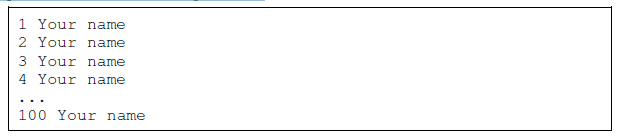
1. Write a program that prints your name 100 times.

2. Write a program to fill the screen horizontally and vertically with your name. [Hint: add the

option end='' into the **print** function to fill the screen horizontally.]

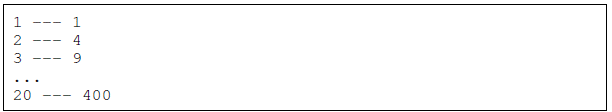
3. Write a program that outputs 100 lines, numbered 1 to 100, each with your name on it. The

output should look like the output below.



4. Write a program that prints out a list of the integers from 1 to 20 and their squares. The output

should look like this:



5. Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.

6. Write a program that uses a for loop to print the numbers 100, 98, 96, . . . , 4, 2.

7. Write a program that uses exactly four for loops to print the sequence of letters below.



8. Write a program that asks the user for their name and how many times to print it. The program

should print out the user’s name the specified number of times.

9. The Fibonacci numbers are the sequence below, where the first two numbers are 1, and each

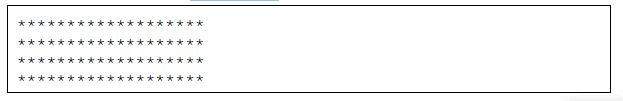
number thereafter is the sum of the two preceding numbers. Write a program that asks the

user how many Fibonacci numbers to print and then prints that many.



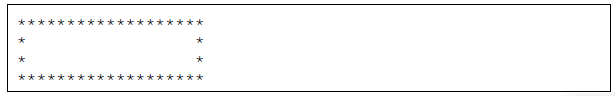
10. Use a for loop to print a box like the one below. Allow the user to specify how wide and how

high the box should be



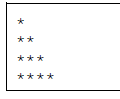
11. Use a for loop to print a box like the one below. Allow the user to specify how wide and how

high the box should be.



12. Use a for loop to print a triangle like the one below. Allow the user to specify how high the

triangle should be.



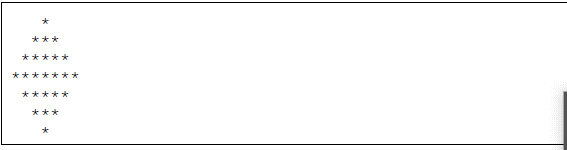
13. Use a for loop to print an upside down triangle like the one below. Allow the user to specify

how high the triangle should be.



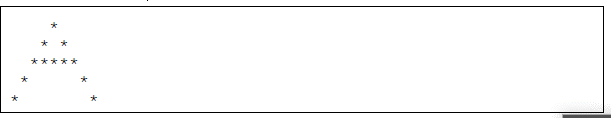
14. Use for loops to print a diamond like the one below. Allow the user to specify how high the

diamond should be.



15. Write a program that prints a giant letter A like the one below. Allow the user to specify how

large the letter should be.



QUESTIONS ON WHILE LOOP

1) A good program will make sure that the data its users enter is valid. Write a program that

asks the user for a weight and converts it from kilograms to pounds. Whenever the user

enters a weight below 0, the program should tell them that their entry is invalid and then ask

them again to enter a weight. [Hint: Use a while loop, not an if statement].

2) Write a program that asks the user to enter a password. If the user enters the right password,

the program should tell them they are logged in to the system. Otherwise, the program

should ask them to reenter the password. The user should only get five tries to enter the

password, after which point the program should tell them that they are kicked off of the

system.

3) Write a program that allows the user to enter any number of test scores. The user indicates

they are done by entering in a negative number. Print how many of the scores are A’s (90 or

above). Also print out the average.

4) Modify the higher/lower program so that when there is only one guess left, it says 1 guess,

not 1 guesses.

5) The GCD (greatest common divisor) of two numbers is the largest number that both are divisible

by. For instance, gcd(18, 42) is 6 because the largest number that both 18 and 42 are

divisible by is 6. Write a program that asks the user for two numbers and computes their gcd.

Shown below is a way to compute the GCD, called Euclid’s Algorithm.

• First compute the remainder of dividing the larger number by the smaller number

• Next, replace the larger number with the smaller number and the smaller number with

the remainder.

• Repeat this process until the smaller number is 0. The GCD is the last value of the larger

number.

6) A 4000-year old method to compute the square root of 5 is as follows: Start with an initial

guess, say 1. Then compute



Next, take that 3 and replace the 1’s in the previous formula with 3’s . This gives



Next replace the 3 in the previous formula with 7*=*3. This gives



If you keep doing this process of computing the formula, getting a result, and plugging it back

in, the values will eventually get closer and closer to



This method works for numbers other than 5. Write a program that asks the user for a number and uses this method to estimate the square root of the number correct to within 10-10

The estimate will be correct to within 10-10

when the absolute value of the difference between consecutive values is less than 10-10

7) Write a program to play the following simple game. The player starts with $100. On each

turn a coin is flipped and the player has to guess heads or tails. The player wins $9 for each

correct guess and loses $10 for each incorrect guess. The game ends either when the player

runs out of money or gets to $200.

8) Ask the user to enter the numerator and denominator of a fraction, and the digit they want to

know. For instance, if the user enters a numerator of 1 and a denominator of 7 and wants to

know the 4th digit, your program should print out 8, because 1/7 = 142856... and 8 is the 4th

digit.

**Algorithm Workbench**

1. Write a while loop that lets the user enter a number. The number should be multiplied

by 10, and the result assigned to a variable named product. The loop should iterate as

long as product is less than 100.

2. Write a program that accepts two numbers by the user, checks if the denominator is

zero, divides them and then prints their division. If the denominator is zero, it should

prompt the user “Division is not possible”.

3. Write a for loop that displays the following set of numbers:

0, 10, 20, 30, 40, 50 . . . 1000

4. Write a loop that asks the user to enter a number. The loop should iterate 10 times and

keep a running total of the numbers entered.

5. Write a program that accepts a number entered by the user and prints its Fibonacci

series up to 10 times.

6. Rewrite the following statements using augmented assignment operators.

a. x = x + 1

b. x = x \* 2

c. x = x / 10

d. x = x − 100

7. Write a program that accepts a number entered by the user and validates if it is odd or

even. If it is even, print a series up to 4 using range() and if it’s odd print a series up to 7.

8. Write code that prompts the user to enter a positive nonzero number and validates the

input.

9. Write code that prompts the user to enter a number in the range of 1 through 100 and

validates the input.

**Programming Exercises**

**1. Bug Collector**

A bug collector collects bugs every day for five days. Write a program that keeps a running

total of the number of bugs collected during the five days. The loop should ask for the

number of bugs collected for each day, and when the loop is finished, the program should

display the total number of bugs collected.

**2. Calories Burned**

Running on a particular treadmill you burn 4.2 calories per minute. Write a program that

uses a loop to display the number of calories burned after 10, 15, 20, 25, and 30 minutes.

**3. Budget Analysis**

Write a program that asks the user to enter the amount that he or she has budgeted for

a month. A loop should then prompt the user to enter each of his or her expenses for the

month and keep a running total. When the loop finishes, the program should display the

amount that the user is over or under budget.

**4. Distance Traveled**

The distance a vehicle travels can be calculated as follows:

*distance* = *speed*  x *time*

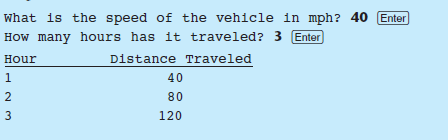
For example, if a train travels 40 miles per hour for three hours, the distance traveled is

120 miles. Write a program that asks the user for the speed of a vehicle (in miles per hour)

and the number of hours it has traveled. It should then use a loop to display the distance

the vehicle has traveled for each hour of that time period. Here is an example of the desired

output:



**5. Average Rainfall**

Write a program that uses nested loops to collect data and calculate the average rainfall

over a period of years. The program should first ask for the number of years. The outer loop

will iterate once for each year. The inner loop will iterate twelve times, once for each month.

Each iteration of the inner loop will ask the user for the inches of rainfall for that month.

After all iterations, the program should display the number of months, the total inches of

rainfall, and the average rainfall per month for the entire period.

**6. Celsius to Fahrenheit Table**

Write a program that displays a table of the Celsius temperatures 0 through 20 and

their Fahrenheit equivalents. The formula for converting a temperature from Celsius to

Fahrenheit is



where *F* is the Fahrenheit temperature and *C* is the Celsius temperature. Your program must

use a loop to display the table.

**8. Sum of Numbers**

Write a program with a loop that asks the user to enter a series of positive numbers. The

user should enter a negative number to signal the end of the series. After all the positive

numbers have been entered, the program should display their sum.

**9. Ocean Levels**

Assuming the ocean’s level is currently rising at about 1.6 millimeters per year, create an

application that displays the number of millimeters that the ocean will have risen each year

for the next 25 years.

**10. Tuition Increase**

At one college, the tuition for a full-time student is $8,000 per semester. It has been

announced that the tuition will increase by 3 percent each year for the next 5 years. Write

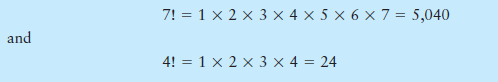
a program with a loop that displays the projected semester tuition amount for the next 5

years.

**11. Calculating the Factorial of a Number**

In mathematics, the notation *n*! represents the factorial of the nonnegative integer *n* . The

factorial of *n* is the product of all the nonnegative integers from 1 to *n*. For example,



Write a program that lets the user enter a nonnegative integer and then uses a loop to calculate

the factorial of that number. Display the factorial.

**12. P opulation**

Write a program that predicts the approximate size of a population of organisms. The

application should use text boxes to allow the user to enter the starting number of organisms,

the average daily population increase (as a percentage), and the number of days the

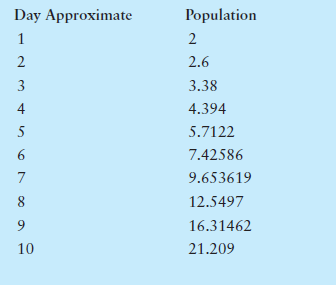
organisms will be left to multiply. For example, assume the user enters the following values:

Starting number of organisms: 2

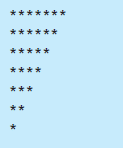
Average daily increase: 30%

Number of days to multiply: 10

The program should display the following table of data:



**13. Write a program that uses nested loops to draw this pattern:**



**14. Write a program that uses nested loops to draw this pattern:**

