CS 102 – Lecture 7

Arrays
one dimensional array

arrayName[Length of List];

const int LENGTH = 5
cchar word[LENGTH];

word = "hello"  //C style string
one dimensional-array

cchar word[6]= {'h', 'e', 'l', 'l', 'o', '\0'};
cchar word[] = {'h', 'e', 'l', 'l', 'o', '\0'};
cchar word[] = "hello";

count from 0

<table>
<thead>
<tr>
<th>H</th>
<th>e</th>
<th>l</th>
<th>l</th>
<th>o</th>
<th>\0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>
to print specific elements

arrayName[index] → value

cout<<"the first element is: 
  "<<word[0]<<endl;

cout<<"the last element is also: 
  "<<word[-1]<<endl;
2d arrays

arrayName[# rows][# columns];

int mult[12][12];

mult = {
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12},
    {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24},
    //more values until table is finished
}
multiplication table take 2

for(int i = 1; i<=12, i++)
    for(int j = 1; j<=12; j++)
        mult[i-1][j-1] = i*j;

arrayName[row index][column index] = value
passing arrays to functions

```c
int findmax(int vals[]);
int findmax(int vals[], int numels){
    int i, max = vals[0];
    for (i = 1; i<numels; i++)
        if (max<vals[i])
            max = vals[i];
    return max;
}
```
Vectors

```cpp
#include<vector>

vector <int> months(10);
for(int i = 0, i<months.size(), i++)
    months[i] = i+2;

Doing this without vectors requires creating a new array and copying all the elements of the existing array.
Linear Search

Search for $x = 0$

Scan the entire array, stop at index containing 0
Linear Search

Search for $x = 0$

Scan the entire array, stop at index containing 0
Binary Search

looking for 5

array has to be sorted, works by constantly restricting range
Selection Sort

<table>
<thead>
<tr>
<th>5</th>
<th>2</th>
<th>1</th>
<th>4</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Swap first element with smallest element in array
Consider rest of list as full list
repeat steps 1 and 2 until there's no more list
| 7 | 1 | 2 | 3 | 4 | 5 | 6 | 8 |

src: https://en.wikipedia.org/wiki/Selection_sort
bubble sort

src: https://en.wikipedia.org/wiki/Bubble_sort
Animated

http://algorithms.openmymind.net/
write a single function that returns the range of an input array of length MAXLEN;

use findmax as a sample:

```c
int i, max = vals[0];
for (i = 1; i<numels; i++)
    if (max<vals[i])
        max = vals[i];
```