

Month name to days

The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display "28 or 29 days" for February so that leap years are addressed.

Sample Input 1

February

Sample Output 1

February has 28 or 29 days in it.

Sample Input 2

March

Sample Output 2

March has 31 days in it.

Sample Input 3

April

Sample Output 3

April has 30 days in it.

For example:

Input	Result
February	February has 28 or 29 days in it.

Program:

```
odd=['January','March','May','July','August','October','December']
```

```
even=['April','June','September','November']
```

```
a=input()
```

```
if(a=='February'):
```

```
    print("February has 28 or 29 days in it.")
```

```
if a in odd:
```

```
    print(a,"has 31 days in it.")
```

```
if a in even:
```

```
    print(a,"has 30 days in it.")
```

	Input	Expected	Got	
✓	February	February has 28 or 29 days in it.	February has 28 or 29 days in it.	✓
✓	March	March has 31 days in it.	March has 31 days in it.	✓
✓	April	April has 30 days in it.	April has 30 days in it.	✓
✓	May	May has 31 days in it.	May has 31 days in it.	✓

Ex. No. : 3.10

Date: 12.04.24

Register No.: 231901018

Name: Kavin Sainath S

Admission Eligibility

Write a program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths ≥ 65

Marks in Physics ≥ 55

Marks in Chemistry ≥ 50

Or

Total in all three subjects ≥ 180

Sample Test Cases

TestCase1

Input

70

60

80

Output

The candidate is eligible

TestCase2

Input

50

80

80

Output

The candidate is eligible

TestCase3

Input

50

60

40

Output

Thecandidateisnoteligible

For example:

Input	Result
50	Thecandidateiseligible
80	
80	

Program:

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

```
b=int(input())
```

```
c=int(input())
```

```
if(a>=65 and b>=55 and c>=50):
```

```
    print("The candidate is eligible")
```

```
elif(a+b+c>=180):
```

```
    print("The candidate is eligible")
```

```
else:
```

```
    print("The candidate is not eligible")
```

	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓
✓	50 60 40	The candidate is not eligible	The candidate is not eligible	✓
✓	20 10 25	The candidate is not eligible	The candidate is not eligible	✓

04 - Iteration Control Structures

Ex. No. : 4.1

Date: 13.04.24

Register No.: 231901018

Name: Kavin Sainath S

Nth Fibonacci

Write a program to return the nth number in the fibonacci series. The value of N will be passed to the program as input.

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	Result
1	0
4	2
7	8

Program:

```
a=int(input())  
  
b=0  
  
c=1  
  
if(a==1):  
    print("0")  
elif(a==2):  
    print("1")  
else:  
    for i in range (3,a+1):  
        d=b+c  
        b=c  
        c=d  
    print(d)
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	4	2	2	✓
✓	7	8	8	✓

Ex. No. : 4.2

Date: 13.04.24

Register No.: 231901018

Name: Kavin Sainath S

Factors of a number

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

For example:

Input	Result
20	1 2 4 5 10 20

Program:

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

	Input	Expected	Got	
✓	20	1 2 4 5 10 20	1 2 4 5 10 20	✓
✓	5	1 5	1 5	✓
✓	13	1 13	1 13	✓

Ex. No. : 4.3

Date: 13.04.24

Register No.: 231901018

Name: Kavin Sainath S

Product of single digit

Given a positive integer N, check whether it can be represented as a product of single digit numbers.

InputFormat:

Single Integer input.

OutputFormat:

Output displays Yes if conditions satisfy else prints No.

Example Input:

14

Output:

Yes

Example Input:

13

Output:

No

Program:

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

```
c=0
```

```
for i in range(1,10): for j in range(1,10):
```

```
if i*j==a:
```

```
c=1
```

```
if(c==1):
```

```
print("Yes")
```

```
▼else:
```

```
print("No")
```

	Input	Expected	Got	
✓	14	Yes	Yes	✓
✓	13	No	No	✓

Unique Digit Count

Write a program to find the count of unique 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 ≥ 1 and ≤ 25000 .

For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits '2' and '9' in this number.

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

For example:

Input	Result
292	2
1015	3

Program:

```
a=input()
```

```
b=len(set(a))
```

```
print(b)
```

	Input	Expected	Got	
✓	292	2	2	✓
✓	1015	3	3	✓
✓	123	3	3	✓

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 $> = 1$ and $< = 25000$.

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 0 non-repeated digits in this number.

For example:

Input	Result
292	1
1015	2
108	3
22	0

Program:

```
a={}
```

```
for i in input:
```

```
    if i in a:a[i]+=1
```

```
    else:a[i]=1
```

```
print(sum([1 for i in a if a[i]==1]))
```

	Input	Expected	Got	
✓	292	1	1	✓
✓	1015	2	2	✓
✓	108	3	3	✓
✓	22	0	0	✓

Ex. No. : 4.6

Date: 13.04.24

Register No.: 231901018

Name: Kavin Sainath S

Next Perfect Square

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

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N .

Example Input:

10

Output:

16

Program:

```
import math
a=int(input())
b = a + 1
while b > 0 :
    m=math.sqrt(b)
    if(m==int(m)):
        print(b)
        break
    else:
        b = b + 1
```

	Input	Expected	Got	
✓	10	16	16	✓

Ex. No. : 4.7

Date: 13.04.24

Register No.: 231901018

Name: Kavın Sainath S

Sum of Series

Write a program to find the sum of the series $1+11+111+1111+\dots+n$ terms (n will be given as input from the user and sum will be the output)

Sample Test Cases

Test Case 1

Input

4

Output

1234

Explanation:

as input is 4, have to take 4 terms

$1+11+111+1111$

Test Case 2

Input

6

Output

123456

For example:

Input	Result
3	123