Searching Arrays

Searching Arrays

- We search an array to find a particular element in an array
- For example, we might like to search an array of student grades for all students who got higher than 90% (i.e. A's)
- How would we do this?

Sequential or Linear Search

- Compare each element of the array with the value (or key) that we are searching for
- This is called linear or sequential search
- Linear Search Algorithm:
  - For each array element
    - If the current element contains the target
    - Return the subscript of the current element
  - Return -1

Write the function `findElement`

```
void findElement(int [], int, int&, int);

int main()
{
   int grades[10];
   int element, index = -1;
   for(int i = 0; i < 10; i++)
        cin >> grades[i];
   cout << "Which element would you like to find?" << endl;
   cin >> element;
   findElement(grades, element, index, 10);
   if(-1 == index)
        cout << "Element could not be found!" << endl;
   else
        cout << "Element was found at index " << index <<
                        endl;
}
```


Function to find element

27.1 Problem

- Write a function to return the index of the smallest element in a subarray
- A subarray is a section of an array. The subarray is determined by its starting and ending indexes
- The function will have the following arguments:
  - The array,
  - The starting index of the subarray,
  - The ending index of the subarray,
  - The index of the smallest element.

Function `findIndexOfMin`

```
void findIndexOfMin(const int x[], int
  startIndex, int endIndex, int& index)
{
  You fill in the rest
}
```


Random Number Generation Revisited

- Remember, the library <cstdlib> contains a function for generating random numbers
- The statement used to produce integers in the range 0 - 5 is
  - `int x = rand() % 6;`
- To simulate the role of a dice we would use the statement
  - `int x = 1 + rand() % 6`

27.2 Random Number Generation

- Write a program that will simulate the roll of a dice 6000 times and show the frequency in which each side appeared

Face	Frequency
1	1003
2	1017
3	983
4	994
5	1004
6	999