Lab 6

CIS FIFTEEEN HUNDRED

Agenda

- General Announcements
- Debug
- One-Dimensional Arrays and Functions
- Program 2
- Program 3

Static Arrays

- Static arrays are defined at compile time
- Indices start at 0 and go up from there
- They can be declared and initialized:

```
int exampleArray[10]; //not initialized to anything
int anotherArray[10] {8,2,5,4,7,11,24,102,32,0};  //
```

  - You can also initialize a declared array using a for loop!
- Array indices are accessed by calling the variable with an index
  - int grabValue = anotherArray[3]; //will assign 4 to grabValue

Program 2

Write a program that defines and populates an array of 10 integers. It then prints the entire array (of integers) on a single line separated by spaces, followed by (on a second line) only as many elements as it takes to reach a sum greater than 50. For example if the array elements are 10 23 3 20 8 9 15 4 16 9, the second line would print 10 23 3 20 and stop printing because the sum of these four numbers is greater than 50. The program then prints the rest of the elements on a third line (e.g. 8 9 15 4 16 9).

Program 3

- Write a program that uses a function called processArray that has the following prototype:
- **int processArray(int toProcess[]);**
- This function takes as input an array of 10 integers from main and prints out the entire array of integers on a single line separated by spaces, followed by (on a second line) only as many elements as it takes to reach a sum greater than 50. For example if the array elements are 10 23 3 20 8 9 15 4 16 9, the second line would print 10 23 3 20 and stop printing because the sum of these four numbers is greater than 50. It also returns to main, the index of the element where it stopped printing (e.g. 3). Main then prints the rest of the elements on a third line (e.g. main prints: 8 9 15 4 16 9).
- Use the given prototype. Changing it will result in penalty.

- Main prompts the user for 10 integers, stores them in an array and then calls the
processArray function. It also prints the final output according to question specifications.

- Sample Input / Output:
- *Enter 10 integers: 10 23 3 20 8 9 15 4 16 9*
- *Printing in function:*
- *10 23 3 20 8 9 15 4 16 9*
- *10 23 3 20*

- *Printing in main starting from subscript 4:*

8 9 15 4 16 9