

Announcements

Thursday Extras: CS Commons on Thursdays @ 4:00 pm

This week: Sriram Pemmaraju, U. of Iowa

Large Scale Distributed Graph Algorithms

Grinnell-U. Iowa 4+1 CS Program

- For Grinnell CS majors
 - Complete normal CS major in 4 years
 - Senior year, as part of regular CS major take 2 U. of Iowa Graduate Courses online
 - Iowa courses count toward Grinnell's graduate
 - Grinnell pays U. of Iowa costs
- Fifth year at U. of Iowa Master's CS program
 - Take courses at U. Iowa
 - Grinnell's CSC 341 and CSC 324 count toward MS
 - Likely have TA position at Iowa, so no tuition and some stipend

Announcements

Friday's Test

Header files to be distributed:

MyroC.h, eSpeakPackage.h

ctype.h, strings.h

Schedule: two readings/labs on today's schedule

Data representation and its consequences: TO BE DONE

Machine-level ops: optional

Today:

An expanded solution for Supplemental Problem 2

Another example of floating point representation

Consequences of floating point numbers

questions

clicker questions

Advantage and Limitations of Number Representation in Computers

Integer

- Exact storage and arithmetic (if within storage range)
- Range of numbers limited; on MathLAN machines
 - short int (16 bits): -32768 to 32767 (inclusive)
 - int (32 bits): -2147483648 to 2147483647 (inclusive)
- Overflow, if numbers go outside range

Floating Point Numbers

- Wide range of number sizes
 - float: approximately 10^{-37} to 10^{37}
 - double: approximately 10^{-307} to 10^{307}
- Limited accuracy
 - float: about 6-7 decimal digits of accuracy
 - double: about 15-16 decimal digits of accuracy
- Errors can accumulate with successive computations
- Addition not associative

Consider the following really silly program:

```
#include <stdio.h>

int main () {
    int num1 = 1500000001;
    int num2 = 1499999999;
    // note:          2147483647 is max stored int

    int avg = (num1 + num2) / 2;

    printf ("the average of %d and %d is %d\n",
           num1, num2, avg);
    return 0;
}
```

What average is printed (on MathLAN computers)?

- A. 1500000000
- B. some other positive number
- C. zero
- D. some negative number
- E. ?!@#\$%^&*()

What can you say about the way the decimal number $1/10$ is represented as a float or a double?

- A. $1/10$ (decimal) can be stored exactly as a float and/or a double.
- B. $1/10$ (decimal) is consistently stored as a number less than its true value.
- C. $1/10$ (decimal) is consistently stored as a number greater than its true value.
- D. $1/10$ (decimal) is not stored exactly as a float or a double, but the value stored might be bigger or might be smaller (with possible different results for floats and doubles).
- E. Why would this matter to anyone anyhow?

Consider the code segment

```
for (RealNumberInMath x = 0.0; x != 1.0; x += 0.1)
    printf ("%f ", x);
```

- A. This loop prints exactly 10 lines
- B. This loop prints exactly 11 lines
- C. This is an infinite loop

Same question, except declare float x

Same question, except declare double x

Consider the code segment

```
for (RealNumberInMath x = 0.0; x < 1.0; x += 0.1)
    printf ("%f ", x);
```

- A. This loop prints exactly 10 lines
- B. This loop prints exactly 11 lines
- C. This is an infinite loop

Same question, except declare float x

Same question, except declare double x

Consider the code segment

```
for (RealNumberInMath x = 1000.0; x < 1001.0; x += 0.1)
    printf ("%f ", x);
```

- A. This loop prints exactly 10 lines
- B. This loop prints exactly 11 lines
- C. This is an infinite loop

Same question, except declare float x

Same question, except declare double x