**Arrays of Pointers**

The following code declares an array of pointers to char:

```c
char *arr[7];
```

Each element can contain a pointer to char.

Often used for strings.

---

The following would define element 3 to point to a string:

```c
arr[3] = "Hello world";
```

The letter 'r' in the string could be addressed with:

```c
*(arr[3] + 8) or arr[3][8]
```
```c
char *month_name(int n) // return name of nth month
{
    static char *name[] =
    {
        "illegal month",
        "January",
        "February",
        "March",
        "April",
        "May",
        "June",
        "July",
        "August",
        "September",
        "October",
        "November",
        "December"
    };
    return (n < 1 || n > 12) ? name[0] : name[n];
} // END month_name
```

Ragged Array
Are These Two Declarations Different?

char a[2][16] =
    {"abc:   ", "a is for apple"};

char x[2] =
    {"abc:   ", "a is for apple"};

Command Line Arguments

int main(int argc, char *argv[])

    argc - the number of arguments on the
    command line (incl the command
    name itself)

    argv - pointer to array of pointers to char -
    an "array of strings" containing the
    command line arguments

    argv[0] - the command name itself
**Command Line Interface**

Typing the following command will produce the values shown:

```
./a.out Hello world
```

```
int main(int argc, char *argv[]) {
    int i;

    cout << "This program was called with " << argc << " arguments\n";

    for(i = 0; i < argc; i++)
        cout << argv[i] << "\n";
}
```