Find average of values in data

Find average

\[
\text{printf} \left( \text{'avg = }\%\text{.2f'} \right) (\text{double} \text{ sum/100)};
\]

\[
\text{avg = sum/100;}
\]

\[
\text{for} (i=0; i<100; i++)
\]

\[
\text{sum = sum + data[i]};
\]

\[
\text{for} (i=0; i<100; i++)
\]

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

\[
\text{printf} \left( \text{'avg = }\%\text{.2f'} \right) (\text{double} \text{ sum/100)};
\]

\[
\text{avg = sum/100;}
\]

\[
\text{for} (i=0; i<100; i++)
\]

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

\[
\text{printf} \left( \text{'avg = }\%\text{.2f'} \right) (\text{double} \text{ sum/100)};
\]

\[
\text{avg = sum/100;}
\]
printf("%d elements are greater than avg", count);

count++;

for (i=1; i<avg; i++)
{
    count++;
    if (data[i] < avg) {
        count = 0;
    }
}

After finding the average as shown in previous slide:

Number of elements greater than average
Set array B to the reverse of array A.

Reverse an array
Reverse an Array

```c
{ 
  int pindex = 0;
  int aindex = 1;
  if (a[aindex] = [a]pindex) {
    pindex = pindex + 1;
  }
  a[aindex] = a[pindex];
}
```
So pairs

<table>
<thead>
<tr>
<th>pair</th>
<th>49</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

Find sum of every pair in data and write into pair array

<table>
<thead>
<tr>
<th>data</th>
<th>99</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>98</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>data</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Find pair sum
for (i = 0; i < 50; i++) {
}

for (i = 0; i < 100; i++) {
    for (j = 0; j < 100; j++) {
        for (k = 0; k < j; k++) {
            data[j + k] = data[j + k] + data[j - k];
        }
    }
}

Solution
Individual elements of an array can be passed as regular arguments.

Function Arguments
another argument to the function varies, so the actual size of the array is usually passed as
The actual number of array elements that are used will
time the array is declared.
The maximum size of the array must be specified at the
The array name is the address of the first element
Modifications to the array are reflected to main program
Arrays are always pass by reference

Passing Arrays to Functions
Write a function to find maximum value in the array data

**Exercist**

Note: You can't change every number.
Exercise
Exercise