More from the C Standard Library

Mathematical Functions in `math.h`

There are many basic maths functions in `<math.h>`: sin(x), exp(x), log(x), sqrt(x), pow(x,y), fabs(x)

All the trigonometric functions use radians!

For exponentiation in C, we must use the function $x^y = \text{pow}(x, y)$ which assumes x and y are of type double. Note that the `pow()` function is often implemented as $\exp(y * \log(x))$

For integer powers (e.g. x^3), explicit multiplication ($x * x * x$) should be used, as it's both faster and more accurate than $\text{pow}()$

Many C compilers require that you explicitly tell the compiler to link the Maths library in, via `-lm` (often in LDLIBS)

Sorting and Searching

The C Standard Library has some useful routines to sort an array and to search a sorted array for a value

`qsort()` implements an in-place array-based quick sort

`bsearch()` implements an array-based binary search

Both are generic, so take `void *` base pointers, an integer number of members, and an integer size of a member, and comparator functions (which take `const void *` pointers, and deliver a `strcmp()` like $-ve$, 0, or $+ve$ value):

```
qsort( void *base, size_t nmemb, size_t size,
        int (*compar)(const void *, const void *) );

bsearch( const void *key, const void *base,
        size_t nmemb, size_t size,
        int (*compar)(const void *, const void *) );
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct { int num; char *name; } month;
month months[] = {
  { 1, "jan" }, { 2, "feb" }, { 3, "mar" }, { 4, "apr" }, { 5, "may" }, { 6, "jun" },
  { 7, "jul" }, { 8, "aug" }, { 9, "sep" }, {10, "oct" }, {11, "nov" }, {12, "dec" }
};

static int compmonth( const void *m1, const void *m2 ) {
  month *a = (month *)m1; month *b = (month *)m2;
  return strcmp( a->name, b->name );
}

int main( int argc, char **argv ) {
  qsort( months, 12, sizeof(months[0]), &compmonth );
  for( int i = 1; i < argc; i++ ) {
    month key; key.name = argv[i];
    month *res = bsearch( &key, months, 12, sizeof(months[0]), &compmonth );
    if( res == NULL ) {
      printf( "'%s': unknown month\n", argv[i] );
    } else {
      printf( "%s: month #%d\n", res->name, res->num);
    }
  }
  return 0;
}
```

After compiling, we can run it via:

```
./qsort+search jan feb jun dec oct max may
jan: month #1
feb: month #2
jun: month #6
dec: month #12
oct: month #10
'max': unknown month
may: month #5
```

It looks each command line argument up in the sorted months table

If it's found, then it displays the month name and the corresponding month number (1..12)

If it's not found, it reports an *unknown month*

Note that the comparator function typecasts the `const void *` element pointers to `month *` pointers `a` and `b` - and compares the `a->name` and `b->name` using

`strcmp()`

The comparator function returns the result of the `strcmp()`