1. Concat Names

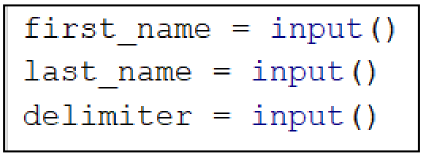
Write a program that reads two names and a delimiter. It should print the names joined by the delimiter.

Examples

Input	Output
John Smith ->	John->Smith
Jan White <->	Jan<->White
Linda Terry =>	Linda=>Terry

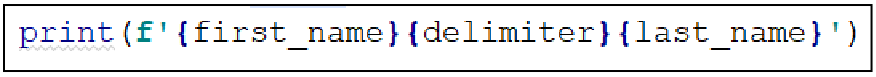
Hints

- Read the data:



```
first_name = input()
last_name = input()
delimiter = input()
```

- Print:



```
print(f'{first_name}{delimiter}{last_name}')
```

2. Convert Meters to Kilometers

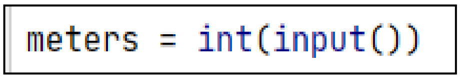
You will be given an **integer** that represents a **distance in meters**. Write a program that **converts meters** to **kilometers** formatted to the second decimal point.

Examples

Input	Output
1852	1.85
798	0.80

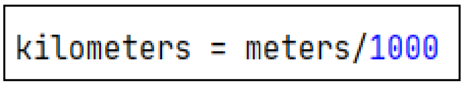
Hints

- First, we read the input number:



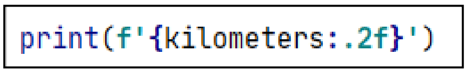
```
meters = int(input())
```

- Then, we convert it to km:



```
kilometers = meters/1000
```

- Finally, print the number formatted to the second decimal point:



```
print(f'{kilometers:.2f}')
```

3. Pounds to Dollars

Write a program that **converts British pounds** (integer) **to US** dollars formatted to the 3rd decimal point.

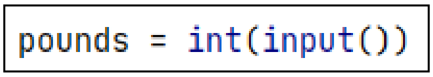
1 British Pound = 1.31 Dollars.

Examples

Input	Output
`80`	`104.800`
`39`	`51.090`

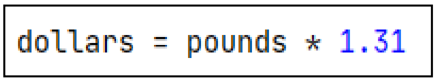
Hints

- Read the pounds:



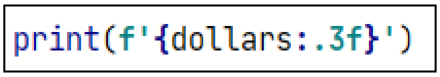
```
pounds = int(input())
```

- Convert them to dollars:



```
dollars = pounds * 1.31
```

- Finally, print the number formatted to the third decimal point:



```
print(f'{dollars:.3f}')
```

4. Centuries to Minutes

Write a program that reads an integer number of **centuries** and converts it to **years**, **days**, **hours**, and **minutes**.

Examples

Input	Output
`1`	`1 centuries = 100 years = 36524 days = 876576 hours = 52594560 minutes`
`5`	`5 centuries = 500 years = 182621 days = 4382904 hours = 262974240 minutes`

Hints

- Assume that one year has 365.2422 days on average (the Tropical year).

5. Special Numbers

Write a program that reads an integer **n**. Then, for all numbers in the range **[1, n]**, prints the number and if it is special or not (**`True`** / **`False`**). A number is **special** when the **sum of its digits is 5, 7, or 11**.

Examples

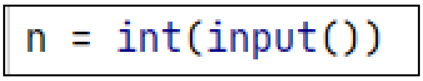
Input	Output

`15`	`1 -> False`
	`2 -> False`
	`3 -> False`
	`4 -> False`
	`5 -> True`
	`6 -> False`
	`7 -> True`

	8 -> False 9 -> False 10 -> False 11 -> False 12 -> False 13 -> False 14 -> True 15 -> False
6	1 -> False 2 -> False 3 -> False 4 -> False 5 -> True 6 -> False

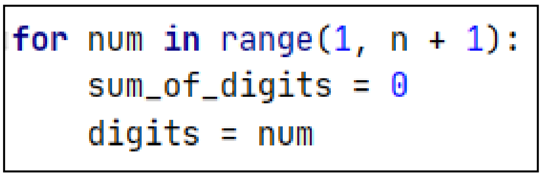
Hints

- First, we read the data:



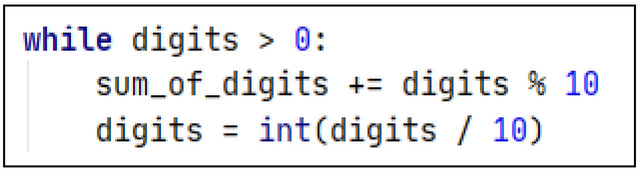
```
n = int(input())
```

- Iterate from 1 to **n** (we write **n+1** because the for loop in Python iterates from 1 to **n-1** by default):



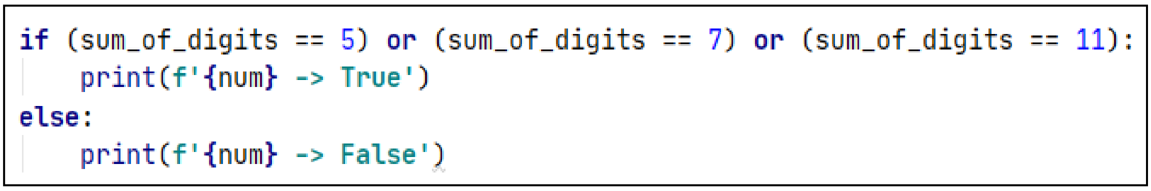
```
for num in range(1, n + 1):
    sum_of_digits = 0
    digits = num
```

- To calculate the sum of digits of given number **num**, you might repeat the following: sum the last digit (**num % 10**) and remove it (**sum = sum / 10**) until **num** reaches **0**.



```
while digits > 0:
    sum_of_digits += digits % 10
    digits = int(digits / 10)
```

- Finally, print the result:



```
if (sum_of_digits == 5) or (sum_of_digits == 7) or (sum_of_digits == 11):
    print(f'{num} -> True')
else:
    print(f'{num} -> False')
```

6. Next Happy Year

You are saying goodbye to your best friend: "***See you next happy year***". Happy Year is the year with only **distinct digits**, for example, 2018. Write a program that receives an integer number and finds the next happy year.

Examples

Input	Output
8989	9012
1001	1023