The Heritage Computer Challenge
2008
Heritage High School
Newport News, Virginia
Java Division

Welcome

Welcome to the Heritage Computer Challenge for 2008! 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 hcc2008.
2. Create a workspace named hcc2008 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.

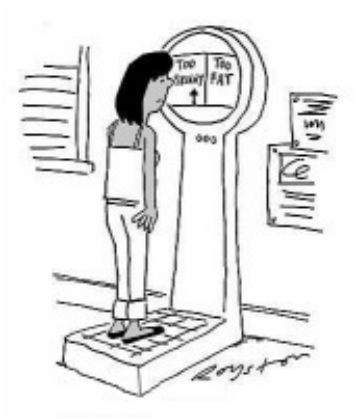
List of problems

Body Mass Index	10 points
Monroe Numbers	10 points
Pascal's Triangle	20 points
Palindromes	20 points
Hex Clock	30 points

The Heritage Computer Challenge *2008*
Java Division

Body Mass Index
(10 points)

Project name: **BodyMass**



The Body Mass Index (BMI) of a person eighteen years of age or older is calculated as follows:

```
BMI = (weight * 703) / (height * height)
```

where `weight` is in pounds and `height` is in inches.

Additionally, the National Institute of Health (NIH) categorizes individuals according to the following:

Body Mass Index	*NIH Weight Category*
less than 18.5	Underweight
18.5 up to just less than 25	Normal
25 up to just less than 30	Overweight
30 and above	Obese

Write a Java program that will allow the user to input a weight in pounds and a height in feet and inches, and then compute and output the Body Mass Index and its corresponding NIH Weight Category.

Source: United States Center for Disease Control and Prevention

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 Java 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 Heritage Computer Challenge *2008*
Java Division

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, he helped create two major new areas of research—projective geometry and probability theory.

High school 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 12) as input and displays Pascal's Triangle for those rows.

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

The Heritage Computer Challenge *2008*

Java Division

Hex Clock
(30 points)

Save in folder named: **HexClock**

The clock is one of the oldest human inventions. There have been many kinds of clocks throughout history—water clocks, rope clocks, sundials, and more recently, analog (round) clocks and digital clocks. But all these clocks have one thing in common—they divide each day into 24 hours, each hour into 60 minutes, and each minute into 60 seconds. Depending on your country or your military status, your clock might display 24 hours or 12 hours twice using a.m. for the first half of the day and p.m. for the second half. Other than that, all of our clocks are basically the same.

In the 1850s, Swedish-American inventor John W. Nystrom proposed a hexadecimal system of weights and measures that included a hexadecimal clock. Sadly, his idea never caught on. Now that we use computers so much, many of our calculations would have been much easier.

The hex clock arranges time in a more consistent and logical manner than our present clock. Instead of dividing the day into arbitrary divisions (twenty-four hours in a day, sixty minutes in an hour, etc.), the hex clock represents time of day as a single hexadecimal number.

On a hex clock, each day is divided into 1016 (hex for 16) "hex hours" numbered 0 through F. Each "hex hour" is divided into 1016 (hex for 16) "hex minutes" also numbered 0 through F. And each "hex minute" is divided into 1016 (hex for 16) "hex seconds" also numbered 0 through F. When A16 (hex for 10) hex hours, D16 (hex for 13) hex minutes, and 916 (hex for 9) seconds have passed, the time is written "A_D9" with an underscore placed between the hex hour and the hex minute.

Your job is to write a Java program that will allow the user to input regular time in 24-hour format using semicolons. Seconds should be optional, so the user might enter "9:25" or "9:25:30". The program should then convert regular time to hex time and display it using the hex time format described above ("h_ms" where h is the hex hour, m is the hex minute, s is the hex second, and each is 0 through F).

If you cannot parse your input string to separate hours, minutes, and seconds out from between the semicolons, you may have the user input hours, minutes, and seconds separately, but you will lose 10 points for doing it that way.

Sample values so you can check your results:

regular time	hex time
0:00	0_00
12:00	8_00
18:00:00	C_00

regular time	hex time
9:30	6_55
4:35:15	3_0E
22:33:44	F_0A

regular time	hex time
14:15:16	9_80
23:42:09	F_CD
23:59:59	F_FF