* Print Hello, World! to stdout.

Ans:- print("Hello, World!")

* Given an integer, , perform the following conditional actions:
* If  is odd, print Weird
* If  is even and in the inclusive range of  to , print Not Weird
* If  is even and in the inclusive range of  to , print Weird
* If  is even and greater than , print Not Weird

Ans:- n=int(input())

print("Weird") if n%2!=0 else print("Not Weird") if n>=2 and n<=5 else print("Weird") if n>=6 and n<=20 else print("Not Weird")

* The provided code stub reads two integers from STDIN,  a and b. Add code to print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).
3. The third line contains the product of the two numbers.

Ans:- a = int(input())

b = int(input())

print(a+b)

print(a-b)

print(a\*b)

* The first line contains the first integer,a .  
  The second line contains the second integer,b

The result of the integer division a//b.

The result of the float division is a/b.

Ans:- a = int(input())

b = int(input())

print(a//b)

print(a/b)

* The provided code stub reads and integer, n, from STDIN. For all non-negative integers i<n , print i2 .

Ans:- n = int(input())

print(\*(i\*\*2 for i in range(n)), sep="\n")

* The included code stub will read an integer,n , from STDIN.Without using any string methods, try to print the following 123…n Print the list of integers from 1 through n as a string, without spaces.

Ans:- n = int(input())

print(\*(range(1, n + 1)), sep="")

* You are given a string and your task is to swap cases. In other words, convert all lowercase letters to uppercase letters and vice versa.

Ans:- swap\_case=lambda s: s.swapcase()

* You are given a string. Split the string on a " " (space) delimiter and join using a - hyphen.

Ans:- split\_and\_join=lambda line: "-".join(line.split(" "))

* Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True, otherwise return False.Note that the code stub provided reads from STDIN and passes arguments to the is\_leap function.

Ans:- is\_leap=lambda year: (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

* A close-up of a number

  Description automatically generated

*int arr:* an array of integers

**Returns**:*float:* the resulting float value rounded to 3 places after the decimal

Ans:- average=lambda array: f"{sum(set(array)) / len(set(array)):.3f}"

* The first line contains n. The second line contains an array A[]  of n integers each separated by a space. Print the runner-up score. Given list is [2,3,6,6,5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

Ans:-     n = int(input())

    arr = map(int, input().split())

     print(sorted(set(arr),reverse=True)[1])

* You are given three integers  x,y and z representing the dimensions of a cuboid along with an integer n. Print a list of all possible coordinates given by (i,j,k) on a 3D grid where the sum of i+j+k is not equal to n. Four integers  x,y,z and n, each on a separate line.

Input:- 1

1

1

2

Output:- [[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]

Ans:-  x = int(input())

     y = int(input())

     z = int(input())

     n = int(input())

     print([[a,b,c] for a in range(x+1) for b in range(y+1) for c in range(z+1) if a+b+c != n])

* You are given the firstname and lastname of a person on two different lines. Your task is to read them and print the following:

Hello firstname lastname! You just delved into python.

Ans:- print\_full\_name=lambda first,last: print("Hello ",first," ",last,"! You just delved into python.",sep="")

* Complete the *mutate\_string* function.

*mutate\_string* has the following parameters:

*string string:* the string to change

*int position:* the index to insert the character at

*string character:* the character to insert

**Returns** *string:* the altered string

Ans:- mutate\_string=lambda string, position, character: "".join([character if x==position else y for x,y in enumerate(string)])

* The provided code stub will read in a dictionary containing key/value pairs of name:[marks] for a list of students. Print the average of the marks array for the student name provided, showing 2 places after the decimal.

Input: The first line contains the integer n, the number of students' records. The next n lines contain the names and marks obtained by a student, each value separated by a space. The final line contains **query\_name**, the name of a student to query.

**Output Format**:Print one line: The average of the marks obtained by the particular student correct to 2 decimal places.

Ans:-  n = int(input())

     student\_marks = {}

     for i in range(n):

        name, \*line = input().split()

        scores = list(map(float, line))

        student\_marks[name] = scores

     query\_name = input()

     print(f"{sum(student\_marks[query\_name])/len(student\_marks[query\_name]):.2f}")

* User enters a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left. The first line of input contains the original string. The next line contains the substring. Output the integer number indicating the total number of occurrences of the substring in the original string.

Input:-

ABCDCDC

CDC

Output:-

2

Ans:- count\_substring=lambda string, sub\_string: len(set(string.find(sub\_string,i) for i in range(len(string)) if string.find(sub\_string,i)!=-1))

* A single line of input containing the full name, S. Print the capitalized string, S.

**Sample Input:** chris alan

**Sample Output:** Chris Alan

Ans:- solve=lambda s: "".join([y.upper() if (x == " " and y != " ") else y for x, y in zip(" " + s, s)])

* You are given a string s and width w.  
  Your task is to wrap the string s into a paragraph of width w.

**Sample Input:** ABCDEFGHIJKLIMNOQRSTUVWXYZ

4

**Sample Output:**

ABCD

EFGH

IJKL

IMNO

QRST

UVWX

YZ

Ans:- wrap=lambda string, max\_width: "\n".join([string[i:i+max\_width] for i in range(0,len(string),max\_width)])

* **Input Format**: The first line contains a, the second line contains b, and the third line contains m.

**Output:** Print two lines.  
On the first line, print the result of pow(a,b). On the second line, print the result of pow(a,b,m),i.e., a^b mod m.

Ans:- a=int(input())

b=int(input())

m=int(input())

print(a\*\*b)

print((a\*\*b)%m)

* **Input Format**: The first line contains the first integer,a , and the second line contains the second integer, b.

The first line is the integer division a//b.  
The second line is the result of the modulo operator: a%b.  
The third line prints the divmod of a and b.

**Sample Input**

177

10

**Sample Output**

17

7

(17, 7)

Ans:- a=int(input())

b=int(input())

print(a//b)

print(a%b)

print((a//b,a%b))

* You are given a positive integer N. Print a numerical triangle of height N-1 like the one below:

1

22

333

4444

55555

......

Ans:- for i in range(1,int(input())):

    print(((10\*\*i-1)//9)\*i)

* Read four numbers,a ,b ,c and d, and print the result of ab+cd.

Ans:- a=int(input())

b=int(input())

c=int(input())

d=int(input())

print(a\*\*b+c\*\*d)

* [**Polar coordinates**](https://en.wikipedia.org/wiki/Polar_coordinate_system) are an alternative way of representing Cartesian coordinates or [Complex Numbers](https://en.wikipedia.org/wiki/Complex_number). A complex number  z=x+yj is completely determined by its real part x and imaginary part y.  
  Here, j is the [imaginary unit](https://en.wikipedia.org/wiki/Imaginary_unit). A polar coordinate (r,w) is completely determined by modulus r and phase angle w.

A blue arrow pointing up

Description automatically generated

If we convert complex number z to its polar coordinate, we find:  
r: Distance from z to origin, i.e., math.sqrt(x^2+y^2)   
w: Counter clockwise angle measured from the positive x-axis to the line segment that joins z to the origin.

**Input Format**: A single line containing the complex number z. Note: complex() function can be used in python to convert the input as a complex number.

Output two lines:  
The first line should contain the value of r.  
The second line should contain the value of w.

Ans:- import math

z=complex(input())

print((z.real\*\*2+z.imag\*\*2)\*\*0.5)

print(math.atan2(z.imag,z.real))

 .