

Day 3 Exercises

Section D. Examples that require loop construct

– use while or do..while loop construct for the following:

1. Write a C# program that would keep prompting you to enter an integer number over and over again until you enter the number 88. If you enter 88 the computer should say:
 "Lucky you..."
and exit the program.
2. Use Euclid's Algorithm given below to determine the LCM and HCF for given two integer numbers.
 - Take in as input two numbers A and B.
 - Subtract the smaller of the two numbers from the Larger Number and assign the answer to the larger number.
 - The above process is repeated until both the numbers are equal, say X.
 - Apparently the residual number (X) that we have obtained is the HCF.
 - LCM could then be computed using the formula $(A*B)/HCF$
 - Print out your answers.

A	B	HCF	LCM
8	4	4	8
120	2000	40	6000

3. *Guess the Number Game:* Write a C# program that would let you guess the number that the computer has in its "mind". Computer thinks of an integer between 0 and 9.
 - a. The program uses the random number function to first "think of" a number. It should then prompt you for a guess. If your guess is correct, then it would congratulate you and tell out how many attempts that you took to make the guess.
 - b. Modify the program you wrote in 3(a) so that in addition to the basic guessing function, it would also say "You are a Wizard!" if you succeed in the first two attempts or say "You are a good guess" if you make it next three attempts else it would say "You are lousy!" Every time you make a wrong guess, the program would prompt "Try again" and accept another guess. The program repeats until you have made the correct guess.

4. Using iteration write a C# program to determine the square root of a given number (N). If required, your instructor would explain the method using a numerical example. Many efficient methods are available; we use a simple one for demonstrating the looping.

STEPS:

- a. Take as input the number for finding the square root.
- b. Take a random number between 1 and N using the function RND. Let the integer be called a Guess, G (not necessarily an integer).
- c. If the Guess is correct then $G * G$ should be N.
- d. If not use the following formula iteratively until $G * G$ approximates to N to an accuracy of 5 decimal places:

$$G = \frac{(G + N/G)}{2}$$

Input	Output
0	0.000
25	5
3	1.732

Note: $G * G$ approximates to N to an accuracy of 5 decimal places means that the difference between value of $(G * G)$ and N is less than 0.00001

Section E Examples that require loop construct with counters

– use *for* loop construct for the following:

1. Given a number find out its factorial.

Write two different C# program variations for the problem:

- a. Using increment counter
- b. Using a decrement counter.

Carefully study the similarities and differences between the two approaches.

List the conditions, if any, under which your program is likely to fail

2. Write a program to print all numbers between 1 and 10 with the values of its inverse, square root and square as below:

NO	INVERSE	SQUARE ROOT	SQUARE
1.0	1.0	1.0	1.0
2.0	0.5	1.414	4.0
3.0	0.333	1.732	9.0
4.0	0.25	2.0	16.0
5.0	0.2	2.236	25.0
6.0	0.167	2.449	36.0
7.0	0.143	2.646	49.0
8.0	0.125	2.828	64.0
9.0	0.111	3.0	81.0
10.0	0.1	3.162	100.0

3. Given an integer as input determine whether the number is a prime number or not. Your program should output “Prime” or “Not Prime” as the case may be.

A Prime Number is one which is only divisible by one and itself.

Consider how the efficiency of the program can be improved. Normally the order of complexity is proportional to the number of times a loop is executed. A more formal definition of “Analysis of Algorithms” will be dealt in a later course.

4. Given an integer as input write a C# program to determine whether the number is a Perfect Number or not.

A perfect number is one for which the sum of its factors (including number one) add up to the number itself. For example number six is a perfect number because,

$$6 = 1 + 2 + 3.$$

5. Modify the Prime Number C# program to print out all the prime numbers from 5 to 10000.
6. Modify the Perfect Number C# program to print out all the perfect numbers from 1 to 1000.

Section F. Examples involving String Handling

1. Program to count the number of vowels in a given phrase and print out the number of each vowel. (a, e, i, o & u are vowels)
 - a. Write a program to read a phrase from the console and count the number of vowels there are in the phrase. You should substring one character at a time and match it to the vowels and increment the counter.
 - b. Make your program explicitly count the number of occurrences of each vowel ie: number of a's, number of e's etc.)

Input	Output
Institute of Systems Science	Total vowels: 9 A: 0 E: 4 I: 3 O: 1 U: 1

2. Write a C# program to determine if a given string is a palindrome.

Your program should take a string from the console and test if the word is a palindrome or not using the approach explained by your instructor (you are expected to follow the steps given by the instructor)

 - A palindrome is a word/phrase that reads the same forwards or backwards.
 - Examples: ABBA, 747, radar, madam