**Ekaba Bisong**
**Programming in C++**
**University of Calabar**

Assignment #3
June 15, 2015

*Question adapted from C++ How to Program, P.J. Deitel, H.M. Deitel. -- 8th ed*

# Assignment #3

## 1. Guess-the-Number Game

Write a program that plays the game of "guess the number" as follows: Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 1000
The program then displays the following:

```
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
```

```
1. Excellent! You guessed the number!
   Would you like to play again (y or n)?
2. Too low. Try again.
3. Too high. Try again.
```

If the player's guess is incorrect, your program should loop until the player finally gets the number right. Your program should keep telling the player *Too high* or *Too low* to help the player "zero in" on the correct answer.

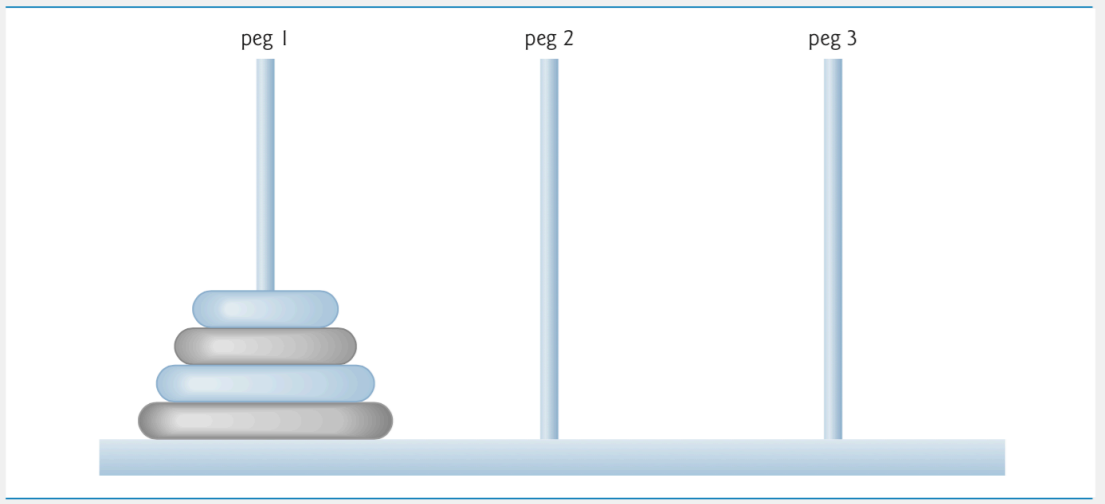
Modify the program of to count the number of guesses the player makes. If the number is 10 or fewer, print, "*Either you know the secret or you got lucky!*" If the player guesses the number in 10 tries, then print "*Ahah! You know the secret!*" If the player makes more than 10 guesses, then print "*You should be able to do better!*" Why should it take no more than 10 guesses? Well, with each "good guess" the player should be able to eliminate half of the numbers. Now show why any number from 1 to 1000 can be guessed in 10 or fewer tries.

## 2. Towers of Hanoi

You studied functions that can be easily implemented both recursively and iteratively. In this exercise, we present a problem whose recursive solution demonstrates the elegance of recursion, and whose iterative solution may not be as apparent.

The Towers of Hanoi is one of the most famous classic problems every budding computer scientist must grapple with. Legend has it that in a temple

in the Far East, priests are attempting to move a stack of golden disks from one diamond peg to another.




The initial stack has 64 disks threaded onto one peg and arranged from bottom to top by decreasing size. The priests are attempting to move the stack from one peg to another under the constraints that exactly one disk is moved at a time and at no time may a larger disk be placed above a smaller disk. Three pegs are provided, one being used for temporarily holding disks. Supposedly, the world will end when the priests complete their task, so there is little incentive for us to facilitate their efforts.

Let's assume that the priests are attempting to move the disks from peg 1 to peg 3. We wish to develop an algorithm that prints the precise sequence of peg-to-peg disk transfers.

If we were to approach this problem with conventional methods, we would rapidly find ourselves hopelessly knotted up in managing the disks. Instead, attacking this problem with recursion in mind allows the steps to be simple. Moving n disks can be viewed in terms of moving only n – 1 disks (hence, the recursion), as follows:

a) Move *n* – 1 disks from peg 1 to peg 2, using peg 3 as a temporary holding area.
b) Move the last disk (the largest) from peg 1 to peg 3.
c) Move the *n* – 1 disks from peg 2 to peg 3, using peg 1 as a temporary holding area.

The process ends when the last task involves moving n = 1 disk (i.e., the base case). This task is accomplished by simply moving the disk, without the need for a temporary holding area. Write a program to solve the Towers of Hanoi problem. Use a recursive function with four parameters:

a) The number of disks to be moved
b) The peg on which these disks are initially threaded
c) The peg to which this stack of disks is to be moved
d) The peg to be used as a temporary holding area

Display the precise instructions for moving the disks from the starting peg to the destination peg. To move a stack of three disks from peg 1 to peg 3, the program displays the following moves:

1 → 3 (This means move one disk from peg 1 to peg 3.)
1→2
3→2
1→3
2→1
2→3
1→3

**3. Function Template minimum**

Write a program that uses a function template called minimum to determine the smaller of two arguments. Test the program using integer, character and floating- point number arguments.