Western Montana High School Programming Contest

Morning Session, March 17, 2012

First of all, a Top O' The Morning to you on this wonderful St. Patty's Day!!!

Rainbow Sort

Everyone knows that the pot of gold lies at the end of the rainbow, and the most important thing about a rainbow is its graceful arc. In honor of this most beautiful thing on this most Irish day, your job is to implement a Rainbow sorting algorithm. The Rainbow Sort will read in a list of positive integers that is ended by a "-1". There will be fewer than 100 integers in the list. The Rainbow Sort will then print out those integers such that the odd integers are printed in increasing sorted order first, and the even integers are printed in decreasing sorted order after them.

Example 1:

Input:

```
1 2 3 4 5 6 7 8 9 10 -1
```

Output:

```
1 3 5 7 9 10 8 6 4 2
```

Example 2:

Input:

```
59801 59601 59102 52358 90210 90125 -1
```

Output:

```
59601 59801 90125 90210 59102 52358
```

4 Leaf Fibonacci

Lots of programming contests have some sort of variation on a Fibonacci problem in them. The first Fibonacci number is 1, and the second is 1. After that, to get the next Fibonacci number in the sequence, just add together the previous two. So the third is 1+1=2, the fourth is 1+2=3, the fifth is 2+3=5 and so on.
But Fibonacci wasn't Irish - he was very Italian. So in honor of the day and the four-leaf clover, our variation of Fibonacci uses the following definition: The first and third 4-leaf Fibonacci numbers are 1. The second and fourth 4-leaf Fibonacci numbers are 0. After that, each new 4-leaf Fibonacci number is the sum of the previous **four** numbers, so the fifth is 1+0+1+0=2. The sixth is 0+1+0+2=3. The seventh is 1+0+2+3=6. The eighth is 0+2+3+6=11.
Your program must print out all the 4-leaf Fibonacci numbers less than a million.

Character Art Gem

Write a program to generate a character-art gem (hexagon) of a given size. The outline of the gem is to be asterisks and the "white space" in the diamonds should actually be period characters (to make it easier to check). The program will read in a single integer, generate a hexagon of that size, and quit. The integer specifies how many asterisks there are on each side of the hexagon.

Example 1:
Input:

```
5
```

Output:

```
....*****
...*.....*
..*.......*
.*.........*
*...........*
.*.........*
..*.......*
...*.....*
....*****
```

Irish Feast

The O'Malleys are planning their St. Patrick's Day dinner at their house in County Cork. Unfortunately, they forgot to go to the grocery store yesterday and since it's already St. Patrick's Day, all the stores are closed and they are stuck trying to feed everyone with just the ingredients that they have on hand: 3 pounds of corned beef, 10 pounds of potatoes, and 8 pounds of cabbage.

Your job is to write a program that will help the O'Malleys figure out which feast recipe will feed the most people. The user of the program will specify how much of each of the ingredients are required per person eating the feast. Your job is to figure out how many people will be fed by the recipe before the first ingredient runs out.

The input to your program will be three decimal numbers: the amount of corned beef per serving, the amount of potatoes per serving and the amount of cabbage per serving. Your program will print out the number of people served before the first ingredient runs out, as well as a list of how much is left of each of the ingredients. Remember, there is no such thing as 0.8 of a person.

Example 1:

Input:

```
1.0 1.0 1.0
```

Output:

```
You can feed 3 people with this recipe and you will have left
0.0 pounds of corned beef
7.0 pounds of potatoes
5.0 pounds of cabbage
```

Example 2:

Input:

```
0.1 1.5 1.0
```

Output:

```
You can feed 6 people with this recipe and you will have left
2.4 pounds of corned beef
1.0 pounds of potatoes
2.0 pounds of cabbage
```

Prime Distances

A prime number, P, is an integer greater than 1 whose only factors are itself and 1. The first prime number (P_1) is 2, followed by P_2=3, P_3=5, P_4=7, and so on. There appears to be no way to predict, given a prime p, the next larger prime, although it can b shown that there are an infinite number of primes.

Write a program that inputs an integer, n, where $1 < n \leq 10{,}000$ and determine the nth prime, P_n , and the largest distance between two consecutive primes, P_i and P_{i+1} where i is less than n. Output P_n , P_i , P_{i+1} and the distance between P_i and P_{i+1} .

Your program must perform this task in less than 30 seconds running time to qualify as a solution.

Example 1:

Input:

```
11
```

Output:

```
P[11] = 31
P[9] = 23
P[10] = 29
distance = 6
```

Example 2:

Input:

```
10
```

Output:

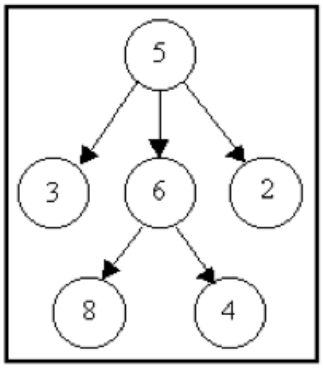
```
P[10] = 29
P[9] = 23
P[10] = 29
distance = 6
```

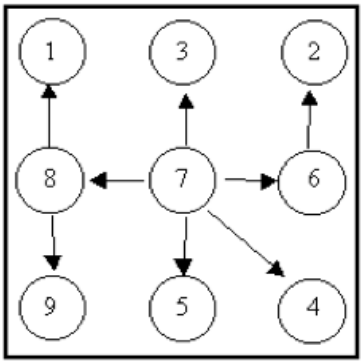
Is it a tree?

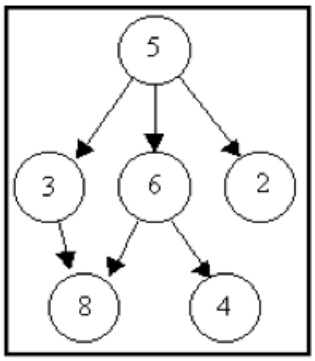
A tree is a well-known data structure that is either empty (null, void, nothing) or it is a set of one or more nodes connected by directed edges between nodes satisfying the following properties

1. There is exactly one node, called the root, to which no directed edges point
2. Every node except the root has exactly one edge pointing to it.
3. There is a unique sequence of directed edges from the root to each node.

For example consider the illustrations below in which nodes are represented by circles and edges are represented by lines with arrowheads. The left image is a tree with root 5. The middle image is a tree with root 7. But the right image is not a tree because node 8 has two edges pointed at it.







In this problem you will be given a list of edges (guaranteed to be no more than 100 edges). Each edge is described as a pair of positive numbers, with the first number being the node where the edge leaves, and the second number being the node where the edge arrives. The end of the edge list will be marked by the "edge" 0 0. Your program is to print out if it is a tree or not, and if so, what the root node is.

Example 1 (the left image):

Input:

```
6 4
6 8
5 3
5 6
5 2
0 0
```

Output:

```
This is a tree with root 5.
```

Example 2 (the right image):

Input:

```
6 4
6 8
5 3
```

```
5 6
5 2
3 8
0 0
```

Output:

```
This is not a tree.
```