Homework 1A Report

2021315385
이건 / Gun Daniel Lee

Problem explanation

- Based on the Collatz Conjecture

- Given an integer number n, repeat the following process until n = 1
  - If n is odd, n = 3n + 1
  - Else, n = n / 2

- For current problem, count the number of elements until n = 1(including n and n = 1)

Problem explanation

- Given X and Y, find and print the maximum number of elements accessed by a number between X and Y, inclusive

- 1 <= X <= Y <= 100,000

Solution explanation

- Personal Observation:
  - Multiples of 3 and prime numbers have a higher chance of accessing more elements

- However, this pattern is inconsistent and very dependent on the X and Y range

Solution explanation

- Brute Force Method:
  - Repeat the process for every number between X and Y
  - Check and update the maximum number of elements accessed
  - Print out the results

Solution explanation

- Line 8: Receive input
- Line 10: For loop to repeat process with all values from X to Y
- Line 15: While loop to do the Collatz Conjecture

❖ N: temporary variable to store the value to be tested

❖ count: variable to store the elements accessed

❖ Max: maximum number of elements accessed

```c
#include <stdio.h>

int main()
{
  int x, y, i;
  int max = 0;

  scanf("%d %d", &x, &y);

  for(i = x; i <= y; i++)
  {
    int n = i;
    int count = 0;

    while(n != 1)
    {
      if(n % 2 != 0)
      {
        n = 3 * n + 1;
      }

```

Solution explanation

- Line 17 ~ 25: If condition
  - Line 17 ~ 20 (n is odd): n -> 3n + 1
  - Line 22 ~ 25 (n is even): n -> n / 2
- Line 26 & 28: Increase count variable
  - 1 loop = 1 additional element accessed
  - Line 28: consideration of n = 1
- Line 36: print results

```
14
15          while(n != 1)
16          {
17              if(n % 2 != 0)
18              {
19                  n = 3 * n + 1;
20              }
21
22              else
23              {
24                  n /= 2;
25              }
26              count++;
27          }
28          count++;
29
30          if(count > max)
31          {
32              max = count;
33          }
34      }
35
36      printf("%d %d %d", x, y, max);
37
38      return 0;
39  }
```

Solution explanation

- Example:
  - X = 11, Y = 15
  - Current max = 0

  - N = 11: 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
    - Count = 15
    - 15 > 0
    - Max = 15

Solution explanation

- Example:
  - X = 11, Y = 15
  - Current max = 15

  - N = 12: 12 6 3 10 5 16 8 4 2 1
    - Count = 10
    - 10 < 15
    - Max = 15

Solution explanation

- Example:
  - X = 11, Y = 15
  - Current max = 15

  - N = 13: 13 40 20 10 5 16 8 4 2 1
    - Count = 10
    - 10 < 15
    - Max = 15

Solution explanation

- Example:
  - X = 11, Y = 15
  - Current max = 15

  - N = 14: 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
    - Count = 18
    - 18 > 15
    - Max = 18

Solution explanation

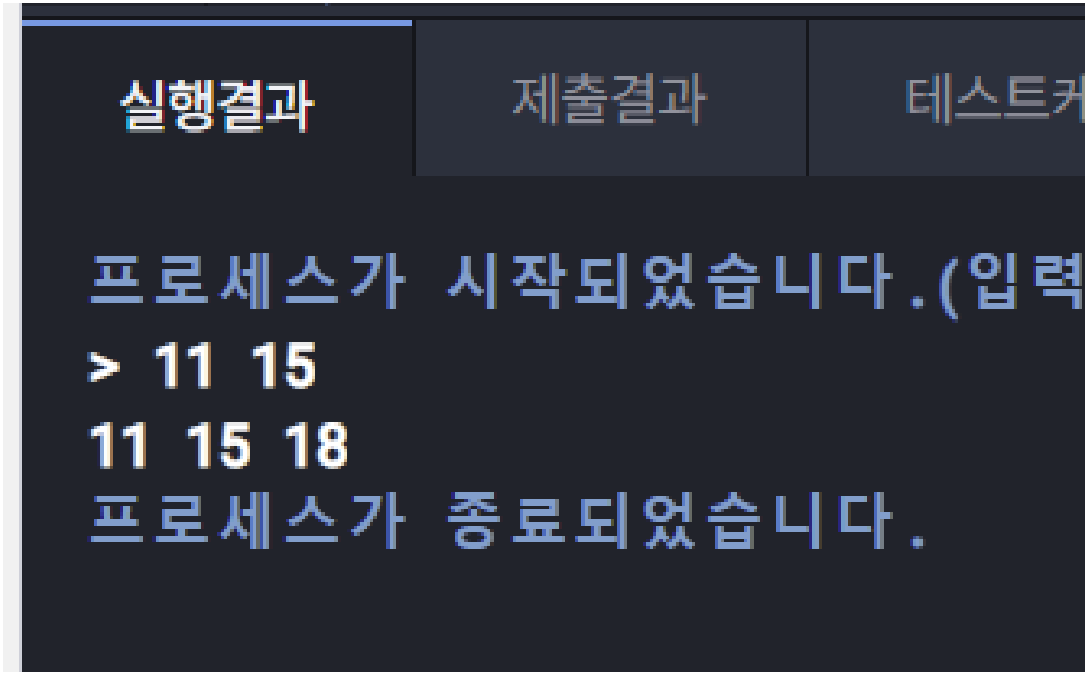
- Example:
  - X = 11, Y = 15
  - Current max = 18

  - N = 15: 15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
    - Count = 18
    - 18 = 18
    - Max = 18

Solution explanation

- Example:
  - X = 11, Y = 15
  - Current max = 18

  - Final Answer: “11 15 18”



Solution analysis

- Pros:
  - Direct approach
  - Fewer chances of errors
- Cons
  - Inefficient due to the need to apply the process to the entire range of numbers

Solution analysis

- Favorable inputs:
  - X and Y are exactly the same (X = Y)
    - Only have to check one case

Thank you!