

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

**Ex. No. : 3.10**

**Date:**

**Register No.: 230701358**

**Name: TARUN C**

---

## **Chinese Zodiac**

The Chinese zodiac assigns animals to years in a 12 year cycle. One 12 year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the dragon, and 1999 being another year of the hare.

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

**Program:**

```
a=eval(input())
b=a%12
if(b==0):
    print(a,"is the year of the Monkey.")
elif(b==1):
    print(a,"is the year of the Rooster.")
elif(b==2):
    print(a,"is the year of the Dog.")
```

```
elif(b==3):  
    print(a,"is the year of the Pig.")  
elif(b==4):  
    print(a,"is the year of the Rat.")  
elif(b==5):  
    print(a,"is the year of the Ox.")  
elif(b==6):  
    print(a,"is the year of the Tiger.")  
elif(b==7):  
    print(a,"is the year of the Hare.")  
elif(b==8):  
    print(a,"is the year of the Dragon.")  
elif(b==9):  
    print(a,"is the year of the Snake.")  
elif(b==10):  
    print(a,"is the year of the Horse.")  
elif(b==11):  
    print(a,"is the year of the Sheep.")
```







## **04 - Iteration Control Structures**

**For example:**

Input	Result
20	1 2 4 5 10 20



**Ex. No. : 4.1**

**Date:**

**Register No.: 230701358**

**Name: TARUN C**

---

## **Factors of a number**

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number).

**Program:**

```
s=int(input())
for i in range(1,s+1):
    if(s%i==0):
        print(i,end=' ')
```

**For example:**

<b>Input</b>	<b>Result</b>
292	1
1015	2
108	3
22	0

Ex. No. : 4.2

Date:

Register No.: 230701358

Name: TARUN C

---

## **Non Repeated Digit Count**

Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number  $\geq 1$  and  $\leq 25000$ .

Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non-repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-repeated digits in this number.

**Program:**

```
a=int(input())
digit=[]
num=str(a)
for i in num:
    count=num.count(i)
    if count==1:
        digit.append(i)
c=len(digit)
print(c)
```

Example1: if the given number N is 7, the method must return 2

Example2: if the given number N is 10, the method must return 1

**For example:**

Input	Result
7	2
10	1

**Ex. No. : 4.3**

**Date:**

**Register No.: 230701358**

**Name: TARUN C**

---

### **Prime Checking**

Write a program that finds whether the given number N is Prime or not. If the number is prime, the program should return 2 else it must return 1.

Assumption:  $2 \leq N \leq 5000$ , where N is the given number.

**Program:**

```
a=int(input())
if a%2!=0:
    print("2")
elif a==2:
    print("2")
else:
    print("1")
```

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

**Ex. No. : 4.4**

**Date:**

**Register No.: 230701358**

**Name: TARUN C**

---

## **Next Perfect Square**

Given a number N, find the next perfect square greater than N.

**Program:**

```
a=int(input())
```

```
b=int(a**0.5)
```

```
c=(b+1)**2
```

```
print(c)
```

NOTE: Fibonacci series looks like –

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

- first Fibonacci number is 0,
- second Fibonacci number is 1,
- third Fibonacci number is 1,
- fourth Fibonacci number is 2,
- fifth Fibonacci number is 3,
- sixth Fibonacci number is 5,
- seventh Fibonacci number is 8, and so on.

**For example:**

**Input:**

7

**Output**

8