Lab: Floating-Point Representations
CSC 161, “Imperative Problem Solving and Data Structures”
Department of Computer Science · Grinnell College
March 1, 2016

In this lab, we’ll explore the mechanics of single- and double-precision floating-point representations of real numbers.

Pairs

Here are the pairs for this lab. Drivers are on the left.

• Sophie Gaschott and Lex Martin
• Dennis Chan and Ella Nicolson
• Cory McCartan and Lilly Webster
• Yuyin Sun and Ying Zhang
• Eli Salm and Zachary Susag
• Tyler Williams and Gemma Nash
• Josh Lavin and Erhaan Ahmad
• Mattori Birnbaum and Tanner Tufto
• Faizaan Ali and Anna Blindermann
• Saung Thuya and Colin Greenman
• Sanjay Sudhir and Jae Hong Shin

Jong Hoon Bae will replace anyone who is absent.

Exercises

Exercise 000: What is the single-precision floating-point representation of the real number +1.0?

Exercise 001: What is the smallest real number that is also an integer but does not have an exact single-precision floating-point representation and so must be approximated when stored as a float on MathLAN systems?

Exercise 010: Here is a loop in C that yields a somewhat surprising result:

```c
float foo;
for (foo = 0.0; foo < 3.0; foo += 0.3)
    (void) printf("foo = %.7f\n", foo);
```

(The format directive %.7f tells the printf function to display seven digits after the decimal point when printing the value of the float.)

Here is the output from the loop:

- foo = 0.0000000
- foo = 0.3000000
- foo = 0.6000000
- foo = 0.9000000
- foo = 1.2000000
- foo = 1.5000000
- foo = 1.8000000
- foo = 2.0999999
- foo = 2.3999999
- foo = 2.6999998
- foo = 2.9999998
Explain why the last line is printed and describe how the value of \texttt{foo} at that point differs from the floating-point representation of +3.0.

Exercise 011: What real number does the following double-precision floating-point representation express?

\begin{verbatim}
1100000001011001111000000000000000000000000000000000000000000000
\end{verbatim}

Exercise 100: Write and execute a C program that determines whether \(-0.0\) and \(+0.0\) are equal.

Exercise 101: The \texttt{math} and \texttt{float} standard libraries define the constants \texttt{INFINITY} (for what the reading calls “positive infinity”) and \texttt{DBL_MAX} (for the largest real number that can be expressed exactly as a \texttt{double}). Write and execute a C program that determines whether \texttt{DBL_MAX} is strictly less than \texttt{INFINITY}.

Exercise 110: The \texttt{math} standard library provides a \texttt{sqrt} function that takes a \texttt{double} value as argument and returns an approximation to its square root. Write and execute a C program that determines whether the square of \(\sqrt{2.0}\) is equal to 2.0. (You’ll need the \texttt{-lm} command-line compiler option to link the code for \texttt{sqrt} into your program.)

Exercise 111: Write and execute a C program that computes and writes out the square root of \texttt{INFINITY}.