CMPSCI 119 MIDTERM #1.5

UMass Spire ID Number

Enter your name as: **Last, First**, as in: **Smith, Fred**

You may use any of your notes, the *Computer Science Companion*, or any materials on the class Web site.

This exam is optional and is to supplement the score you got on Midterm #1, since we are not allowed to go retrieve the actual exam papers. Not doing this exam, or doing it and doing poorly, CANNOT hurt your grade on the first midterm. The score on this exam will be used to scale the points you missed on the first midterm. For example, if you scored 60 on the first midterm and 80 on this exam, you'll get back 80% of the 40 points you missed, or 32 points, which will be added to the Midterm #1 score, giving you a 60+32=92.

NOTE: Your **LAST** submission will be the one graded. You get ONE FREE RESUBMIT in case of an "OMG, I screwed up an answer" moment. However, there will be a 5-point penalty assessed for the third submission, a 10-point penalty for the fourth submission, a 15-point penalty for the fifth submission, and so on. You will **NOT** receive an immediate feedback of your score when you submit the exam. Make sure it is right before you hit the SUBMIT button!

Click here to Submit Midterm 1.5 Information

Question #1 (20 Points)

Short Answer Here are two ways of stepping through every character in a string called **S**. Do they do the same things? What makes them different, if anything? What is known or not known in the body of each of the loops?

```
# OPTION 1
S = input("Please enter a string --- ")
for CH in S:
    # Do something interesting with CH

# OPTION 2
S = input("Please enter a string --- ")
for I in range(len(S)):
    CH = S[I]
    # Do something interesting with CH
```

```
They effectively do the same thing.  That is,
they both execute the body of the loop each
time with a successive character from S.
During the first pass CH will be the first
character of S, during the second pass CH will
be the second character, and so on.

They differ in that OPTION 2 **also** knows
where the character is located within S
(through loop control variable I).  OPTION 1
does NOT know where each character came from
within S.
```

Question #2 (20 Points)

Short Answer Using our graphics system developed in lecture and used in labs #3 and #4, describe in words what happens on the canvas as a result of executing the following code in the main program (**Canvas** has already been properly defined and initialized).

```
X = 100
Y = 150
for Row in range(8):
    YOffset = Y + Row
    for Col in range(8):
        XOffset = X + Col
        setPixel(Canvas,XOffset,YOffset,black)
```

It plots an 8x8 black square on the canvas, where the upper left corner of the square is at coordinates <100,150>.

Note that this is very similar to the code used in Lab #4 to plot the 8x8 text characters on the canvas.

Question #3 (20 Points)

Rewrite the following code to replace the **for**-loop with a **while**-loop that does *exactly* the same thing (with the same variables).

```
L = [3,9,2,5,4,7,2,5,4,7,8]
for I in range(len(L)):
    X = L[I]
    print (I,X)
```

```
L = [3,9,2,5,4,7,2,5,4,7,8]
I = 0
while (I < len(L)):
    X = L[I]
    print (I,X)
    I = I + 1
```

Question #4 (20 Points)

Write a complete function called **Summer** in the box below, with one integer argument **N**, that computes and returns (but does not print) the sum of all integers from 1 up through and including **N**. Parameter **N** must have a default value of 100.

```
def Summer(N=100):
    Total = 0
    for I in range(1,N+1):
        Total = Total + I
    return Total
```

Question #5 (20 Points)

The code below to compute the distance between two points <X1,Y1> and <X2,Y2> contains a number of errors. Fix all the errors by changing the existing lines so that the function runs correctly. Do not delete or add any code, or rearrange any lines; just fix what what you see here. If you need to, you can reset the problem by clicking the button below, but you will lose any changes that you've made:

Original:

```
import Math

define Distance (X1,Y1,X2,Y2)
    deltaX = X2 - X1   Difference in X
    deltaY = Y2 - Y1   Difference in Y
    DX2 = DeltaX * DeltaX
    DY2 = DeltaY * DeltaY
    Result = sqrt(DX2 + DY2)
return Result
```

Fixed:

```
import math

def~~ine~~ Distance (X1,Y1,X2,Y2):
    DeltaX = X2 - X1   # Difference in X
    DeltaY = Y2 - Y1   # Difference in Y
    DX2 = DeltaX * DeltaX
    DY2 = DeltaY * DeltaY
    Result = math.sqrt(DX2 + DY2)
    return Result                (indent was wrong)
```