# Lab: Data Types and Variables

Problems for exercise and homework for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3953/programming-fundamentals-with-python-january-2023).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/1721>.

## 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 |

first\_name = input()  
second\_name = input()  
delimiter = input()  
  
print(f'{first\_name}{delimiter}{second\_name}')

### Hints

* Read the data:



* Print:



## 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 |

meters = int(input())  
kms = meters /1000  
  
print(f'{kms:.2f}')

### Hints

* First, we read the input number:



* Then, we convert it to km:



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



## 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 |

gbp = int(input())  
usd = gbp \* 1.31  
print(f'{usd:.3f}')

### Hints

* Read the pounds:



* Convert them to dollars:



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



## 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 |

centuries = int(input())  
years = centuries \* 100  
days = int(years \* 365.2422)  
hours = days \* 24  
minutes = hours \* 60  
  
print(f'{centuries} centuries = {years} years = {days} days = {hours} hours = {minutes} minutes')

### Hints

* Assume that one year has 365.2422 days on average ([the Tropical year](https://en.wikipedia.org/wiki/Tropical_year)).

## 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 |

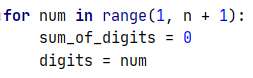
n = int(input())  
  
for num in range(1,n+1):  
 sum\_digits = 0  
 for digit in str(num):  
 sum\_digits += int(digit)  
 if sum\_digits == 5 or sum\_digits ==7 or sum\_digits == 11:  
 print(f'{num} -> True')  
 else:  
 print(f'{num} -> False')

### Hints

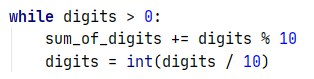
* First, we read the data:



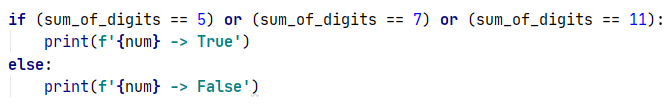
* Iterate from 1 to n (we write n+1 because the for loop in Python iterates from 1 to n-1 by default):



* 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.



* Finally, print the result:



## 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 |

year = int(input())  
year += 1  
  
while True:  
 year\_as\_string = str(year)  
   
 if len(year\_as\_string) != len(set(year\_as\_string)):  
 year += 1  
  
 else:  
 break  
  
print(year)