The Heritage Computer Challenge *2009*

Java Division

Heritage High School
Newport News, Virginia

Welcome

Welcome to the Heritage Computer Challenge for 2009! You are to be commended for taking the time and making the effort to be here today. Have a great time and may all your programming efforts be successful!

~Mr. Charles F. Monroe, Contest Director

Instructions

The problems for this contest appear on the following pages, listed in order of difficulty. The maximum number of points you can earn is indicated under the title to each problem.

Problems are designed in the format used by The Great Computer Challenge, held annually each Spring at Old Dominion University. Some of these problems were actually used at the Great Computer Challenge in previous years.

Remember, this is a timed contest. Therefore, it is unlikely that you or anyone else will have time to complete all 5 problems in the allotted time. The winners will be the persons who earn the most points. You must earn at least 1 point to place.

How to save your work:

1. Create a folder on your personal drive K named hcc2009.
2. Create a workspace named hcc2009 and save it inside this folder.
3. Create all projects within this workspace.
4. Each solution should be saved as a project using the project name provided.

List of problems

Monroe Numbers	10 points
Chinese Animal Zodiac Year Problem	10 points
Pascal's Triangle	20 points
Palindromes	20 points
I-Soar	30 points

Monroe Numbers
(10 points)

Project name: **MonroeNumbers**

You have heard of Fibonacci numbers, but have you ever heard of Monroe numbers? The first Monroe number is 0. The second one is 0. And the third Monroe number is 1. From then on, each Monroe number is the sum of the previous 3 Monroe numbers. Write a C++ program to allow the user to indicate which Monroe number he/she would like to view (1 to 39 please) and then display the requested Monroe number. Thus, the 29^{th} Monroe number would be 4700770.

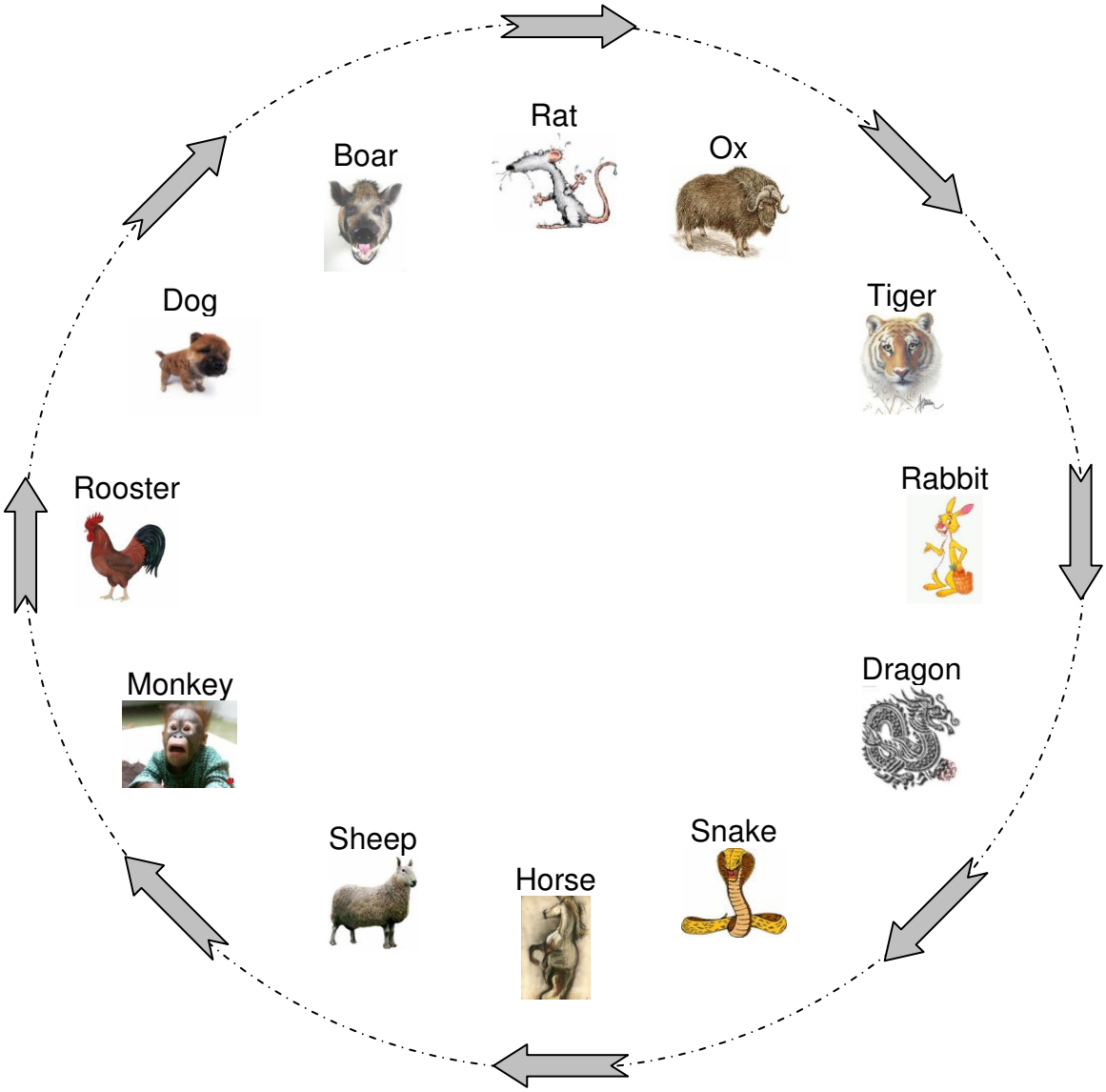
To get full credit, your program must find the 39^{th} Monroe number as quickly as it finds the 29^{th}.

The Chinese Animal Zodiac Year Problem (10 points)

Project name: **ChineseYear**

This problem is on two pages.

In the Chinese Animal Zodiac calendar, the years, for which we use numbers, are designated by twelve animals, beginning with the Rat:



Years are called “Year of the Rat”, “Year of the Ox”, etc.

When the “Year of the Boar” is reached, the next year is “Year of the Rat” again and the cycle repeats.

Although the Chinese New Year falls on different days yearly, somewhere between late January and early February based on the cycles of the moon, for the purposes of this problem, we will assume that Chinese Animal Zodiac years correspond exactly to years on our Western calendar (so years begin on January 1).

1996 was “The Year of the Rat”.

Write a Java program that allows the user to input a Western numerical year from 1500 to 2999 inclusive and then outputs the Chinese Animal Zodiac year in the format used in the sample run below. Use "was", "is", or "will be" properly. Replace "Sample" with your school's name.

Input repeats until a year outside the given range is entered.

Sample run:

```
Program to convert a Western Year
to a Chinese Animal Zodiac Year.
By the Java team from Sample High School.

Enter a year (1500-2999, any other year to quit): 1500
1500 was the Year of the Monkey
Enter a year (1500-2999, any other year to quit): 2006
2006 was the Year of the Dog
Enter a year (1500-2999, any other year to quit): 2007
2007 is the Year of the Boar
Enter a year (1500-2999, any other year to quit): 2008
2008 will be the Year of the Rat
Enter a year (1500-2999, any other year to quit): 2999
2999 will be the Year of the Sheep
Enter a year (1500-2999, any other year to quit): 3000
Press any key to continue . . .
```

Pascal's Triangle
(20 points)

Project name: **PascalsTriangle**

Blaise Pascal was a French mathematician, physicist, and religious philosopher who lived from 1623 to 1662. He was a child prodigy who was educated by his father. As a mathematician, helped create two major new areas of research—projective geometry and probability theory.

High schools students in the USA know him best for his tree of numerical values we call Pascal's Triangle, which can be used to recognize the coefficients of a quantity raised to a power. The rules for forming this triangle of integers are such that each row must start and end with 1, and each entry in a row is the sum of the two values diagonally above the new entry. Thus, four rows of Pascal's Triangle are

```
          1
        1   1
      1   2   1
    1   3   3   1
```

Write a program that takes the number of rows (1 to 10) as input and displays Pascal's Triangle for those rows.

The Heritage Computer Challenge *2009*
Java Division

The Heritage Computer Challenge *2007*
Java Division

Palindromes
(20 points)

Project name: **Palindromes**

A palindrome is a word, phrase, verse, paragraph, etc., which reads the same forwards or backwards (excluding punctuation, spacing, and capitalization). For example, the following are palindromes:

Madam, I'm Adam.
Poor Dan is in a droop.
Now Sir, a war is never even. Sir, a war is won.
Sue Zues.
Evade Dave.

Write a program which takes an arbitrary list of alphabetic characters and determines if it is a palindrome or not.

Input for each palindrome will not exceed 80 characters. Note that blanks (spaces), capitalization, and punctuation do not affect the determination of a palindrome. Numeric characters will not occur in the input. Your program should consider only one palindrome at a time, i.e., for each run of the program, there is only one palindrome to read.

Output should be PALINDROME if the expression is a palindrome, or NOT A PALINDROME if it is not.

Example:

Palindrome Test
Not now, no strap parts on Won Ton.
PALINDROME

I-Soar
(30 points)

This problem is on two pages.

Project name: **Soar**

The town meeting was not going well. "It's noisy", some town residents complained. "It's ugly", others stated. "It's an eyesore", many agreed. "It's here," said the mayor, "and it's not going to go away."

The cause of all this furor was the new stretch of Interstate Highway that had been just opened. Straight as an arrow, it ran along the entire northern edge of the town.

"Look," said the mayor, "we can reduce the noise and improve the view by planting trees and tall hedges along the road, but we don't have an unlimited budget. Luckily, much of the highway is already hidden by some of the buildings in our commercial district on the north. We'll see what we can do by planting in the visible gaps between the buildings."

Write a program to compute the total linear length of planting that will be required to block the view of the Interstate by an observer looking straight north (orthogonal to the highway) from the southern side of the commercial district.

INPUT

Input consists of multiple data sets. The first line in each data set contains the length of the town border adjacent to the highway, expressed as a floating point number (called L , below). A nonpositive value for this number signals the end of input.

This is followed by zero or more lines containing the positions of buildings within the commercial district. Each such line gives a pair of x positions (floating point numbers) representing the portion of the interstate whose view is hidden from view by the building. For example, the pair 10.0 and 25.0 means that a 15.0 unit portion of the interstate is hidden from view. These numbers are expressed in the same units of measurement as the length of the border, such that 0 denotes the western end of the border and L the eastern end. The end of a data set is signaled by any pair x1 , x2 for which x1 > x2 .

OUTPUT

For each data set, print one line of the form

```
The total planting length is ##
```

where ## is a floating point number, printed to one decimal place precision, denoting the total length of the Interstate visible between the buildings.

A sample run is shown on the next page.

Sample run:

```
Enter length of town border: 100.0
Enter position of building: 20.0 30.0
Enter position of building: 40.0 50.0
Enter position of building: 1.0 0.0
The total planting length is 80.0

Enter length of town border: 100.0
Enter position of building: 20.0 30.0
Enter position of building: 22.0 28.0
Enter position of building: 1.0 0.0
The total planting length is 90.0

Enter length of town border: 200.0
Enter position of building: 20.0 40.0
Enter position of building: 30.0 45.0
Enter position of building: 5.0 10.0
Enter position of building: 8.0 11.5
Enter position of building: 1.0 0.0
The total planting length is 168.5

Enter length of town border: 200.0
Enter position of building: 195.0 210.0
Enter position of building: 5.0 0.0
The total planting length is 195.0

Enter length of town border: -1.0

Press any key to continue ...
```