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];
```

```plaintext
0
1
2
3
4
```

Some Array Examples

```c
int a[5];
int b[6] = {46, 23, 57, 11, 78, 66};
int c[] = {34, 76, 44, 27, 80};
char ch[] = "This is a string";

const MAXSIZE = 100;
int x[MAXSIZE];
```

```plaintext
Note use of constants
a[0] = 5;
a[1+3] = 6;
i = 2;
a[i] = 14;
a[i+1] = 24;
```
Arrays and the for Loop

The perfect marriage!

```c
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

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

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

```cpp
int quiz[8], sum;
int i;

for(i = 0; i < 8; i++)
    cin >> quiz[i];
for(i = 0; i < 8; i++)
    sum += quiz[i];
for(i = 0; i < 8; i++)
    cout << quiz[i];

cout >> sum;
```

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

Strings in C

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

```cpp
char char[] = "string";
```

<table>
<thead>
<tr>
<th>Index</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>'s'</td>
</tr>
<tr>
<td>1</td>
<td>'t'</td>
</tr>
<tr>
<td>2</td>
<td>'r'</td>
</tr>
<tr>
<td>3</td>
<td>'i'</td>
</tr>
<tr>
<td>4</td>
<td>'n'</td>
</tr>
<tr>
<td>5</td>
<td>'g'</td>
</tr>
<tr>
<td>6</td>
<td>\0</td>
</tr>
</tbody>
</table>
String Input Example

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

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

Example – Counting the Characters In a String

```cpp
m = 0;
while(line[m] != '\0')
    m++;

OR:

#include <string.h> // or <cstring>

m = strlen(line);
```
Arrays as Function Arguments

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

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

*NOTE: arrays are ALWAYS passed by reference!*

What Will Happen here?

```c
float x[100];

func(x[4]);
```

*What does the prototype for func look like?*