**Fundamental Questions( Very Important)**

Q1**. All Prime Numbers**

**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.**

**Input Format :**

Integer N

**Output Format :**

Prime numbers in different lines

**Constraints :**

1 <= N <= 100

**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

Q2. **Nth Fibonacci Number**

**Nth 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 Nth Fibonacci Number.**

**Input Format :**

The first line of each test case contains a real number ‘N’.

**Output Format :**

For each test case, return its equivalent Fibonacci number.

**Constraints:**

1 <= N <= 10000

Where ‘N’ represents the number for which we have to find its equivalent Fibonacci number.

Time Limit: 1 second

**Sample Input 1:**

6

**Sample Output 1:**

8

**Explanation**

Now the number is ‘6’ so we have to find the “6th” Fibonacci number

So by using the property of the Fibonacci series i.e

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

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

Q3. **Trailing Zeros in Factorial**

**Given an integer n, find the number of trailing 0s in its factorial. Do this without actually finding the n!, as the integer given can be large and its factorial can exceed the integer limits.**

**Input format :**

Integer n

**Output Format :**

Count of trailing 0s

**Constraints :**

**1 <= n <= 10^11**

**Sample Input**

50

**Sample Output**

12

Q4. **Count Characters**

**Write a program to count and print the total number of characters (lowercase english alphabets only), digits (0 to 9) and white spaces (single space, tab i.e. '\t' and newline i.e. '\n') entered till '$'.**

**That is, input will be a stream of characters and you need to consider all the characters which are entered till '$'.**

**Print count of characters, count of digits and count of white spaces respectively (separated by space).**

**Input Format :**

A stream of characters terminated by '$'

**Output Format :**

3 integers i.e. count\_of\_characters count\_of\_digits count\_of\_whitespaces (separated by space)

**Sample Input :**

abc def4 5$

**Sample Output :**

6 2 2

**Sample Output Explanation :**

**Number of characters : 6 (a, b, c, d, e, f)**

**Number of digits : 2 (4, 5)**

**Number of white spaces : 2 (one space after abc and one newline after 4)**

Q5. **Sum or Product**

**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)

**Input format :**

Line 1 : Integer N

Line 2 : Choice C

**Output Format :**

Sum or product according to user's choice

**Constraints :**

1 <= N <= 12

**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

Q6. **Terms of AP**

**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

Q7. **Square Root (Integral)**

**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

Q8. **Reverse of a number**

**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

Q9. **Decimal to Binary**

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

**The binary number should be in the form of an integer.**

**Note: The given input number could be large, so the corresponding binary number can exceed the integer range. So you may want to take the answer as long for CPP and Java.**

**Note for C++ coders: Do not use the inbuilt implementation of "pow" function. The implementation of that function is done for 'double's and it may fail when used for 'int's, 'long's, or 'long long's. Implement your own "pow" function to work for non-float data types.**

**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

Q10. **Binary to decimal**

**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

**CHOCOLATE QUESTION**

**Check Number sequence**

**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.**

**Input format :**

Line 1 : Integer 'n'

Line 2 and Onwards : 'n' integers on 'n' lines(single integer on each line)

**Output Format :**

"true" or "false" (without quotes)

**Constraints :**

1 <= n <= 10^7

**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'