CS 115 Sample Final Exam #2

1. Consider the following code:

```
for i in range(10, 0, -1):
    print(str(3 - i * 4) + str(i * 2) + "<", end="")
```

What does it output?

2. Parameter Mystery

At the bottom of the page, write the output produced by the following program.

```
def main():
    a = "felt"
    b = "saw"
    c = "drew"
    saw = "sue"
    drew = "b"

    mystery(a, b, c)
    mystery(b, a, saw)
    mystery(drew, c, saw)
    mystery("a", saw, drew)
    mystery(a, a, "drew")

def mystery(b, a, c):
    print(c, a, "the", b)

main()
```

3. Return Mystery

At the bottom of the page, write the output produced by the following program.

```
def main():
    a = 42
    b = 7
    c = 3
    print(function_3())
    a = function_1() + function_2()
    f = function_2()
    c = function_3()
    print(a, b, c, f)

def function_1():
    function_3()
    print("B")
    return 1

def function_2():
    e = function_3()
    print("C")
    d= function_1()
    return d + e

def function_3():
    print("A")
    return 2

main()
```

4. List Mystery 1

Consider the following function:

```
def list_mystery(a):
    a[1] = 2
    a[3] = 82
    a[4] = a[2]
    a[a[1]] = 20
    num = a[2]
```

Indicate in the right-hand column what values would be stored in the list after the subroutine `list_mystery` executes if the passed in list contains the elements in the left column.

Original Contents of List	Final Contents of List
`a1 = [7, 1, 5, 4, 3, 2, 1]` `list_mystery(a1)`	_____
`a2 = [4, 3, 6, 14, 55, 12]` `list_mystery(a2)`	_____
`a3 = [7, 4, 8, 6, 2, 1, 1]` `list_mystery(a3)`	_____
`a4 = [10, 2, 5, 10, 10]` `list_mystery(a4)`	_____
`a5 = [2, 4, -1, 6, -2, 8]` `list_mystery(a5)`	_____

5. List Mystery 2

Consider the following function:

```
def list_mystery(a):
    for i in range(len(a) - 2, 0, -1):
        if a[i + 1] <= a[i - 1]:
            a[i] += 1
```

Indicate in the right-hand column what values would be stored in the list after the function `list_mystery` executes if the integer list in the left-hand column is passed as a parameter to it.

Original Contents of List	Final Contents of List
`a1 = [42]` `list_mystery(a1)`	_____
`a2 = [1, 8, 3, 6]` `list_mystery(a2)`	_____
`a3 = [5, 5, 5, 5, 5]` `list_mystery(a3)`	_____
`a4 = [10, 7, 9, 6, 8, 5]` `list_mystery(a4)`	_____
`a5 = [1, 0, 1, 0, 0, 1, 0]` `list_mystery(a5)`	_____

6. While Loop Mystery

For each call of the function below, write the value that is returned:

```
def mystery(i, j):
    k = 0
    while i > j:
        i = i - j
        k = k + (i - 1)
    return k
```

Function Call	**Value Returned**
`mystery(2, 9)`	_____
`mystery(5, 1)`	_____
`mystery(38, 5)`	_____
`mystery(5, 5)`	_____
`mystery(40, 10)`	_____

7. Programming

Write a function named `three_heads` that repeatedly flips a coin until three heads *in a row* are seen. You should use `randint` to give an equal chance to a head or a tail appearing. Each time the coin is flipped, what is seen is displayed (`H` for heads, `T` for tails). When 3 heads in a row are flipped a congratulatory message is printed. Here are possible outputs of two calls to `three_heads`:

```
T T T H T H H H
Three heads in a row!
```

```
T H T H T T T T T H H T H H H
Three heads in a row!
```

8. Programming

Write a function named `print_multiples` that accepts an integer `n` and an integer `m` as parameters and that prints a complete line of output reporting the first `m` multiples of `n`. For example, the following calls:

```
print_multiples(3, 5)
print_multiples(7, 3)
```

should produce this output:

```
The first 5 multiples of 3 are, 3, 6, 9, 12, 15
The first 3 multiples of 7 are, 7, 14, 21
```

You must exactly reproduce this format. You may assume that the number of multiples you will be asked to generate is greater than or equal to 1.

9. Programming

Write a function named `is_sorted` that accepts a list of real numbers as a parameter and returns `True` if the list is in sorted (nondecreasing) order and `False` otherwise. For example, if lists named `list1` and `list2` store `[16.1, 12.3, 22.2, 14.4]` and `[1.5, 4.3, 7.0, 19.5, 25.1, 46.2]` respectively, the calls `is_sorted(list1)` and `is_sorted(list2)` should return `False` and `True` respectively. Assume the list has at least one element. A one-element list is considered sorted.

10. Programming

Write a function called `report_blank_lines` that takes a string representing a file name as a parameter and that prints out the line numbers of any blank lines and the total number of blank lines in the file. For example, given the following input file:

```
Remember that a file
can have blank lines
like the one below:

A blank line:

is read as a String
of length 1

by readlines
```

Your function should print the following output:

```
line 4 is blank
line 6 is blank
line 9 is blank
total blank lines = 3
```

Notice that each blank line produces a line of output and that there is a final line of output reporting the total number of blank lines. Also notice that lines are numbered starting with 1 (first line is line 1, second line is line 2, and so on). You are to exactly reproduce the format of this output.

Solutions

1. `for` Loop Mystery

```
-3720<-3318<-2916<-2514<-2112<-1710<-138<-96<-54<-12<
```

2. Parameter Mystery

```
drew saw the felt
sue felt the saw
sue drew the b
b sue the a
drew felt the felt
```

3. Return Mystery

```
A
2
A
B
A
C
A
B
A
C
A
B
A
4 7 2 3
```

4. List Mystery 1

```
[7, 2, 20, 82, 5, 2, 1]
[4, 2, 20, 82, 6, 12]
[7, 2, 20, 82, 8, 1, 1]
[10, 2, 20, 82, 5]
[2, 2, 20, 82, -1, 8]
```

5. List Mystery 2

Call	Final Contents of List
`a1 = [42]` `list_mystery(a1)`	`[42]`
`a2 = [1, 8, 3, 6]` `list_mystery(a2)`	`[1, 8, 4, 6]`
`a3 = [5, 5, 5, 5, 5]` `list_mystery(a3)`	`[5, 6, 5, 6, 5]`
`a4 = [10, 7, 9, 6, 8, 5]` `list_mystery(a4)`	`[10, 8, 10, 7, 9, 5]`
`a5 = [1, 0, 1, 0, 0, 1, 0]` `list_mystery(a5)`	`[1, 1, 1, 1, 0, 2, 0]`

6. While Loop Mystery

Function Call	Value Returned
mystery(2, 9)	0
mystery(5, 1)	6
mystery(38, 5)	119
mystery(5, 5)	0
mystery(40, 10)	57

7. Programming

```
def three_heads():
        num_heads = 0

        while num_heads < 3:
            flip = randint(0, 2)
            if flip == 0:
                num_heads = num_heads + 1
                print("H ")
            else:
                num_heads = 0
                print("T ")
        print()
        print("Three heads in a row!")
```

8. Programming

```
def print_multiples(n, times):
     print("The first", times, "multiples of", n, "are")
     for i in range(1, times + 1):
         print(", " + str(i * n))
     print()
```

9. Programming

```
def is_sorted(list):
    for i in range(0, len(list) - 1):
       if list[i] >= list[i + 1]:
           return False
    return True
```

10. Programming

```
def report_blank_lines(file_name):
    file = open(file_name);
    lines = file.readlines()
    count = 0
    for i in range(len(lines)):
        text = lines[i]
        if len(text) == 1:
            print("line", i, "is blank")
            count += 1
    print("total blank lines =", count)
```