

UNIVERSIDAD GALILEO
INSTITUTO DE INVESTIGACIÓN DE OPERACIONES
MAESTRÍA EN DATA SCIENCE
ESTADÍSTICA APLICADA A LA CIENCIA DE DATOS I



TAREA 4

EDGAR RUBÉN ROJAS OVANDO

CARNÉ: 21000567

CIUDAD DE GUATEMALA, GUATEMALA, MARZO DE 2021

USUARIO: **rbencho**

- **PY IF ELSE PROBLEM**

Practice > Python > Introduction > Python If-Else

Python If-Else ★

Problem

Submissions

Leaderboard

Discussions

Editorial 

Tutorial

Check [Tutorial](#) tab to know how to solve.

Task

Given an integer, n , perform the following conditional actions:

- If n is odd, print `Weird`
- If n is even and in the inclusive range of **2** to **5**, print `Not Weird`
- If n is even and in the inclusive range of **6** to **20**, print `Weird`
- If n is even and greater than **20**, print `Not Weird`

Input Format

A single line containing a positive integer, n .

Constraints

- $1 \leq n \leq 100$

Output Format

Print `Weird` if the number is weird. Otherwise, print `Not Weird`.

Sample Input 0

```
3
```

Sample Output 0

```
Weird
```

Sample Input 1

```

1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9
10 def operar(n):
11     if n > 0 and n <= 100 :
12         if(n%2 == 0):
13             if(n in range (2,5)):
14                 print("Not Weird")
15             elif(n in range (6,20)):
16                 print("Weird")
17             elif(n > 20):
18                 print("Not Weird")
19         else:
20             print("Weird")
21
22
23 if __name__ == '__main__':
24     n = int(input().strip())
25     operar(n)
26
27
28

```

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Input (stdin) [Download](#)

13

Expected Output [Download](#)

1Weird

Problem	Submissions	Leaderboard	Discussions	Editorial	Tutorial
RESULT	SCORE	LANGUAGE	TIME		
Accepted	10.0	Python 3	a minute ago	View Results	

• PYTHON ARITHMETIC OPERATORS

Practice > Python > Introduction > Arithmetic Operators

Arithmetic Operators ★

Problem

Submissions

Leaderboard

Discussions

Editorial 𐀀

Tutorial

Check [Tutorial](#) tab to know how to solve.

Task

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.

Example

a = 3

b = 5

Print the following:

```
8
-2
15
```

```
1  def operar(a,b):
2
3     suma = a + b
4     diferencia = a - b
5     producto = a * b
6
7     print(suma)
8     print(diferencia)
9     print(producto)
10
11 if __name__ == '__main__':
12     a = int(input())
13     b = int(input())
14     operar(a,b)
15
16
```

Test case 0

Test case 1

Compiler Message

Success

Input (stdin) [Download](#)

1 3
2 2

Expected Output [Download](#)

1 5
2 1
3 6

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

RESULT	SCORE	LANGUAGE	TIME	
Accepted	10.0	Python 3	a few seconds ago	View Results

• Python Division

Python: Division ★

Rani

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Check the [Tutorial](#) tab to know learn about division operators.

Task

The provided code stub reads two integers, a and b , from STDIN.

Add logic to print two lines. The first line should contain the result of integer division, $a // b$. The second line should contain the result of float division, a / b .

No rounding or formatting is necessary.

Example

$a = 3$
 $b = 5$

- The result of the integer division $3 // 5 = 0$.
- The result of the float division is $3 / 5 = 0.6$.

Print:

```
0
0.6
```

```
1  ✓ if __name__ == '__main__':
2      a = int(input())
3      b = int(input())
4
5      valor1 = a/b
6      valor2 = a/b
7
8      print(int(valor1))
9      print(valor2)
```

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

[Next Challenge](#)

✓ Test case 0

✓ Test case 1

Compiler Message

Success

Input (stdin) [Download](#)

1 4

2 3

Expected Output [Download](#)

1 1

2 1.333333333333

Problem	Submissions	Leaderboard	Discussions	Editorial	Tutorial
RESULT	SCORE	LANGUAGE	TIME		
✓ Accepted	10.0	Python 3	a few seconds ago	View Results	

- Loops

Loops ★

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Check [Tutorial](#) tab to know how to solve.

Task

The provided code stub reads and Integer, n , from STDIN. For all non-negative integers $i < n$, print i^2 .

Example

$n = 3$

The list of non-negative integers that are less than $n = 3$ is $[0, 1, 2]$. Print the square of each number on a separate line.

```
0
1
4
```

```
1 ✓ if __name__ == '__main__':
2     n = int(input())
3
4 ✓     for i in range(0,n):
5         print(i**2)
```

✓ Test case 0

✓ Test case 1

Compiler Message

Success

Input (stdin)



[Download](#)

```
1 5
```

Expected Output


[Download](#)

```
1 0
2 1
3 4
4 9
5 16
```

Problem	Submissions	Leaderboard	Discussions	Editorial 	Tutorial
RESULT	SCORE	LANGUAGE	TIME		
 Accepted	10.0	Python 3	a minute ago	View Results	

- **Write a Function**

Write a function ★

Problem	Submissions	Leaderboard	Discussions	Editorial 
---------	-------------	-------------	-------------	---

An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

- The year can be evenly divided by 4, is a leap year, unless:
 - The year can be evenly divided by 100, it is NOT a leap year, unless:
 - The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. [Source](#)

Task

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. It is only necessary to

```

1  def is_leap(year):
2      leap = False
3
4      if(year % 4 == 0):
5          if(year % 100 == 0):
6
7              if (year % 400 ==0):
8                  leap = True
9                  return leap
10             leap = False
11             return leap
12         leap = True
13         return leap
14
15     return leap
16
17 > year = int(input()) ...

```


Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Input (stdin) [Download](#)

1 1990

Expected Output [Download](#)

1 False

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
Accepted	10.0	Python 3	a minute ago	View Results

Find Second Maximum Number in a List

Find the Runner-Up Score! ★

■
R

Problem

Submissions

Leaderboard

Discussions

Editorial

Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. You are given n scores. Store them in a list and find the score of the runner-up.

Input Format

The first line contains n . The second line contains an array $A[]$ of n integers each separated by a space.

Constraints

- $2 \leq n \leq 10$
- $-100 \leq A[i] \leq 100$

```

✓ if __name__ == '__main__':
    n = int(input())
    arr = list(map(int, input().split()))

    maxNumber = max(arr)
    lista = []

    for i in arr:
        if(i < maxNumber):
            lista.append(i)

    print(max(lista))

```

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

Compiler Message

Success

Input (stdin) [Download](#)

1	5
2	2 3 6 6 5

Expected Output [Download](#)

1	5
---	---

[Problems](#)
[Contests](#)
[Leaderboard](#)
[Help](#)
[Feedback](#)

Find the Runner-Up Score! ★

Rank: 6

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
✓ Accepted	10.0	Python 3	2 minutes ago	View Results

- Finding the percentage

Finding the percentage ★

Ran

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

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.

Example

marks key:value pairs are

'alpha': [20, 30, 40]

'beta': [30, 50, 70]

query_name = 'beta'

The **query_name** is 'beta'. beta's average score is $(30 + 50 + 70)/3 = 50.0$.

Input Format

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.

Change Theme

Python 3



```
1 def promedio(punteos):
2     suma = 0
3     iteraciones = 0
4     for i in punteos:
5         suma += i
6         iteraciones += 1
7
8     return (suma/iteraciones)
9
10 if __name__ == '__main__':
11     n = int(input())
12     student_marks = {}
13     for _ in range(n):
14         name, *line = input().split()
15         scores = list(map(float, line))
16         student_marks[name] = scores
17     query_name = input()
18
19     print("%.2f" % (promedio(student_marks[query_name])))
20
21
22
23
24
```

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Input (stdin) [Download](#)

1 3
2 Krishna 67 68 69
3 Arjun 70 98 63
4 Malika 52 56 60
5 Malika

Expected Output [Download](#)

1 56.00

Finding the percentage ★

Rank: 1

Problem	Submissions	Leaderboard	Discussions	Editorial	Tutorial
RESULT	SCORE	LANGUAGE	TIME		
Accepted	10.0	Python 3	a minute ago	View Results	

• Python Lists

Lists ★

Rank: 1

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Consider a list (`list = []`). You can perform the following commands:

- `insert i e`: Insert Integer e at position i .
- `print`: Print the list.
- `remove e`: Delete the first occurrence of Integer e .
- `append e`: Insert Integer e at the end of the list.
- `sort`: Sort the list.
- `pop`: Pop the last element from the list.
- `reverse`: Reverse the list.

Initialize your list and read in the value of n followed by n lines of commands where each command will be of the 7 types listed above. Iterate through each command in order and perform the corresponding operation on your list.

```

1  ✓ if __name__ == '__main__':
2      N = int(input())
3
4      arreglo = []
5
6  ✓   for i in range(N):
7       entrada = input().split()
8  ✓       for i in range(1, len(entrada)):
9           entrada[i] = int(entrada[i])
10
11  ✓       if entrada[0] == "append":
12           arreglo.append(entrada[1])
13  ✓       elif entrada[0] == "extend":
14           arreglo.extend(entrada[1:])
15  ✓       elif entrada[0] == "insert":
16           arreglo.insert(entrada[1], entrada[2])
17  ✓       elif entrada[0] == "remove":
18           arreglo.remove(entrada[1])
19  ✓       elif entrada[0] == "pop":
20           arreglo.pop()
21  ✓       elif entrada[0] == "index":
22           print(arreglo.index(entrada[1]))
23  ✓       elif entrada[0] == "count":
24           print(arreglo.count(entrada[1]))
25  ✓       elif entrada[0] == "sort":
26           arreglo.sort()
27  ✓       elif entrada[0] == "reverse":
28           arreglo.reverse()
29  ✓       elif entrada[0] == "print":
30           print(arreglo)
31
32

```

✓ Test case 0

✓ Test case 1 

```

5  print
6  remove 6
7  append 9
8  append 1
9  sort
10 print
11 pop
12 reverse
13 print

```

Expected Output

[Download](#)

```

1  [6, 5, 10]
2  [1, 5, 9, 10]
3  [9, 5, 1]

```

Lists ★

Ran!

Problem

Submissions

Leaderboard

Discussions

Editorial 

Tutorial

RESULT

SCORE

LANGUAGE

TIME

✓ Accepted

10.0

Python 3

a few seconds ago

[View Results](#)

• Python Tuples

Tuples ★

Ran

Problem

Submissions

Leaderboard

Discussions

Editorial 🔒

Tutorial

Task

Given an integer, n , and n space-separated integers as Input, create a tuple, t , of those n integers. Then compute and print the result of $hash(t)$.

Note: `hash()` is one of the functions in the `__builtins__` module, so it need not be imported.

Input Format

The first line contains an integer, n , denoting the number of elements in the tuple.

The second line contains n space-separated integers describing the elements in tuple t .

Output Format

Print the result of $hash(t)$.

Sample Input 0

```
2
1 2
```

Sample Output 0

```
3713081631934410656
```

```
1  import builtins
2
3  ✓ if __name__ == '__main__':
4      n = int(input())
5      integer_list = map(int, input().split())
6
7      tupla = tuple(integer_list)
8
9      print(hash(tupla))
```

✔ Test case 0

✔ Test case 1

Compiler Message

Success

Input (stdin)

Download

1	2
2	1 2

Expected Output

Download

1	3713081631934410656
---	---------------------

Tuples ★

Problem	Submissions	Leaderboard	Discussions	Editorial	Tutorial
RESULT	SCORE	LANGUAGE	TIME		
✔ Accepted	10.0	Python 3	a minute ago		View Results

- DefaultDict tutorial

DefaultDict Tutorial ★

 Ran

Problem

Submissions

Leaderboard

Discussions

Editorial 

The defaultdict tool is a container in the collections class of Python. It's similar to the usual dictionary (dict) container, but the only difference is that a defaultdict will have a default value if that key has not been set yet. If you didn't use a defaultdict you'd have to check to see if that key exists, and if it doesn't, set it to what you want.

For example:

```
from collections import defaultdict
d = defaultdict(list)
d['python'].append("awesome")
d['something-else'].append("not relevant")
d['python'].append("language")
for i in d.items():
    print i
```

This prints:

```
('python', ['awesome', 'language'])
('something-else', ['not relevant'])
```

In this challenge, you will be given 2 integers, n and m . There are n words, which might repeat, in word group A . There are m words belonging to word group B . For each m words, check whether the word has appeared in group A or not. Print the indices of each occurrence of m in group A . If it does not appear, print -1 .

```
1  # Enter your code here. Read input from STDIN. Print output to STDOUT
2
3  from collections import defaultdict
4
5  n, m = map(int, input().split())
6
7  d = defaultdict(list)
8
9  for i in range(n):
10     d[input()].append(i+1)
11
12  for j in range(m):
13     s = input()
14
15     if(s in d):
16         print(*d[s])
17     else:
18         print(-1)
```


Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Input (stdin)

Download

15 2

2a

3a

4b

5a

6b

7a

8b

DefaultDict Tutorial ★

Rank: 1

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
Accepted	20.0	Python 3	a minute ago	View Results

• Collections – Counter

collections.Counter() ★

Rai

Problem

Submissions

Leaderboard

Discussions

Editorial 𐀀

collections.Counter()

A counter is a container that stores elements as dictionary keys, and their counts are stored as dictionary values.

Sample Code

```
>>> from collections import Counter
>>>
>>> myList = [1,1,2,3,4,5,3,2,3,4,2,1,2,3]
>>> print Counter(myList)
Counter({2: 4, 3: 4, 1: 3, 4: 2, 5: 1})
>>>
>>> print Counter(myList).items()
[(1, 3), (2, 4), (3, 4), (4, 2), (5, 1)]
>>>
>>> print Counter(myList).keys()
[1, 2, 3, 4, 5]
>>>
>>> print Counter(myList).values()
[3, 4, 4, 2, 1]
```

Task

Raghu is a shoe shop owner. His shop has X number of shoes.

He has a list containing the size of each shoe he has in his shop.

There are N number of customers who are willing to pay x_i amount of money only if they get the shoe of their desired size.

Your task is to compute how much money *Raghu* earned.

```
1  # Enter your code here. Read input from STDIN. Print output to STDOUT
2
3  import collections
4
5  X = int(input())
6  tamanios = collections.Counter(map(int, input().split()))
7  N = int(input())
8
9  dinero = 0
10
11  for i in range(N):
12      (tamanio, precio) = map(int, input().split())
13
14      if tamanios[tamanio] > 0:
15          tamanios[tamanio] -= 1
16          dinero += precio
17
18  print(dinero)
```

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

Compiler Message

Success

Input (stdin) [Download](#)

```

1 10
2 2 3 4 5 6 8 7 6 5 18
3 6
4 6 55
5 6 45
6 6 55
7 4 40
8 18 60
9 10 50

```

collections.Counter() ★

Rank:

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
✓ Accepted	10.0	Python 3	a few seconds ago	View Results

• Minimum Swaps 2

Minimum Swaps 2 ★

Problem

Submissions

Leaderboard

Discussions

Editorial 

You are given an unordered array consisting of consecutive integers $\in [1, 2, 3, \dots, n]$ without any duplicates. You are allowed to swap any two elements. You need to find the minimum number of swaps required to sort the array in ascending order.

For example, given the array $arr = [7, 1, 3, 2, 4, 5, 6]$ we perform the following steps:

i	arr	swap (indices)
0	[7, 1, 3, 2, 4, 5, 6]	swap (0,3)
1	[2, 1, 3, 7, 4, 5, 6]	swap (0,1)
2	[1, 2, 3, 7, 4, 5, 6]	swap (3,4)
3	[1, 2, 3, 4, 7, 5, 6]	swap (4,5)
4	[1, 2, 3, 4, 5, 7, 6]	swap (5,6)
5	[1, 2, 3, 4, 5, 6, 7]	

It took 5 swaps to sort the array.

Function Description

Complete the function `minimumSwaps` in the editor below. It must return an integer representing the minimum number of swaps to sort the array.

`minimumSwaps` has the following parameter(s):

- `arr`: an unordered array of integers

```
1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9  # Complete the minimumSwaps function below.
10 def minimumSwaps(arr):
11     answer = 0
12
13     for i in range(len(arr)):
14         while(arr[i] != i+1):
15             temp = arr[i]
16             arr[i] = arr[temp-1]
17             arr[temp-1] = temp
18             answer = answer + 1
19     return answer
20
21 if __name__ == '__main__':
22     fptr = open(os.environ['OUTPUT_PATH'], 'w')
23
24     n = int(input())
25
26     arr = list(map(int, input().rstrip().split()))
27
28     res = minimumSwaps(arr)
29
30     fptr.write(str(res) + '\n')
31
32     fptr.close()
```

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Test case 7

Compiler Message

Success

Input (stdin)

Download

1 4

2 4 3 1 2

Expected Output

Download

1 3

Minimum Swaps 2 ★

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
Accepted	40.0	Python 3	a minute ago	View Results

- Floor, Ceil and Rint

Floor, Ceil and Rint ★

Rar

Problem

Submissions

Leaderboard

Discussions

Editorial 

floor

The tool floor returns the floor of the Input element-wise.

The floor of x is the largest Integer i where $i \leq x$.

```
import numpy

my_array = numpy.array([1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9])
print numpy.floor(my_array)      #[ 1.  2.  3.  4.  5.  6.  7.  8.  9.]
```

ceil

The tool ceil returns the ceiling of the Input element-wise.

The ceiling of x is the smallest Integer i where $i \geq x$.

```
import numpy

my_array = numpy.array([1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9])
print numpy.ceil(my_array)      #[ 2.  3.  4.  5.  6.  7.  8.  9. 10.]
```

rint

The rint tool rounds to the nearest Integer of Input element-wise.

```
import numpy

my_array = numpy.array([1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9])
print numpy.rint(my_array)      #[ 1.  2.  3.  4.  6.  7.  8.  9. 10.]
```

Task

You are given a 1-D array, A . Your task is to print the *floor*, *ceil* and *rint* of all the elements of A .

```
1  import numpy
2  numpy.set_printoptions(legacy="1.13")
3
4  arreglo = numpy.array(list(map(float, input().split())))
5
6  print(numpy.floor(arreglo))
7  print(numpy.ceil(arreglo))
8  print(numpy.rint(arreglo))
```


- **Sorting: Bubble Sort**


Sorting: Bubble Sort ★

Problem

Submissions

Leaderboard

Discussions

Editorial 

Consider the following version of Bubble Sort:

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < n - 1; j++) {  
        // Swap adjacent elements if they are in decreasing order  
        if (a[j] > a[j + 1]) {  
            swap(a[j], a[j + 1]);  
        }  
    }  
}
```

Given an array of integers, sort the array in ascending order using the Bubble Sort algorithm above. Once sorted, print the following three lines:

1. Array is sorted in `numSwaps` swaps., where *numSwaps* is the number of swaps that took place.
2. First Element: `firstElement`, where *firstElement* is the first element in the sorted array.
3. Last Element: `lastElement`, where *lastElement* is the last element in the sorted array.

Hint: To complete this challenge, you must add a variable that keeps a running tally of all swaps that occur during execution.

```
1  #!/bin/python3  
2  
3  import math  
4  import os  
5  import random  
6  import re  
7  import sys  
8  from itertools import product  
9  
10 # Complete the countSwaps function below.  
11 def countSwaps(a):  
12     swaps = 0  
13     for i,j in product(range(len(a)), range(len(a)-1)):  
14         if a[j]>a[j+1]:  
15             a[j],a[j+1] = a[j+1],a[j]  
16             swaps += 1  
17     print("Array is sorted in %s swaps." %swaps)  
18     print("First Element:", a[0])  
19     print("Last Element:", a[-1])  
20  
21 if __name__ == '__main__':  
22     n = int(input())  
23  
24     a = list(map(int, input().rstrip().split()))  
25  
26     countSwaps(a)  
27
```


✔ Test case 0

✔ Test case 1

✔ Test case 2

✔ Test case 3

Compiler Message

Success

Input (stdin) [Download](#)

1

3

2

1 2 3

Expected Output [Download](#)

1

Array is sorted in 0 swaps.

2

First Element: 1

3

Last Element: 3

Sorting: Bubble Sort ★

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
✔ Accepted	30.0	Python 3	a minute ago	View Results

• Arrays

Arrays ★

Problem

Submissions

Leaderboard

Discussions

Editorial 

The NumPy (Numeric Python) package helps us manipulate large arrays and matrices of numeric data.

To use the NumPy module, we need to import it using:

```
import numpy
```

Arrays

A NumPy array is a grid of values. They are similar to lists, except that every element of an array must be the same type.

```
import numpy

a = numpy.array([1,2,3,4,5])
print a[1]      #2

b = numpy.array([1,2,3,4,5],float)
print b[1]      #2.0
```

In the above example, `numpy.array()` is used to convert a list into a NumPy array. The second argument (`float`) can be used to set the type of array elements.


Task

You are given a space separated list of numbers.

Your task is to print a reversed NumPy array with the element type `float`.

```
1 import numpy
2
3 def arrays(arr):
4     arrnumpy = numpy.array(arr, float)
5     return [arrnumpy[::-1]]
6
7 arr = input().strip().split(' ')
8 result = arrays(arr)
9 print(result)
```

✓ Test case 0

✓ Test case 1 

Compiler Message

Success

Input (stdin)

[Download](#)

```
1 1 2 3 4 -8 -10
```

Expected Output

[Download](#)

```
1 [-10. -8.  4.  3.  2.  1.]
```

- Zeros and Ones

Zeros and Ones ★

Problem

Submissions

Leaderboard

Discussions

Editorial 

zeros

The zeros tool returns a new array with a given shape and type filled with 0's.

```
import numpy

print numpy.zeros((1,2))           #Default type is float
#Output : [[ 0.  0.]]

print numpy.zeros((1,2), dtype = numpy.int) #Type changes to int
#Output : [[0 0]]
```

ones

The ones tool returns a new array with a given shape and type filled with 1's.

```
import numpy

print numpy.ones((1,2))           #Default type is float
#Output : [[ 1.  1.]]

print numpy.ones((1,2), dtype = numpy.int) #Type changes to int
#Output : [[1 1]]
```

Task

You are given the shape of the array in the form of space-separated integers, each Integer representing the size of different dimensions, your task is to print an array of the given shape and integer type using the tools `numpy.zeros` and `numpy.ones`.

```
1  import numpy
2  dimensiones = [int(x) for x in input().strip().split()]
3
4  print(numpy.zeros(tuple(dimensiones), dtype = numpy.int))
5  print(numpy.ones(tuple(dimensiones), dtype = numpy.int))
```

Test case 0

Test case 1

Test case 2

Test case 3

Compiler Message

Success

Input (stdin)

Download

13 3 3

Expected Output

Download

1[[0 0 0]
2[0 0 0]
3[0 0 0]
4
5[[0 0 0]
6[0 0 0]

Zeros and Ones

Rank

Problem	Submissions	Leaderboard	Discussions	Editorial
RESULT	SCORE	LANGUAGE	TIME	
Accepted	20.0	Python 3	a few seconds ago	View Results