cisc1110 fall 2010 lecture VIII.1

- *linear array operations*
- traverse
- min
- max
- sum
- search
- count

linear array traversal

- visit each element in the array, in the order of the array index

```
for ( int i = 0; i < MAX; i++ ) {
  cout << "A[i] = " << A[i] << endl;
}
```


linear algorithm to find smallest value in array

- visit each element in the array, in the order of the array index, and test for smallest value

```
smallest = A[0];
for ( int i = 1; i < MAX; i++ ) {
  if ( A[i] < smallest ) {
    smallest = A[i];
  }
}
```


linear algorithm to find largest value in array

- visit each element in the array, in the order of the array index, and test for largest value

```
largest = A[0];
for ( int i = 1; i < MAX; i++ ) {
  if ( A[i] > largest ) {
    largest = A[i];
  }
}
```

linear algorithm to sum elements in an array

- visit each element in the array, in the order of the array index, and add their values together

```
sum = 0;
for ( int i = 1; i < MAX; i++ ) {
  sum += A[i];
}
```


linear algorithm to find an element in an array

- visit each element in the array, in the order of the array index, and look for the first element that matches a specified value

```
found = false;
i = 0;
while (( ! found ) && ( i < MAX )) {
  if ( A[i] == key ) {
    found = true;
  }
  else {
    i++;
  }
}
if ( ! found ) {
  i = -1;
}
// "i" is index of first array element whose value matches "key"; OR
// "i" is -1, indicating that there are no elements that match
```


linear algorithm to count the number of "key"s in an array

- visit each element in the array, in the order of the array index, and count the number of elements that match a specified value

```
count = 0;
for ( i = 0; i < MAX; i++ ) {
  if ( A[i] == key ) {
    count++;
  }
}
```