Arrays and Matrices

Some things done automatically in Java must be done manually in C.
|------|-----|-----|-----|------|-----|------|-----|-----|-----|

```
int X[100];

variables having the same data type:
```

An array is an indexed data structure to represent several:

```
int X1, X2, ... X100;
```

Define 100 variables:

```
```

Suppose, you need to store years of 100 cars. Will you

**One-Dimensional Arrays**

```
Indices ALWAYS start at 0 in C
Can allocate multiple variables of same type
```
The name of the array is the address of the first element and so \( \bar{X} \) is the sixth element.

In C, the subscripts always start with 0 and increment by 1, and an index or subscript, e.g., \( \bar{Y} \), refers to an element of an array as accessed using the array name (like pointers).

One-Dimensional Arrays
When data structures have data added/removed, they grow/shrink respectively.

- Size must be a constant
- Subscript of last element is size-1
- Subscript of first element is 0
- Allocates memory for size elements

```c
datatype array name[size] =
```

An array is defined using a declaration

Definition and Initialization
Example

```c
int list[5];

// C does not check bounds on arrays
list[6] = 5; /* will give segmentation fault later */
```

- allocates memory for 5 integer variables
- subscript of first element is 0
- subscript of last element is 4
- C does not check bounds on arrays
- C does not do bounds checking
Arrays can be initialized at the time they are declared.
Example:

Assigning values to an array

for loops are often used to assign values to an array

\[
\text{list}[4] = 4 \\
\text{list}[3] = 3 \\
\text{list}[2] = 2 \\
\text{list}[1] = 1 \\
\text{list}[0] = 0
\]

\[
\text{int list[5], j;} \\
\text{for (j=0; j<5; j++)} \\
\text{list[j]} = j;
\]
Give a for loop to assign the below values to `tist`

Assigning values to an array
Arrays are often used to store information from a data file.
double z[4]
{
    a, b, c
} = [ ]

char letters = [ ]
{
    3, 4, 5
}

int x[10] = [-5, 4, 3]

Show the contents of the arrays defined in each of the following sets of statements.

Exercise
\( \text{data} = [\text{rand int}(10,109)] \)

\( \text{for } i = 0; i > 100; i++ \)

OR

\( \text{data} = [\text{rand}] \)

\( \text{for } i = 0; i > 100; i++ \)

Store random numbers \([10,109]\) in data

\( \text{data} = [\text{rand}] \)

\( \text{for } i = 0; i > 100; i++ \)

Store random numbers \([0,99]\) in data

\( \text{int data}[100], i \)

Exercise
Computations on arrays
Maximum in an array
\begin{verbatim}
printf("max \%d\n", max);

max = data[i]
if (max < data[i])
    max = data[i];

for (i=0; i<100; i++)
    printf("data[%d] = %d\n", i, data[i]);

for (i=0; i<100; i++)
    printf("data[%d] = %d\n", i, rand());

for (i=0; i<100; i++)
    printf("data[%d] = %d\n", i, data[i]);

max = data[0];

for (i=0; i<100; i++)
    if (data[i] > max)
        max = data[i];

printf("Find maximum value in data:\nmax = %d\n", max);
end
\end{verbatim}