Notes From Class

Wednesday, July 25, 2018  10:00 AM

Project 3 Scores Have Been Released

Throwing Out The Code That Did Not Build, The Average Was 84.657 With Half The Scores 92 Or Better And 14 Perfect Scores Of 100! Nice Job!

Midterms Have Been Returned

The Average Was 83.39 But Half The Scores Were 85 Or Better! Nice Job!
do
{
    if (l == 12)
        break;
    if (j != 1)    continue;
    l = l + 1;
    x = x - l;
    cout << " x " << x << end;
} while (x < 5);

bool firstTime = true;
while( x < 5 || firstTime )
{
    firstTime = false;
    if (l == 12)
        break;
    if (j != 1)    continue;
    l = l + 1;
    x = x - l;
    cout << " x " << x << end;
}

    // "unravel" the loop...
    if (l == 12)
        break;
    if (j != 1)    continue;
    l = l + 1;
    x = x - l;
    cout << " x " << x << end;

    while( x < 5 )
{ 
    if (I == 12) 
        break;
    if (j != 1)   continue;
    I = I + 1;
    x = x - I;
    cout << " x " << x << end;
}

Project 4 Is Coming Due Next Wednesday
- These Are Functions With Array Parameters And Array Arguments
- You Need To Supply A Main That Builds But I Don't Care What It Does. I Won't Be Calling It
- Don't Walk Off The Edge Of The Array!
- Remember, Arrays Appear To Pass By Reference Automagically
- Due August 1st At 9 PM
- Based On Arrays Of String
- Please See The FAQ And Read The Specification Carefully
- As With Project 4, The Work Product Is The Set Of Requested Functions
  ○ You Can Write Other Functions Of Your Own
  ○ Your main( ) Function Will Be Thrown Away, As I Will
Replace It And Call My Own

- My Suggestion
  - Write `assert`'s Before You Write Your Code
    This Helps To Prove You Understand What Is Being Asked Of Your Code
  - Take Small Steps To The Goal
    Don't: Write All Functions, Save, Build, Submit
  - Uncomment Your `assert`'s One-By-One As You Get Your Code Closer And Closer To The Truth
  - Recall This Code To Loop Through A String

```cpp
string s;
size_t length = s.size();
size_t length = s.length();
// loop to walk the entire string...
for ( size_t i = 0; i < s.size(); i++ )
{
    char c = s[ i ];
    cout << c << endl;
}
```

Strings Definitely Do Compare....

- Think of the dictionary
- Not based on the length, it's based on the dictionary sort
- BOLDING is the value that is least
- How Would The Dictionary Sort: "abc" "a" < > == <= => !=
- "Lexicographically" ---> according to the dictionary - and your collating sequence
- "AAAA" "aaaa" < **bolded** item is what sorts first
- "AAA" "aaa" <
- "abc" "ab" <
- "Sad" "happy" <
- "sad" "happy" <

Int array[ 55 ] = { 0, 0, 0, 0,0 };    /// dump 0
String a[ 55 ] = { "asfd", "asfd" };
int locateMinimum( const string array[], int n )


Assert( locateMinimum( a, 0 ) == -1 );  // and don't access any part of the array
Assert( locateMinimum( a, -100 ) == -1 );  // don't access any part of the array
Assert( locateMinimum( a, 5 ) == 2 );
Assert( locateMinimum( a, 4 ) == 2 );
Assert( locateMinimum( a, 3 ) == 2 );
Assert( locateMinimum( a, 2 ) == 1 );
Assert( locateMinimum( a, 1 ) == 1 );

I'd Like To Move Forward With Lecture 7 : Classes And Structures
We'll Come Back To Lecture 6 On Monday