

CSI 32 LECTURE NOTES (Ojakian)

Topic 8: Types and Variables

OUTLINE

PRIMER: 2.1, 2.2, 3.5

TRANSITION GUIDE: 4.1

1. Types (including floating point issues)
 2. Sizes, bytes, and out-of-range
-

1. Types and Sizes

Note: Size measured in bytes

- (a) Include “typeinfo” to get type information.
- (b) Then get type with: typeid(BLAH).name()
- (c) Find the size of various number types with: sizeof

PROBLEM 1. *Find types of various expressions.*

- (d) Built-in types and Everything Else ...
 - i. General dicotomy for numbers in most programming languages: Integer-types versus Floating-point-types
 - ii. Only built-in: int, float, double, char, bool (see page 36)
 - iii. Variants using: long, short, signed, unsigned
 - iv. Not built-in (but in standard library with includes): string, vector, etc

- (e) Amount of memory allocated

See page 36 (and take with many grains of salt!)

- i. Bits and Bytes

NOTE: Can use library “bitset” to see binary string reps.

PROBLEM 2. *How many values can n bits store?*

PROBLEM 3. *Suppose we want to express non-negative integers with no gaps (e.g. 0, 1, 2, 3, etc). Which integers can we express with 4 bits?*

PROBLEM 4. *Suppose we want to express integers (positive and negative) with no gaps (e.g. -2, -1, 0, 1, 2, 3, etc). Which integers can we express with 4 bits? (there is some vagueness in this question!)*

PROBLEM 5. *How many values should you be able to store in some of the built-in integer types, and what happens if you go out of range?*

ii. Floating Point

Number of significant digits (in decimal) ...

PROBLEM 6. *Pick various values for float and double, and see what it printed out.*

PROBLEM 7. *For these problems ignore the detail of the exponent multiplier in scientific notation.*

- A. *Imagine a pathetic floating point with only 2 significant decimal digits. What is the fewest number of bits needed to store such a number? (Do by guess and check)*
- B. *Repeat the same question for 3 significant decimal digits. Also can you think of a method that is more systematic than guess and check?*
- C. *Now what about for C++ float and double type (assuming they have 6 and 10 significant decimal digits)?*

PROBLEM 8. *Try pushing the range on the size of float and double.*

(f) char type

- i. For representing individual characters.

PROBLEM 9. *Convert between int and char with some examples.*

PROBLEM 10. *8 bits are reserved for a char.*

- A. *How many char values can you represent? (note: they take on non-negative integral values)*
- B. *Rather than converting to int, what is more space efficient? Try that.*
- C. *At what integral value does char cycle back to the char value associated to 0.*

PROBLEM 11. *Write a program to print out all the char values. Note some “non printable characters”.*

2. Details on Floating Point

See website link for more details than you want ...

- (a) Precision: set print out precision for cout
- (b) Type `float` usually 7 digits of accuracy
- (c) Type `double` usually 16 digits of accuracy
- (d) Key point: The actual floating point number stored in memory is just an approximation of your number!
- (e) For this reason, so NOT expect comparison operators to work as you expect!
- (f) Another danger: Compounding round-off errors!