            increment count
    print count
```

(b) search(ls, key, first, last)

```
search(ls, key, first, last)
    while first is less than last
        Set mid to first + last / 2
        if ls[mid] equals key
            return mid
        else if ls[mid] < key
            first = mid + 1
        else
            last = mid -1
    return -1
```

9. Write a program that reads in a file, **infile.txt**. For each line in the file it should print out the number of lowercase e in that line. At the end, it should print out the average number of lowercase e per line.

10. Write a program which does the following:

    (a) Takes in a CSV file, where each line of the file contains:
    `<Last Name>,<First Name>,<Grade>`

    (b) Asks for the user to input a particular grade to search by

    (c) Searches for all entries in the CSV which contains a grade greater than the given grade

    (d) Prints to screen all names that match the criteria

    For example, given a CSV file labeled `students.txt`:

```
Oppenheimer,Robert,80
Fermi,Enrico,90
Feynman,Richard,70
Teller,Edward,60
Frisch,Otto,50
```

    If the user enters the grade ”75”, the resulting output of the program would be:

```
The following people have a better grade than 75:
Robert Oppenheimer
Enrico Fermi
```

Useful String Methods: (from p 140 of textbook)

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

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

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

Name:
Email:
Signature:
Circle course section: MW 11-1 MW 1-3 TTh 6-8

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

Lehman College, CUNY
CIS 166 Final Exam, Version 3, Fall 2014

1. What will the following code print:

```
a = ",Dec,Nov,Oct,Sep,Aug,Jul,Jun,May,Apr,Mar,Feb,Jan,"
b = "Nov 15, 2014"
c = b.split()
print(c)
d = a.split(",")
print(d[1:12])
e = (a.find(c[0]) - 1) / 4 + 1
print(e)
f = c[1][:-1]
print(str(int(e)) + "/" + f + "/" + c[2])
```

2. Write a program to print the fine for speeding. The program must read the speed from user input, then compute and print the fine. The fine is \$15 for each mph over 60 and less than or equal to 70, and \$20 for each additional mph over 70.

   For example, if the speed is 63 mph, then the fine would be \$45 = \$15 x 3. If the speed is 72 mph, then the fine would be \$190 = \$15 x 10 + \$20 x 2.

3. Complete the following program, which reads in a file that has multiple grades, each separated by a colon, and prints out the computed average. That is, write the functions `extractGrades()` and `processAverage()`:

```
def main():
    grades = extractGrades() #get the file name containing the grades
                             #and return the contents of the file
    avg = processAverage(grades) #separate the grades into numbers and compute
                             #the average
    print("The calculated average is:", avg)

main()
```

4. Given the following function definitions:

```
def help(g):
    s = 1
    for h in g:
        s = s + 1
        print(s)
    return s

def abc(d):
    e = len(d)
    print("e is ", e)
    if 5 > e > 2:
        f = help(d[0:3])
    elif e > 5:
        f = help(d[2:5])
    else:
        f = 8
    return f
```

(a) What does `abc([10,20,30,40,50,60])` return?

Write output for partial credit:

(b) What does `abc([5,6,7])` return?

Write output for partial credit:

5. Given the following code:

```
def main():
    file = open("story.txt", 'r')
    count = 0
    for line in file:
        line2 = "!" + line[:-1]
        if count == 2:
            print(line2)
        else:
            print(line.count("a"))
        count = count + 2

main()
```

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

story.txt:

```
Once
upon a
time.
```

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

story.txt:

```
Here
is
a
story...
```

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):

    for i in range(n):
        t.backward(50)
        if i % 2 == 0:
            t.left(90)
        else:
           t.right(90)

def draw(t, n):
    mystery(t, n)

t = Turtle()
draw(t, 5)
```

Graphics Displayed:

7. Write a program that reads in a file called **infile.txt**. For each line in the file it should print out the line number and the number of times the lower-case letter *a* appears in that line.

8. Write the Python code for the algorithms below:

(a) `count(ls)`

```
count(ls)
        Set count to 0
        for each item in the list ls
            If item is negative
                increment count
        print count
```

(b) `search(ls, key, first, last)`

```
search(ls, key, first, last)
        while first is less than last
            Set mid to first + last / 2
            if ls[mid] equals key
                return mid
            else if ls[mid] < key
                first = mid + 1
            else
                last = mid -1
        return -1
```

9. Given the following input file `mathproblems.dat`, write a `program` that reads in the input file, executes the `operation` in the middle on the numbers to the left and right. Print the `result` of each line.

   mathproblems.dat (Number, operand, Number)

   ```
   10, +, 2
   20, -, 5
   11, *, 2
   50, /, 5
   ```

10. Write a program which does the following:

    (a) Takes in a CSV file, where each line of the file contains: `<Last Name>,<First Name>,<Pet Preference>`

    (b) Asks for the user to input a pet to search by

    (c) Searches for all entries in the CSV which contains the given pet

    (d) Prints to screen all names that have that pet preference

    For example, given a CSV file labeled `petowners.txt`:

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

    If the user enters ”cat”, the resulting output of the program would be:

```
The following people like cats:
Robert Oppenheimer
Enrico Fermi
Richard Feynman
```

Useful String Methods: (from p 140 of textbook)

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

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

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