Midterm #2  Review

Topics Covered

All Topics form Midterm #1

While Loops
  Conditions, initialization, avoiding infinite loops
  Sentinels
  Input validation loops

Accumulator pattern
  Running totals
  Self-referential variable update (e.g., total = total + 1)
  Augmented assignments  (e.g., total += 1)

For Loops
  Iterating over a loop (for x in [1, 2, 3, 4, 5])
  The range function (range(lb, ub, step))
  Negative step
  Using the target variable
  Nested loops

Functions
  Passing arguments
  Calling functions
  Scope
  Local variables
  Global variables
  Updating global variables in functions
  Returning results
  Returning multiple results
  Defining and using modules
Consider each program below. If running the program will produce a runtime error, write down what the error is. If the program is correct, write the output from executing the program.

(1)
```python
x = 1
while x < 10 :
    print("x =", x)
    x += 2
```

(2)
```python
x = 1
y = 11
while x <= y :
    print("x =", x, "y=" , y)
    x += 1
    y -= 1
```

(3)
```python
x = 1
stop = False
while not stop :
    print("x =", x)
    x += 7
    if x > 20 :
        stop = True
```

(4)
```python
x = 1
while x < 10:
    print("x=",x)```
(5)
for x in [5, 2, 7, 4, 1] :
    print("x =", x)

(6)
for x in range(5) :
    print("x =", x)

(7)
cnt = 0
for x in range(7) :
    cnt *= x
print("cnt = ", cnt)

(8)
for x in range(10, 20, -1) :
    print("x = ", x)

(9)
cnt = 0
for x in range(7, 1, -1) :
    for y in range(2) :
        cnt += 1
print("cnt= ", cnt)

(10)
for x in range(4) :
    for y in range(2) :
        print("x="+x,"y="+y)
(11)
for x in range(4):
  cnt = 0
  for y in range(1,2):
    cnt += 1
  print("cnt =", cnt)

(12)
for x in range(1,3):
  for y in range(x):
    print("x=", x, "y=", y)

(13)
def update_x(x, amt):
  x = x + amt
  return x

x = 1
print(update_x(x, 3))

(14)
def show_sum(num1, num2):
  num3 = num1 + num2
  print(num3)

show_sum(4, 5)
(15)
def show_sum(num1, num2):
    num3 = num1 + num2
    return num3

print(show_sum(5, 6))

(16)
def show_sum(num1, num2):
    num3 = num1 + num2

    show_sum(5, 4)
    print(num3)

(17)
def addup(a, b, c):
    result = a + b + c
    return result

print(addup(4, 5, 6.7))
print(addup("a", "b", "c"))

(18)
def addup(a, b):
    num = a+b
    print(num)

num = 5
addup(num, 8)
print(num)
def order(x, y):
    if x < y:
        return x, y
    else:
        return y, x

print(order("ABC", "ABCDEF"))
print(order(1, .005))

def check(x):
    if x >= 'a' and x <= 'm':
        return True
    else:
        return False

for name in ['apple', 'pear', 'peach', 'apricot']:
    result = check(name)
    if result:
        print(str.upper(name))
    else:
        print(str.lower(name))

def fun(a, b):
    c = a*b
    return c

for x in range(2, 4):
    for y in range(2, 4):
        print(x, y, fun(x, y))
```python
(22)
def f1(a):
    if a < 0:
        return 0
    elif a < 2:
        return 1
    elif a < 4:
        return 2
    else:
        return 3

for x in range(5):
    print ('*' * f1(x))

(23)
for x in range(1.0, 10.0):
    print(x)
```
Rewrite the Following For loops as While loops

```python
total = 0
for x in range(7):
    total += x
print("total=", total)
```

```python
for x in range(100, 50, -10):
    print("x =", x)
```

```python
for x in [1, "red", 3.0]:
    print(x)
```

Rewrite the Following While loop as a For Loops

```python
x = 20
product = 1
while x > 0:
    product *= x
    x -= 2
print("product=", product)
```
Rewrite the following program without any functions

```python
def sumup(a, b):
    result = a + b
    return result

total = sumup(5, 4)
print(total)
print(sumup("dog", "cat"))
```
Write programs:

1. Write a program that reads in a number N and prints out N! (N factorial).

2. The following infinite series adds up to the number 2:

\[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} \ldots. \]

Write a program that computes a finite number of terms for this infinite sequence. Your program should first prompt the user to enter a number of terms N. Then it computes the sum of the first N terms of the series in a loop. Finally, the program should print out the result.

3. Write a function that picks a card from a deck of 52 cards randomly. The function returns the card (1 - 13) and the suit (hearts, diamonds, spades, clubs).

4. Write a program that picks a card 100 times, using the above function. Print out the average number of spades, and the average number of aces picked. Print out the average number of times the ace of spades was picked.

5. Write a program that prompts the user to enter the amount of rainfall in inches for each month in a year, and prints out the average. Your output should look like:

   Month 1:  4 inches
   Month 2:  12 inches
   Month 3:  15 inches
   ....
   Average yearly rainfall:  10 inches

6. Write a function with one argument, which is a name. The function returns True if the name begins with a letter A through R, and returns False otherwise.
7. Write a program that repeatedly prompts the user to enter a name. Using the function from the previous question, count up the number of names that begin with A through R, and count up the number of names that begin with S through Z. Terminate the loop when the user enters -99. Then, print out the percentage of names in each group.

8. Write a function whose parameter is a number, which represents cents. The function converts this number to dollars and cents, and returns both results.

9. Write a function called print_rect, whose parameters are the width and length of a rectangle. The function prints the rectangle filled in with the character ‘+’. For example, if you call the function as `rect(3, 4)`, your program should print:

```
++++
++++
++++
```

10. Write a function called double, whose parameters are the width and length of a rectangle. It calls print_rect to print the rectangle with the width and length doubled.

11. Write a program that computes the grocery bill for 7 days. For each day, prompt the user to enter the number of items bought, and then to enter the price of each item. For each day, print out the total cost of all the items.

12. Write a program that prints the numbers 100-1 in a 10x10 table:

```
100 99 98 97 96 95 94 93 92 91
90 89 88 87 86 85 84 83 82 81
80 79 78 77 76 75 74 73 72 71
70 69 68 67 66 65 64 63 62 61
60 59 58 57 56 55 54 53 52 51
50 49 48 47 46 45 44 43 42 41
40 39 38 37 36 35 34 33 32 31
30 29 28 27 26 25 24 23 22 21
20 19 18 17 16 15 14 13 12 11
10  9  8  7  6  5  4  3  2  1
```