Write a program to accept number from user and

calculate the sum of all number from 1 to a given number.

**Input Format:**

Line 1: <Integer number>

**Output Format:**

<Integer Number – Sum value>

**Input:**

10

**Output:**

55

Ron has been given 10 numbers which includes both positive and negative

values by his teacher. His task is to tell the total of positive and negative

numbers separately. Our task is to help him in this task

**Sample Input:**

10

50

-30

15

-45

-96

5

78

-99

253

**Expected Output:**

Sum of Positive Numbers = 411

Sum of Negative Numbers =-270

Write a program to check whether the given number is an

Armstrong number or not.

Armstrong Number:abcd... = pow(a,n) + pow(b,n) + pow(c,n) + pow(d,n) + ....

where n represents the number of digits

**Input:**

153

**Output:**

True

153 is an Armstrong number. 1\*1\*1 + 5\*5\*5 + 3\*3\*3 = 153

**Input:**

120

**Output:**

False

120 is not an Armstrong number. 1\*1\*1 + 2\*2\*2 + 0\*0\*0 = 9

**Input:**

1253

**Output:**

False

1253 is not an Armstrong Number

1\*1\*1\*1 + 2\*2\*2\*2 + 5\*5\*5\*5 + 3\*3\*3\*3 = 723

**Input:**

1634

**Output:**

True

1634 is an Armstrong Number

1\*1\*1\*1 + 6\*6\*6\*6 + 3\*3\*3\*3 + 4\*4\*4\*4 = 1634

Write a program to display all the Prime Numbers

between a given range of numbers.

**Note**: A Prime Number is a whole number that cannot be made by

multiplying other whole numbers.

Example –

✓ 6 is not a Prime Number because it can be made by 2×3 = 6

✓ 37 is a Prime Number because no other whole numbers multiply

together to make it.

**Input Format:**

Line 1: <Integer Number – starting value>

Line 2: <Integer Number – ending value>

**Output Format:**

<Numbers separated by a single space>

**Input:**

25

50

**Output:**

29 31 37 41 43 47

Write a Python program to find the sum of the series

2 +22 + 222 + 2222 + .. n terms

**Input Format:**

Line 1: <Integer Number>

**Output Format:**

<Integer Number – sum of the series>

**Input:**

5

**Output:**

24690

Write a program to find whether a given number is a strong number or not. Note:Strong number is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since, 1! + 4! + 5! = 145 Input Format Input consist of an Integer Output Format Output consist of String

**Test case 1**

**Input**

145

**Output**

Yes

Write a program to generate the following series 7, 5, 8, 6, 9….

**Case 1**

**Input (stdin)**

5

**Output (stdout)**

7 5 8 6 9

**Case 2**

**Input (stdin)**

10

**Output (stdout)**

7 5 8 6 9 7 10 8 11 9

Write a program to generate the following series 1, 2, 3, 5, 8….

**Case 1**

**Input (stdin)**

5

**Output (stdout)**

1 2 3 5 8

**Case 2**

**Input (stdin)**

10

**Output (stdout)**

1 2 3 5 8 13 21 34 55 89

Write a program to generate the following series 6, 9, 14, 21, 30….

**Case 1**

**Input (stdin)**

5

**Output (stdout)**

6 9 14 21 30

**Case 2**

**Input (stdin)**

29

**Output (stdout)**

6 9 14 21 30 41 54 69 86 105 126 149 174 201 230 261 294 329 366 405 446 489 534 581 630 681 734 789 846

**Harshard Number**

Write a program to find whether the given number is a Harshad number or not. Note that Harshard number is an integer that is divisible by the sum of its digits.

**INPUT & OUTPUT FORMAT:**

Input consists of 1 integer. If the given number is a Harshad Number, display "Harshad Number" or display "Not Harshad Number".

**Case 1**

**Input (stdin)**

1729

**Output (stdout)**

Harshard Number

**Case 2**

**Input (stdin)**

64

**Output (stdout)**

Not Harshard Number

**Abundant Number**

Write a C program to find whether the given number is an Abundant number or not. Note: An abundant number is a number for which the sum of its proper divisors is greater than the number itself. For example, integer 12 is an abundant number. The divisors of 12 are 1, 2, 3, 4 and 6. The sum of divisors of 12 is 16. As 12 < 16, it is an abundant number.

**INPUT & OUTPUT FORMAT:**

Input consists of 1 integer. If it is an Abundant number display “Abundant Number” or display “Not Abundant Number”.

**Case 1**

**Input (stdin)**

6

**Output (stdout)**

Not Abundant Number

**Case 2**

**Input (stdin)**

14770

**Output (stdout)**

Abundant Number

**Wipro 2022**

A company wishes to transmit data to another server. The data consists of numbers only. To secure the data during transmission, they plan to reverse the data first. Write an algorithm to reverse the data.

**Input**The input consists of an integer data, representing the data to be transmitted.

**Output**  
Print an integer representing the given data in reverse form.

**Constraints**NA

**Example**

**Input:**5783789

**Output:**9873875

**Explanation:**On reversing the given value, the output is 9873875.

**Wipro 2022**

An e-commerce company plans to give their customers a discount for the New Years holiday. The discount will be calculated on the basis of the bill amount of the order placed. The discount amount is the sum of all the odd digits on the customer’s total bill amount. If no odd digit is present in the bill amount, then the discount will be zero. Write an algorithm to find a discount for the given total bill amount.

**Input**  
The input consists of an integer bill amount, representing the customer’s total bill amount.

**Output**Print an integer representing the discount for the given total bill amount.

**Constraints**0 <billAmount ≤ 109

**Example**

**Input:**2514795

**Output:**27

**Explanation:**Odd digits in the given number 2514795 are 5, 1, 7, 9, 5. The sum of these odd digits is 27. So, the output is 27.

**Wipro 2022**

An e-commerce website wishes to find the lucky customer who will be eligible for full value cash back. For this purpose, a number N is fed to the system. It will return another number that is calculated by an algorithm. In the algorithm, a sequence is generated, in which each number is the sum of the two proceeding numbers. Initially the sequence will have two 1’s in it. The system will return the Nth number from the generated sequence which is treated as the order ID. The lucky customer will be the one who has placed that order. Write an algorithm to help the website find the lucky customer.

**Input**The input consists of an integer token, representing the number fed to the system (N).

**Output**Print an integer representing the order ID of the lucky customer.

**Constraints**NA

**Example**

**Input:**8

**Output:**21

**Explanation:**The sequence generated by the system will be 1,1,2,3,5,8,13,21. The 8th number in the sequence is 21. The lucky customer is the one who has placed the order with order ID 21.

**Wipro 2022**

You are playing an online game. In the game, a numbers is displayed on the screen. In order to win the game, you have to Count the trailing zeros in the factorial value of the given number. Write an algorithm to count the trailing zeros in the factorial value of the given number.

**Input**The input consists of an integer num, representing the number displayed on the screen.

**Output**Print An integer representing the count of trailing zeros in the factorial of the given numbers.  
**Note**: The factorial of the number is calculated as the product of integer numbers from 1 to num.

**Constraints**NA

**Example**

**Input:**5

**Output:**1

**Explanation:**On calculating the factorial of 5, the output is 120 (1 x2x3x4x5). There is only one trailing 0 in 120, So the output is 1.

**Wipro 2022**

An e-commerce site wishes to enhance its ordering process. They plan to implement a new scheme of OTP (One Time Password) generation for order confirmations. The OTP can be any number of digits. For OTP generation, the user will be asked to enter two random numbers. The first number entered should always be smaller than the second number. The OTP is calculated as the sum of the maximum and the minimum prime values in the range of the user-entered numbers. Write an algorithm to find the OTP.

**Input**The input consists of two space-separated integers – firstNumber and secondNumber, representing the two numbers entered by the user. Both numbers are considered in the range.

**Output**Print an integer representing sum of largest and smallest prime number in the range of given numbers.

**Constraints**-109< firstNumber <secondNumbers<109

**Example**

**Input:**  
-97 50

**Output:**50

**Explanation**The smallest and largest prime numbers within the given numbers are -97 and 47, respectively. The sum of -97 and 47 is 50. So, the output is 50.

**The Torres County Visa Center (Wipro 2022)**

The Torres county visa center generates the token number for its applicants from their application ID. The application ID is a numeric value. The token number generated in a specific form. The even digits in the applicant’s ID are replaced by the digit one greater then the even digit and the odd digits in the applicant’s ID are replaced by the digit one less than the odd digit. The numeric value thus generated represents the taken number of the applicant.

**Input:**

The input consists of an integer – applicantID, representing the application ID of the applicant.

**Output:**

Print an integer representing the token number.

**Example:**

**Input**

245567

**Output**

354476

**Explanation:**

The first digit in the application ID is ‘2’ i.e., even, it is replaced by one greater digit i.e. 3.

The second digit in the application ID is ‘4’ i.e., even, it is replaced by one greater digit i.e. 5.

The third digit in the application ID is ‘5’ i.e., odd, it is replaced by one lesser digit i.e. 4 and so on.

Hence the output is: 354476

**The cosmetic company Wipro 2022**

The cosmetic company “BeautifyMe” has decided to reward its brand promoters with a batch of cosmetics. Each batch is identified by a unique numeric barcode. Each digit of the barcode number represents a product ID of the products present in the batch. Before distributing the products to its promoters, the company wants to make sure that all products present in the batch are still fresh and safe to use. The distribution center has therefore been advised to identify the oldest product in each batch is the one with the smallest product ID. There can be multiplies of the same product in one batch.

**Input:**

Read n denoting the product ID

**Output:**

Print an integer representing the ID …..oldest product present in the batch.

**Example:**

**Input:**

345627

**Output:**

2

**Explanation:**

The smallest product ID is 2. Hence it is the oldest product.

**TCS Ninja 2022**

A supermarket maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. The task here is to design the software such that given the code of any item N the product (multiplication) of all the digits of value should be computed(price).

**Example 1:**

**Input :**

5244 -> Value of N

**Output :**160 -> Price

**Explanation:**

From the input above

Product of the digits 5,2,4,4

5\*2\*4\*4= 160

Hence, output is 160.

**Fibonacci Number**

Write a program to find whether the given number is a Fibonacci number. **INPUT & OUTPUT FORMAT:**

Input consists of 1 integer. If it is a Fibonacci number display “Fibonacci Number” or display “Not Fibonacci Number”. The output consists of 1 integer.

**Case 1**

**Input (stdin)**

3

**Output (stdout)**

Fibonacci Number

**Case 2**

**Input (stdin)**

122

**Output (stdout)**

Not Fibonacci Number

**Multiplication table**

Write a program to find the multiplication tables of a given number.

**INPUT & OUTPUT FORMAT:**

Input consists of 2 integers. First input corresponds to the number for which the multiplication tables have to be generated. Second input corresponds to the number until which the tables need to be generated.

**Case 1**

**Input (stdin)**

1

5

**Output (stdout)**

1 \* 1 = 1

1 \* 2 = 2

1 \* 3 = 3

1 \* 4 = 4

1 \* 5 = 5

**Case 2**

**Input (stdin)**

15

15

**Output (stdout)**

15 \* 1 = 15

15 \* 2 = 30

15 \* 3 = 45

15 \* 4 = 60

15 \* 5 = 75

15 \* 6 = 90

15 \* 7 = 105

15 \* 8 = 120

15 \* 9 = 135

15 \* 10 = 150

15 \* 11 = 165

15 \* 12 = 180

15 \* 13 = 195

15 \* 14 = 210

15 \* 15 = 225