3N+1 - Prog 6 (20 points)
CECS 325-01 System Programming with C++
Spring 2023
Due: April 17, 2023

YouTube video: https://youtu.be/094y1Z2wpJg
Wikipedia link: https://en.wikipedia.org/wiki/Collatz_conjecture

The **Collatz conjecture** is one of the most famous unsolved problems in mathematics. The conjecture asks whether repeating two simple arithmetic operations will eventually transform every positive integer into 1. It concerns sequences of integers in which each term is obtained from the previous term as follows:

If the previous term is even,
the next term is one half of the previous term.
If the previous term is odd,
the next term is 3 times the previous term plus 1.

The conjecture is that these sequences always reach 1, no matter which positive integer is chosen to start the sequence.

It is named after mathematician Lothar Collatz, who introduced the idea in 1937.

It is also known as the **3*n* + 1 problem**.

As of October 2018 the largest number verified for Collatz conjuncture is $2^{100000} - 1$.

Your task is to write a program that we can use to explore the 3n+1 problem.

Program requirements:

- Your program source code will be called threeN.cpp
- The executable will be called threeN
- You must have a recursive function in your program to calculate 3n+1.
- If you run threeN with no parameters, the program will ask you for a starting number. The user will type in a positive integer at that prompt and the program will calculate and print the 3n+1 results.
- If you run threeN with integer parameters on the command line, the program will calculate and display the 3n+1 results for each number.
- If the program experiences integer overflow, it will throw an exception in the recursive function. The exception will be caught in the main function and the program will continue processing the next input number – it will not crash
- You also need to print out the steps of each starting number as well as these five 3n+1 stats as shown in the examples below: **`start, steps, max, odds, evens.`**

Here are the expected results for various inputs:

```
$ threeN                                                  // program ran with no parameters
Enter a 3n+1 candidate number:12          // program asks for a start number. I type 12
->(12)->(6)->(3)->(10)->(5)->(16)->(8)->(4)->(2)->(1) // the steps are listed here
        start:12                                          // next 5 lines show stats
        steps:9
        max:16
        odds:2
        evens:7
```

```
$ threeN 12 13 14 8976543 43          // program runs with 5 command line inputs
Solving 3n+1 - starting value:12
->(12)->(6)->(3)->(10)->(5)->(16)->(8)->(4)->(2)->(1)
        start:12
        steps:9
        max:16
        odds:2
        evens:7

Solving 3n+1 - starting value:13    //auto running with second command line
value
->(13)->(40)->(20)->(10)->(5)->(16)->(8)->(4)->(2)->(1)
        start:13
        steps:9
        max:40
        odds:2
        evens:7

Solving 3n+1 - starting value:14
->(14)->(7)->(22)->(11)->(34)->(17)->(52)->(26)->(13)->(40)->(20)->(10)->(5)-
>(16)->(8)->(4)->(2)->(1)
        start:14
        steps:17
        max:52
        odds:5
        evens:12

Solving 3n+1 - starting value:8976543
->(8976543)->(26929630)->(13464815)->(40394446)->(20197223)->(60591670)-
>(30295835)->(90887506)->(45443753)->(136331260)->(68165630)->(34082815)-
>(102248446)->(51124223)->(153372670)->(76686335)->(230059006)->(115029503)-
>(345088510)->(172544255)->(517632766)->(258816383)->(776449150)->(388224575)-
>(1164673726)->(582336863)->(1747010590)->(873505295)->(###overflow###)

overflow detected for n:873505295
3n + 1:-1674451410
something broke dude
overflow

Solving 3n+1 - starting value:43// previous error does not affect next run
->(43)->(130)->(65)->(196)->(98)->(49)->(148)->(74)->(37)->(112)->(56)->(28)-
>(14)->(7)->(22)->(11)->(34)->(17)->(52)->(26)->(13)->(40)->(20)->(10)->(5)-
>(16)->(8)->(4)->(2)->(1)
        start:43
        steps:29
        max:196
        odds:9
        evens:20
```

What to submit:

1) threeN.cpp
2) Screenshot of results when command line arguments are 12 8976543 42

Learning Objectives:

- Command Line Parameters,

- integer overflow detection,
- exception handling, try/catch blocks.