Bryce Jensen
CS-405-10914
1-3 Activity: Numeric Overflow Coding
9-6-2024

Here is a summary of the process being used in the program:

Overflow Testing

This function runs `test_overflow()` for various C++ primitive types (char, int, float, etc.). Each call to `test_overflow()` calculates the result of adding a large increment multiple times to a starting value of 0. The `add_numbers()` is responsible for performing the addition step by step. It detects if an overflow will occur before each addition and if detected, it returns a sentinel value (`std::numeric_limits<T>::max()`) to signal that an overflow occurred. The result is printed for the case before and after the overflow.

Underflow Testing

This function runs `test_underflow()` on the same types. Each call to `test_underflow()` calculates the result of subtracting a large decrement multiple times from a starting value. The `subtract_numbers()` performs the subtraction step by step and checks if an underflow will occur before each subtraction. If detected, it prevents the underflow by returning the sentinel value (`std::numeric_limits<T>::min()`). The result is printed for the case right before and after the underflow.

Both the `add_numbers` and `subtract_numbers` functions use a sentinel value (the maximum or minimum possible values for the type) to detect and prevent numeric overflow and underflow. These sentinel values signal a calculation failure, which is then communicated to the user through the console.

I have something weird happening with the underflow detection. The char, short, int, long, and __int64 types are not underflowing as they should be. It might have something to do with rounding, although I don't know enough about what is happening to determine why that might be the case.

Screenshots of the console are included below:

Bryce Jensen
CS-405-10914
1-3 Activity: Numeric Overflow Coding
9-6-2024

Microsoft Visual Studio Debu

```
Starting Numeric Underflow / Overflow Tests!

***********************************************
*** Running Overflow Tests ***
***********************************************
Overflow Test of Type = char
        Adding Numbers Without Overflow (0, 25, 5) = 125
        Adding Numbers With Overflow (0, 25, 6) = Overflow detected!
127
Overflow Test of Type = wchar_t
        Adding Numbers Without Overflow (0, 13107, 5) = 65535
        Adding Numbers With Overflow (0, 13107, 6) = Overflow detected!
65535
Overflow Test of Type = short
        Adding Numbers Without Overflow (0, 6553, 5) = 32765
        Adding Numbers With Overflow (0, 6553, 6) = Overflow detected!
32767
Overflow Test of Type = int
        Adding Numbers Without Overflow (0, 429496729, 5) = 2147483645
        Adding Numbers With Overflow (0, 429496729, 6) = Overflow detected!
2147483647
Overflow Test of Type = long
        Adding Numbers Without Overflow (0, 429496729, 5) = 2147483645
        Adding Numbers With Overflow (0, 429496729, 6) = Overflow detected!
2147483647
Overflow Test of Type = __int64
        Adding Numbers Without Overflow (0, 1844674407370955161, 5) = 9223372036854775805
        Adding Numbers With Overflow (0, 1844674407370955161, 6) = Overflow detected!
9223372036854775807
Overflow Test of Type = unsigned char
        Adding Numbers Without Overflow (0, 51, 5) = 255
        Adding Numbers With Overflow (0, 51, 6) = Overflow detected!
255
Overflow Test of Type = unsigned short
        Adding Numbers Without Overflow (0, 13107, 5) = 65535
        Adding Numbers With Overflow (0, 13107, 6) = Overflow detected!
65535
Overflow Test of Type = unsigned int
        Adding Numbers Without Overflow (0, 858993459, 5) = 4294967295
        Adding Numbers With Overflow (0, 858993459, 6) = Overflow detected!
4294967295
Overflow Test of Type = unsigned long
        Adding Numbers Without Overflow (0, 858993459, 5) = 4294967295
        Adding Numbers With Overflow (0, 858993459, 6) = Overflow detected!
4294967295
Overflow Test of Type = unsigned __int64
        Adding Numbers Without Overflow (0, 3689348814741910323, 5) = 18446744073709551615
        Adding Numbers With Overflow (0, 3689348814741910323, 6) = Overflow detected!
18446744073709551615
Overflow Test of Type = float
        Adding Numbers Without Overflow (0, 6.80565e+37, 5) = 3.40282e+38
        Adding Numbers With Overflow (0, 6.80565e+37, 6) = Overflow detected!
3.40282e+38
Overflow Test of Type = double
        Adding Numbers Without Overflow (0, 3.59539e+307, 5) = 1.79769e+308
        Adding Numbers With Overflow (0, 3.59539e+307, 6) = Overflow detected!
1.79769e+308
Overflow Test of Type = long double
        Adding Numbers Without Overflow (0, 3.59539e+307, 5) = 1.79769e+308
        Adding Numbers With Overflow (0, 3.59539e+307, 6) = Overflow detected!
1.79769e+308
```

Bryce Jensen
CS-405-10914
1-3 Activity: Numeric Overflow Coding
9-6-2024

Microsoft Visual Studio Debu

```
************************************************
*** Running Underflow Tests ***
************************************************
Underflow Test of Type = char
        Subtracting Numbers Without Overflow (127, 25, 5) = 2
        Subtracting Numbers With Overflow (127, 25, 6) = -23
Underflow Test of Type = wchar_t
        Subtracting Numbers Without Overflow (65535, 13107, 5) = 0
        Subtracting Numbers With Overflow (65535, 13107, 6) = Underflow detected!
0
Underflow Test of Type = short
        Subtracting Numbers Without Overflow (32767, 6553, 5) = 2
        Subtracting Numbers With Overflow (32767, 6553, 6) = -6551
Underflow Test of Type = int
        Subtracting Numbers Without Overflow (2147483647, 429496729, 5) = 2
        Subtracting Numbers With Overflow (2147483647, 429496729, 6) = -429496727
Underflow Test of Type = long
        Subtracting Numbers Without Overflow (2147483647, 429496729, 5) = 2
        Subtracting Numbers With Overflow (2147483647, 429496729, 6) = -429496727
Underflow Test of Type = __int64
        Subtracting Numbers Without Overflow (9223372036854775807, 1844674407370955161, 5) = 2
        Subtracting Numbers With Overflow (9223372036854775807, 1844674407370955161, 6) = -1844674407370955159
Underflow Test of Type = unsigned char
        Subtracting Numbers Without Overflow (255, 51, 5) = 0
        Subtracting Numbers With Overflow (255, 51, 6) = Underflow detected!
0
Underflow Test of Type = unsigned short
        Subtracting Numbers Without Overflow (65535, 13107, 5) = 0
        Subtracting Numbers With Overflow (65535, 13107, 6) = Underflow detected!
0
Underflow Test of Type = unsigned int
        Subtracting Numbers Without Overflow (4294967295, 858993459, 5) = 0
        Subtracting Numbers With Overflow (4294967295, 858993459, 6) = Underflow detected!
0
Underflow Test of Type = unsigned long
        Subtracting Numbers Without Overflow (4294967295, 858993459, 5) = 0
        Subtracting Numbers With Overflow (4294967295, 858993459, 6) = Underflow detected!
0
Underflow Test of Type = unsigned __int64
        Subtracting Numbers Without Overflow (18446744073709551615, 3689348814741910323, 5) = 0
        Subtracting Numbers With Overflow (18446744073709551615, 3689348814741910323, 6) = Underflow detected!
0
Underflow Test of Type = float
        Subtracting Numbers Without Overflow (3.40282e+38, 6.80565e+37, 5) = 0
        Subtracting Numbers With Overflow (3.40282e+38, 6.80565e+37, 6) = Underflow detected!
1.17549e-38
Underflow Test of Type = double
        Subtracting Numbers Without Overflow (1.79769e+308, 3.59539e+307, 5) = 9.9792e+291
        Subtracting Numbers With Overflow (1.79769e+308, 3.59539e+307, 6) = Underflow detected!
2.22507e-308
Underflow Test of Type = long double
        Subtracting Numbers Without Overflow (1.79769e+308, 3.59539e+307, 5) = 9.9792e+291
        Subtracting Numbers With Overflow (1.79769e+308, 3.59539e+307, 6) = Underflow detected!
2.22507e-308

All Numeric Underflow / Overflow Tests Complete!
```