

Answer Key: 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])
```

Answer Key:

```
['Apr', '15,', '2014']
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov']
4.0
4/15/2014
```

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 = \12×3 . If the speed is 72 mph, then the fine would be $\$90 = \$12 \times 5 + \$15 \times 2$.

Answer Key:

```
def answer1():
    speed = eval(input("Enter the speed in mph:"))
    if (speed < 65):
        print("No fine")
    else:
        fine = (speed - 65) * 12
        if speed > 70:
            fine = fine + (speed - 70) * 3
        print("The fine is", fine)

def answer2():
    speed = eval(input("Enter the speed in mph:"))
    if speed <= 65:
        print("No fine")
    elif speed <= 70:
        print("The fine is", (speed - 65) * 12)
    else:
        print("The fine is", 60 + (speed - 70) * 15)
```

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()
```

Answer Key:

```
def getGrades():
    contents = open(input("Enter the file with the grades"), "r").read()
    grades = contents.split(",")
    for i in range(len(grades)):
        grades[i] = int(grades[i])
    return grades
def calculateAverage(grades):
    sum = 0
    for i in grades:
        sum = sum + i
    return sum/len(grades)
```

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?

Answer Key: 4

Write output for partial credit:

Answer Key:

```
e is 4
1
2
4
```

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

Answer Key: 50

Write output for partial credit:

Answer Key:

```
e is 1
50
```

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.
```

Answer Key:

```
What a?
4
day.?
6
```

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

poetry.txt:

No rain
but
cloudy.

Answer Key:

```
No rain?  
3  
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):

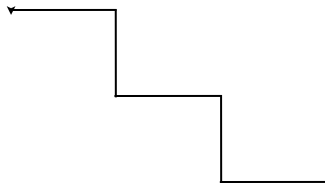
    for i in range(n):
        t.forward(50)
        if i % 2 == 0:
            t.right(90)
        else:
            t.left(90)

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

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

Graphics Displayed:

Answer Key:



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.

Answer Key:

```
#some comments
```

```

def main():
    infile = open('infile.txt')
    lines = 0
    es = 0
    for line in infile:
        lines += 1
        es += line.count('e')

    print("There were", lines, "lines and", es, "e's, for an average of", es/lines, "e's per li

    infile.close()

```

8. Write the Python code for the algorithms below:

(a) total(ls)

```

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

```

Answer Key:

```

def total(ls):
    total = 0
    for item in ls:
        total = total + ls
    print(total)

```

(b) 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

```

Answer Key:

```

def search(ls, key, first, last):
    while first < last:
        mid = (first + last) / 2
        if ls[mid] < key
            first = mid + 1
        else

```

```

        last = mid
    if (last == first) and (ls[first] == 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

```

Answer Key:

```

def main():
    infile = open("mathproblems.dat", "r")
    for l in infile.readlines():
        words = l.split(",")
        num1 = int(words[0])
        num2 = int(words[2])
        operand = words[1]

        if operand == "-":
            print(num1 - num2)
        elif operand == "+":
            print(num1 + num2)
        elif operand == "*":
            print(num1 * num2)
        elif operand == "/":
            print(num1/num2)
    infile.close()

```

10. Write a program which does the following:

- Takes in a CSV file, where each line of the file contains:
<Last Name>,<First Name>,<City>,<State>,<Zip>
- Asks for the user to input a particular State to search by
- Searches for all entries in the CSV which contains the given state
- 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

Answer Key:

```
def main():
    file = open(input("Enter the csv file: "), "r")
    state = input("Enter the state to search by: ")
    print("The following people live in", state)
    for line in file:
        record = line.split(",")
        if record[3] == state:
            print(record[1] + " " + record[0])
main()
```

Answer Key: CIS 166 Final Exam, Version 2, Fall 2014

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])
```

Answer Key:

```
['Mar', '15,', '2014']
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov']
3.0
3/15/2014
```

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 = \10×3 . If the speed is 67 mph, then the fine would be $\$130 = \$10 \times 10 + \$15 \times 2$.

Answer Key:

```
def answer1():
    speed = eval(input("Enter the speed in mph:"))
    if (speed<55):
        print("No fine")
    else:
        fine = (speed - 55) * 10
        if speed > 65:
            fine = fine + (speed - 65) * 5
        print("The fine is", fine)

def answer2():
    speed = eval(input("Enter the speed in mph:"))
    if speed <= 55:
        print("No fine")
    elif speed <= 65:
        print("The fine is", (speed - 55) * 10)
    else:
        print("The fine is", 100 + (speed - 65) * 15)
```


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()
```

Answer Key:

```
def retrieveGrades():
    contents = open(input("Enter the file with the grades"), "r").read()
    grades = contents.split(";")
    for i in range(len(grades)):
        grades[i] = int(grades[i])
    return grades
def computeAverage(grades):
    sum = 0
    for i in grades:
        sum = sum + i
    return sum/len(grades)
```

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?

Answer Key: 2

Write output for partial credit:

Answer Key:

```
e is 2
2
```

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

Answer Key: 10 Write output for partial credit:

Answer Key:

```
e is 2
```

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.
```

Answer Key:

```
Hi ho0
it's off
to program2
I go.
```

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

```
song.txt:
Sitting on
the
dock.
```

Answer Key:

```
Sitting on0
the
dock.2
```

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):

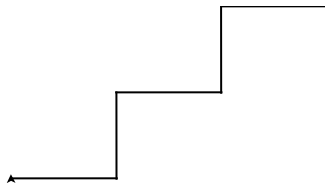
    for i in range(n):
        t.forward(50)
        if i % 2 == 0:
            t.left(90)
        else:
            t.right(90)

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

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

Graphics Displayed:

Answer Key:



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.

Answer Key:

```
#some comments
```

```

def main():
    infile = open('infile.txt')
    outfile = open('outfile.txt', 'w')
    lines = 0
    for line in infile:
        lines += 1
        print('- * -', file = outfile)
    print("Total lines:", lines)
    outfile.close()
    infile.close()

```

8. Write the Python code for the algorithms below:

(a) count(ls)

```

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

```

Answer Key:

```

def count(ls):
    count = 0
    for item in ls:
        if item > 0:
            count = count + 1
    print(count)

```

(b) 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

```

Answer Key:

```

def search(ls, key, first, last):
    while first < last:
        mid = (first + last) / 2
        if ls[mid] == key:
            return mid

```

```

        elif 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.

Answer Key:

```

#some comments

def main():
    infile = open('infile.txt')
    lines = 0
    totalEs = 0
    for line in infile:
        lines += 1
        es = line.count('e')
        totalEs += es
        print(es)

    print("There were an average of", es/lines, "e's per line.")

    infile.close()

```

10. Write a program which does the following:

- Takes in a CSV file, where each line of the file contains:
<Last Name>,<First Name>,<Grade>
- Asks for the user to input a particular grade to search by
- Searches for all entries in the CSV which contains a grade greater than the given grade
- 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

```

Answer Key:

```
def main():
    file = open(input("Enter the csv file: "), "r")
    grade = input("Enter the grade to search by: ")
    print("The following people have a grade better than", grade)
    for line in file:
        record = line.split(",")
        if record[3] > grade:
            print(record[1] + " " + record[0])
main()
```

Answer Key: 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])
```

Answer Key:

```
['Nov', '15,', '2014']
['Dec', 'Nov', 'Oct', 'Sep', 'Aug', 'Jul', 'Jun', 'May', 'Apr', 'Mar', 'Feb']
2.0
2/15/2014
```

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 = \15×3 . If the speed is 72 mph, then the fine would be $\$190 = \$15 \times 10 + \$20 \times 2$.

Answer Key:

```
def answer1():
    speed = eval(input("Enter the speed in mph:"))
    if (speed < 60):
        print("No fine")
    else:
        fine = (speed - 60) * 15
        if speed > 70:
            fine = fine + (speed - 70) * 5
        print("The fine is", fine)

def answer2():
    speed = eval(input("Enter the speed in mph:"))
    if speed <= 60:
        print("No fine")
    elif speed <= 70:
        print("The fine is", (speed - 60) * 15)
    else:
        print("The fine is", 150 + (speed - 70) * 20)
```

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()
```

Answer Key:

```
def extractGrades():
    contents = open(input("Enter the file with the grades"), "r").read()
    grades = contents.split(":")
    for i in range(len(grades)):
        grades[i] = int(grades[i])
    return grades
def processAverage(grades):
    sum = 0
    for i in grades:
        sum = sum + i
    return sum/len(grades)
```

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?

Answer Key: 4

Write output for partial credit:

Answer Key:

e is 6
2
3
4

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

Answer Key: 4

Write output for partial credit:

Answer Key:

e is 3
2
3
4

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.

Answer Key:

0
1
!time.

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

`story.txt:`

Here
is
a
story...

Answer Key:

0
0
!a
0

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:

Answer Key:



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.

Answer Key:

```
#some comments

def main():
    infile = open('infile.txt')
    line = 0
```

```

for line in infile:
    line += 1
    as = line.count('a')
    print(line, as)

infile.close()

```

8. Write the Python code for the algorithms below:

(a) count(ls)

```

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

```

Answer Key:

```

def count(ls):
    count = 0
    for item in ls:
        if item < 0:
            count = count + 1
    print count

```

(b) 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

```

Answer Key:

```

def search(ls, key, first, last):
    while first < last:
        mid = (first + last) / 2
        if ls[mid] == key:
            return mid
        elif 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
```

Answer Key:

```
def main():
    infile = open("mathproblems.dat", "r")
    for l in infile:
        words = l.split(",")
        num1 = int(l[0])
        num2 = int(l[2])
        operand = l[1]

        if operand == "-":
            print(num1 - num2)
        elif operand == "+":
            print(num1 + num2)
        elif operand == "*":
            print(num1 * num2)
        else operand == "/":
            print(num1/num2)
    infile.close()
```

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
```

Answer Key:

```
def main():
    file = open(input("Enter the csv file: "), "r")
    pet = input("Enter the state to search by: ")
    print("The following people like", pet + "s")
    for line in file:
        record = line.split(",")
        if record[3] == pet:
            print(record[1] + " " + record[0])
main()
```