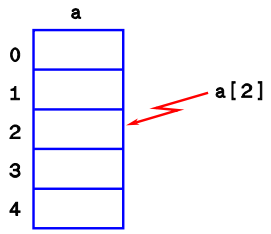
Arrays in C

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

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



Some Array Examples

```
int a[5];

int b[6] = {45,23,87,11,73,55};

int c[] = {34,76,44,27,90};

char ch[] = "This is a string";

const MAXSIZE = 100;
int x[MAXSIZE];

a[0] = 5;
a[1+3] = 6;
i = 2;
a[i] = 14;
a[i+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], 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

```
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')

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

0	's'
1	't'
2	'r'
3	'i'
4	'n'
5	'g'
6	'\0'

String Input Example

```
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';
```


Example – Counting the Characters in a String

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

OR:

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


m = strlen(line);
```

Arrays as Function Arguments

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


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 here?

```
float x[100];

func(x[4]);
```

What does the prototype for func look like?