Week 14: Problems

CS 5301
Fall 2014
Jill Seaman

Problem solving

• Understand the problem description
• Generate a hypothetical solution
  - Think in terms of steps computer must perform
• Encode the solution (write the C++ code)
• Check your work
  - Trace your code, go through it step by step, carry out the instructions to see if they will solve the problem (you must be able to read code)
  - Look for errors in your solution (incorrect syntax, missing or misplaced {}, undefined variables, etc).

Practice Problem #1

• Problem: write a function that will return true if all of the elements in an array are equal to 0.
• How to think about this problem:
  - ALL of them must be 0 to be true. I have to look at ALL of them before I can return true.
  - If any one of them is not 0, it is false. I need ONE bad example to return false.

Practice Problem #2

• Write a function that computes the maximum value in an integer array a[] of size N.
Practice Problems #3

• Write a function RemoveFirst() that removes the first occurrence of a given value x from an array a[ ] of size N. It is not known whether the value actually occurs in the array. For example, if a = {2,4,5,6,4,7,2,3,4,2} then RemoveFirst( a , 4 ) produces a = {2,5,6,4,7,2,3,4,2}

The interface for the function is:

```c
void RemoveFirst( int a[], int & N, int x )
```

//Removes first x from array a[], decrements // N if x is removed

Practice Problems #4

• Write a **recursive** function SumUp() that returns the sum of the values in a singly linked list. For example the SumUp (L) applied to the list L: 3,5,4,2,5,7 returns 26. Assume the declarations:

```c
struct node {
    int data
    node *link;
};
int SumUp( node* L );
// returns the sum of the values in L
```

Practice Problems #5

• Write a function **mode** that returns the most commonly occurring element in an array of ints. For example mode (L) applied to the array L: 3,5,4,2,5,7 returns 5. If L = {1,2,3,3,4,3,5,2} it returns 3. You may assume there is only one mode in the array.