Lecture 12.2 CSI333 Characters and Strings in C

char C++ type Stroustrup (4.3); also (20.2) for today’s Internationalization Challenges.

- Holds a character, typically ASCII coded, almost universally 8 bits; unit for most HW memory addresses. Unit of size reported by C’s sizeof () operator (so chars are smallest).
- Literal values (character constants): ’a’ ’0’(=48) ’\n’ ’\0’=0
- Built-in and converts to/from other integral types.
- Signed or unsigned varies; use “signed char” or “unsigned char” if your program does arithmetic on chars.
Character handling library:

- **Documentation:** Stroustrup (20.4.2), p.600-601.  
  `man ctype` in Lab.  
  (Ch 16 of Deitel & Deitel, maybe review Sect. 5.12)

- Header file: `<ctype.h>`

- Tests and manipulations on character data.

- Functions operate in integer data and perform arithmetic operations. C/C++ converts char values to/from int values.

**Examples:**

```
int isdigit (int x); // '0'..'9'
```

- Returns a non-zero int if the value stored in x represents an ASCII digit, and 0 otherwise.

---

```
int tolower (int x)
```

Returns the lower case equivalent of the char stored in x, if it is an upper case letter; otherwise, returns x itself.

**Implementation in C++:**

```
int tolower (int x) {
    if ( (x >= 'A') && (x <= 'Z') )
        return (x - 'A' + 'a');
    else
        return x;
}
```

**Implementation in MAL:**

```
tolower:
    move $v0, $a0
    blt $v0, 65, done
    bgt $v0, 90, done
    addiu $v0, $v0, 32
    return x;
    done: jr $ra
}
```

These implementations rely on the Locale Dependent! fact that the ASCII codes for the alphabet are contiguous. This is false for the EBCDIC (old IBM) code.
Table Lookup Implementation in C++:

```c
unsigned int tolower_TABLE[ 256 ] =
{ 0, 1, ..., ..., 'a', 'b', 'c', ...
  ..., ..., 'a', 'b', 'c', ...
  ..., ..., ..., ..., 253, 254, 255 };

int tolower (int x)
{
  x = x & 0xFF; // Ignore high bits above lower 8

  return tolower_TABLE[ x ];
}
```