Arrays in C

An array is a consecutive group of storage locations all known by the same name. Individual elements are identified by a subscript.

```c
int a[5];
```

![Array Diagram]

Some Array Examples

```c
int a[5];
int b[6] = {45,28,87,11,73,55};
int c[] = {34,75,44,27,30};
char ch[] = "This is a string";

const MAXSIZE = 100;
int x[MAXSIZE];

a[0] = 5;
a[1+3] = 0;
l = 2;
a[4] = 14;
a[4+1] = 24;
```

Note use of constant!
Arrays and the for Loop

The perfect marriage!

int a[5], i;

for(i=0; i < 5; i++)
    a[i] = 0;

for(i=0; i < 5; i++)
    a[i] = i*10;

Array Input and Output

int quizzes[8];
int i, sum = 0;

for(i = 0; i < 8; i++)
{
    cin >> quizzes[i];
    sum += quizzes[i];
    cout << quizzes[i];
} // END for i
cout << sum;
A New Organization for Programs

```c
int quizzes[8];
int i, sum = 0;

// input
for(i = 0; i < 8; i++)
  cin >> quizzes[i];

// compute
for(i = 0; i < 8; i++)
  sum += quizzes[i];

// output
for(i = 0; i < 8; i++)
  cout << quizzes[i];

cout << sum;
```

Instead of one large loop, we can now have several small loops, each doing just one thing.

"C-Style" Strings

Strings are arrays of char. They are terminated with the "null" character ('\0')

```c
char str[] = "string";
```

0 's'
1 't'
2 'r'
3 'i'
4 'n'
5 'g'
6 '\0'
C-Style String Input Example

```c
char line[81];
int k;

    cin.get(line[0]);
    k = 0;
    while(line[k] != '\n')
    {
        k++;
        cin.get(line[k]);
    } // END while
    line[k] = '\0';
```

String Output Example

```c
k = 0;
    while(line[k] != '\0')
    {
        cout.put(line[k]);
        k++;
    } // END while
    cout.put('\n');
```

The same thing can be accomplished with:

```c
    cout << line << endl;
```
Example – Counting the Characters in a String

```c
n = 0;
while(line[n] != '\0')
    n++;

OR:

#include <cstring>
int strlen(line);
```

Arrays as Function Arguments

```c
// prototype
void initarray( int [], int);

// function definition
void initarray( int a[], int n)
{
    int m = 0;
    for (m = 0; m < n; m++)
        a[m] = 0;
} // end initarray
```

NOTE: arrays are ALWAYS passed by reference!

What Will Happen

```c
float x[5];
```

```c
func(x[4]);
```

What does the prototype mean?