Software Tools

C, Unix (Linux), and tools

Types, Operators, Expressions

- Names of variables contain letters, digits and underscores. Should not start with a digit, and only library variables may start with underscore.
- Keywords like `if`, `else`, `while`, `for`, etc cannot be used as variable names.
- Variable names should be chosen carefully.
  - Should either have meaning, or be standard names
  - Short ones for local variables
  - Shorter ones for loop variables
  - Longer for global (`external`) ones.

Integer Constants

- Things like 453 are integer constants.
- Things that may need a `long` to be stored may have a trailing `L`.
  - Lower case `L` (while legal) should be avoided!
- Unsigned integers may have a trailing `U`.
- Things that start with zero like `031` are octal.
- Things that start with `0x` like `0x10` or `0xFF` are hexadecimal
- Things like `45+0x12` are integer constants.

Character constants

- Things like `'a'`, `'A'` are single characters
- Things like `'\n'`, `'\t'`, `'\r'`, `'\\'`, `'\a'` (alert, bell), `'\0'` are also single characters
- Things like `'\035'`, `'\100'` are octal representations of character constants.
- Things like `'\0x20'` (space) are hexadecimal representations of character constants
- All of them are really integers

String constants

- Things like `"I am a string"` are string constants
  - Notice that `"x"` is not the same as `'x'`
- Two constant strings next to each other are concatenated
  - Useful for multiline strings and for building strings with `#define`s
- All of them end with a null character.

String libraries

- There is a standard C library that manipulates strings
  - `strlen`: string length
  - `strcpy`: string copy
  - `strcmp`: string compare
  - `strncpy`: safe string copy
- See `man -s 3 string`

String LengthThe wordcnt Prog.

```
int strlen(char s[])
{
  int i;
  for (i=0; s[i]!='\0'; i++);
  return i;
}
```

Enumerated constants

- Just like the `#define`s but visible to the compiler
  - So that the compiler can give warnings/errors when misused
  - `enum boolean { NO, YES };`
  - `YES == 1, NO == 0`
- Can be initialized
  - `enum escapes { BELL='\a', TAB='\t', NEWLINE='\n'};`
- Values need not be distinct, but names need.

Variable Declarations/Definitions

- All variables must be declared/defined (almost)
  - There are also implicit declarations
- The declarations may have initializations
  - For automatic variables can happen many times
  - For external and static happens once
- The default initialization for static/external is zeor
- The default for automatic is garbage.

Data Types

- We have seen `float`, `double`, `long double`, `int`, `short`, `long`, `long long`.
- Also the qualifiers: `signed`, `unsigned`
- May have seen `const`, `volatile`, `static`, `extern`..
  - Eg `int strlen(const char[ ])`
- We will see what these mean in detail later.

Operators

- A whole zoo of arithmetic:
  - + - * / % ~ & | ^
- Relational and logic operators:
  - < > <= >= == != && || !

Type conversions

- We often need to convert one data type to another
- The rule is to convert a narrower type to a wider one (to minimize loss of information)
- Characters a routinely treated as integers
  - It only makes sense to treat characters as characters when we want to pack more of them in an array.

Function atoi

```
int atoi(const char s[])
{
  int i, n;

  n=0;
  for (i=0; s[i]>='0' && s[i]<='9'; i++)
    n = 10*n + (s[i]-'0');
  return n;
}
```

Portability Issues

- This works fine for ASCII
- Does not work for EBCDIC
- So just in case you ever encounter an EBCDIC computer you should use:
  - `isdigit(c)`
  - Instead of
  - `c>='0' && c<='9'`
- C has several functions that handle these subtle issues.

Problems to Play with

- Write a program that prints out the number of occurrences of Latin characters (upper or lower case is the same) in its standard input. It prints out 26 numbers that correspond to the frequency of a/A, the frequency of b/B, etc.
- Write a program that beaks lines greater than 80 columns into two. The break occurs at the last space before the 80th column.