Day 3 Date: 15-09-2024 Homework

Problem 1:

The n-th term of Fibonacci series F(n), where F(n) is a function, is calculated using the following formula -

F(n) = F(n - 1) + F(n - 2),

Where, F(1) = 1, F(2) = 1

Provided 'n' you have to find out the n-th Fibonacci Number. Handle edges cases like when 'n' = 1 or 'n' = 2 by using conditionals like if else and return what's expected.

"Indexing is start from 1"

Example :

Input: 6

Output: 8

Explanation: The number is ‘6’ so we have to find the “6th” Fibonacci number.

So by using the given formula of the Fibonacci series, we get the series:

[1, 1, 2, 3, 5, 8, 13, 21]

So the “6th” element is “8” hence we get the output.

Input Format :

The first line contains an integer ‘n’.

Output Format :

Print the n-th Fibonacci number.

Sample Input 1:

6

Sample Output 1:

8

Problem no 2.

Given an integer N, print all the prime numbers that lie in the range 2 to N (both inclusive).

Print the prime numbers in different lines.

Sample Input 1:

9

Sample Output 1:

2

3

5

7

Sample Input 2:

20

Sample Output 2:

2

3

5

7

11

13

17

19

Problem no 3.

Write a program that asks the user for a number N and a choice C. And then give them the possibility to choose between computing the sum and computing the product of all integers in the range 1 to N (both inclusive).

If C is equal to -
1, then print the sum
2, then print the product
Any other number, then print '-1' (without the quotes)

Sample Input 1:
10
1

Sample Output 1:
55

Sample Input 2:
10
2

Sample Output 2:
3628800

Sample Input 3:
10
4

Sample Output 3:
-1

Problem no 4.

Write a program to print first x terms of the series 3N + 2 which are not multiples of 4.

Input format:
Integer x

Output format:
Terms of series (separated by space)

Constraints:
1 <= x <= 1,000

Sample Input 1:
10

Sample Output 1:

5 11 14 17 23 26 29 35 38 41

Sample Input 2:
4

Sample Output 2:

5 11 14 17

Explanation:
Input is 4 and print the first 4 numbers that are not divisible by 4 and are of the form 3N + 2, where k is a non-negative integer.

Problem no 5.

Write a program to generate the reverse of a given number N. Print the corresponding reverse number.

Note: If a number has trailing zeros, then its reverse will not include them.

For e.g., reverse of 10400 will be 401 instead of 00401.

Input format:

Integer N

Output format:

Corresponding reverse number

Constraints:

0 <= N < 10^8

Sample Input 1:

1234

Sample Output 1:

4321

Sample Input 2:

1980

Sample Output 2:

891

Problem no 6.

Given a binary number as an integer N, convert it into decimal and print.

Input format:

An integer N in the Binary Format

Output format:

Corresponding Decimal number (as integer)

Constraints:

0 <= N <= 10^9

Sample Input 1:

1100

Sample Output 1:

12

Sample Input 2:

111

Sample Output 2:

7

Problem no 7.

Given a decimal number (integer N), convert it into binary and print.

Input format:

Integer N

Output format:

Corresponding Binary number (long)

Constraints:

0 <= N <= 10^5

Sample Input 1:

12

Sample Output 1:

1100

Sample Input 2:

7

Sample Output 2:

111

Sample Input 2:

0

Sample Output 2:

0

Problem no 8.

Given a number N, find its square root. You need to find and print only the integral part of square root of N.

For eg. if number given is 18, answer is 4.

Input format :

Integer N

Output Format :

Square root of N (integer part only)

Constraints :

0 <= N <= 10^8

Sample Input 1 :

10

Sample Output 1 :

3

Sample Input 2 :

4

Sample Output 2 :

2

Problem no 9.

You are given S, a sequence of n integers i.e. S = s1, s2, ..., sn. Compute if it is possible to split S into two parts : s1, s2, ..., si and si+1, si+2, ..., sn (0 <= i <= n) in such a way that the first part is strictly decreasing while the second is strictly increasing one.

Note : We say that x is strictly larger than y when x > y. So, a strictly increasing sequence can be 1 4 8. However, 1 4 4 is NOT a strictly increasing sequence.

That is, in the sequence if numbers are decreasing, they can start increasing at one point. Thereafter, they cannot decrease at any point further.

Sequence made up of only increasing numbers or only decreasing numbers is a valid sequence. So, in both the cases, print true.

You just need to print true/false. No need to split the sequence.

Sample Input 1 :

5

9

8

4

5

6

Sample Output 1 :

true

Sample Input 2 :

3

1

2

3

Sample Output 2 :

true

Sample Input 3 :

3

8

7

7

Sample Output 3 :

false

Explanation for Sample Format 3 :

8 7 7 is not strictly decreasing, so output is false.

Sample Input 4 :

6

8

7

6

5

8

2

Sample Output 4 :

false

Explanation for Sample Input 4 :

The series is :

8 7 6 5 8 2

It is strictly decreasing first (8 7 6 5). Then it's strictly increasing (5 8). But then it starts strictly decreasing again (8 2). Therefore, the output for this test case is 'false'

Problem no 10.

Create a function that determines whether a given number N belongs to the Fibonacci sequence. If N is found in the Fibonacci sequence, the function should return true; otherwise, it should return false.

Sample Input 1 :

5

Sample Output 1 :

true

Explanation :

Fibonacci sequence begins 0, 1, 1, 2, 3, 5, ... and so on. Since 5 appears in the sequence.

Sample Input 2 :

14

Sample Output 2 :

false