



Say "Hello, World!" With Python ★

91/115 challenges solved

Rank: 10359 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

Here is a sample line of code that can be executed in Python:

```
print("Hello, World!")
```

You can just as easily store a string as a variable and then print it to stdout:

```
my_string = "Hello, World!"  
print(my_string)
```

The above code will print Hello, World! on your screen. Try it yourself in the editor below!

Input Format

You do not need to read any input in this challenge.

Output Format

Print Hello, World! to stdout.

Sample Output 0

```
Hello, World!
```

[Change Theme](#)

Language

Python 3



```
1 print("Hello, World!")  
2
```



Line: 2 Col: 1

 Upload Code as File Test against custom input**Run Code****Submit Code**

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)





Python If-Else ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

Explanation 0

$n = 3$

n is odd and odd numbers are weird, so print Weird.

Sample Input 1

24

Sample Output 1

Not Weird

Explanation 1

n = 24

n > 20 and ***n*** is even, so it is not weird.

Change Theme Language Python 3 ▾ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇

```
1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8  def name:
9      if __name__ == '__main__':
10         n = int(input().strip())
11         if n%2==0:
12             print("Not Weird")
13         else:
14             print("Weired")
15
16
17
```

Line: 17 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code



Arithmetic Operators ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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
```

Input Format

The first line contains the first integer, a .

The second line contains the second integer, b .

Constraints

$1 \leq a \leq 10^{10}$

$1 \leq b \leq 10^{10}$

Output Format

Print the three lines as explained above.

Sample Input 0

```
3
2
```

Sample Output 0

```
5
1
6
```

Explanation 0

$3 + 2 \Rightarrow 5$

$3 - 2 \Rightarrow 1$

$3 * 2 \Rightarrow 6$

```
1 if __name__ == '__main__':
2     a = int(input())
3     b = int(input())
4     add = a + b
5     diff = a - b
6     mul = a * b
7     print (add)
8     print (diff)
9     print (mul)
10
```

[Change Theme](#)[Language](#)

Python 3



Line: 10 Col: 1

[Upload Code as File](#)[Test against custom input](#)[Run Code](#)[Submit Code](#)



Python: Division ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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
```

Input Format

The first line contains the first integer, a .

The second line contains the second integer, b .

Output Format

Print the two lines as described above.

Sample Input 0

```
4
3
```

Sample Output 0

```
1
1.333333333333
```

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

Line: 9 Col: 1

 [Upload Code as File](#)

Test against custom input

[Run Code](#)[Submit Code](#)



Loops ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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
```

Input Format

The first and only line contains the integer, n .

Constraints

$1 \leq n \leq 20$

Output Format

Print n lines, one corresponding to each i .

Sample Input 0

```
5
```

Sample Output 0

```
0
1
4
9
16
```

[Change Theme](#)

Language Python 3



```
1  if __name__ == '__main__':
2      n = int(input())
3      for i in range(n):
4          print(i*i)
5
6
```

Line: 6 Col: 1

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



91/115 challenges solved

Rank: 10360 | Points: 1855 !



Write a function ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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 complete the `is_leap` function.

Input Format

Read `year`, the year to test.

Constraints

$1900 \leq \text{year} \leq 10^5$

Output Format

The function must return a Boolean value (True/False). Output is handled by the provided code stub.

Sample Input 0

1990

Sample Output 0

False

Explanation 0

1990 is not a multiple of 4 hence it's not a leap year.

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

Line: 1 Col: 1

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)



Print Function ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

The included code stub will read an integer, **n**, from STDIN.

Without using any string methods, try to print the following:

123...n

Note that "..." represents the consecutive values in between.

Example

n = 5

Print the string **12345**.

Input Format

The first line contains an integer **n**.

Constraints

1 ≤ n ≤ 150

Output Format

Print the list of integers from **1** through **n** as a string, without spaces.

Sample Input 0

3

Sample Output 0

123

Change Theme Language Python 3 ▾



```
1 if __name__ == '__main__':
2     n = int(input())
3     for j in range(n):
4         print(j+1,end="")
5
```

Line: 5 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



List Comprehensions ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



Let's learn about list comprehensions! 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 . Here,

$0 \leq i \leq x; 0 \leq j \leq y; 0 \leq k \leq z$. Please use list comprehensions rather than multiple loops, as a learning exercise.

Example

$x = 1$

$y = 1$

$z = 2$

$n = 3$

All permutations of $[i, j, k]$ are:

$[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [0, 1, 2], [1, 0, 0], [1, 0, 1], [1, 0, 2], [1, 1, 0], [1, 1, 1], [1, 1, 2]]$.

Print an array of the elements that do not sum to $n = 3$.

$[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0, 1], [1, 1, 0], [1, 1, 2]]$

Input Format

Four integers x, y, z and n , each on a separate line.

Constraints

Print the list in lexicographic increasing order.

Sample Input 0

```
1
1
1
2
```

Sample Output 0

```
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]
```

Explanation 0

Each variable x, y and z will have values of **0** or **1**. All permutations of lists in the form

$[i, j, k] = [[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0, 1], [1, 1, 0], [1, 1, 1]]$.

Remove all arrays that sum to $n = 2$ to leave only the valid permutations.

Sample Input 1

```
2  
2  
2  
2
```

Sample Output 1

```
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 2], [0, 2, 1], [0, 2, 2], [1, 0, 0], [1, 0, 2], [1, 1, 1], [1, 1, 2]]
```

Change Theme Language Python 3 ▾ ⌂ ⌃ ⌄ ⌅

```
1 if __name__ == '__main__':
2     x = int(input())
3     y = int(input())
4     z = int(input())
5     n = int(input())
6     output=[]
7     for i in range(x+1):
8         for j in range(y+1):
9             for k in range(z+1):
10                if i+j+k==n:
11                    continue
12                else:
13                    output.append([i,j,k])
14
15 print(output)
16
```

Line: 16 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code



Find the Runner-Up Score! ★

91/115 challenges solved

Rank: 10360 | Points: 1855

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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

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

Output Format

Print the runner-up score.

Sample Input 0

```
5
2 3 6 6 5
```

Sample Output 0

```
5
```

Explanation 0

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.

[Change Theme](#)

Language

Python 3



```
1 if __name__ == '__main__':
2     n = int(input())
3     arr = map(int, input().split())
4     print(sorted(list(set(arr)))[-2])
5
```

Line: 5 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Nested Lists ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



Given the names and grades for each student in a class of N students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

Note: If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

Example

records = [["chi", 20.0], ["beta", 50.0], ["alpha", 50.0]]

The ordered list of scores is [20.0, 50.0], so the second lowest score is 50.0. There are two students with that score: ["beta", "alpha"].

Ordered alphabetically, the names are printed as:

```
alpha
beta
```

Input Format

The first line contains an integer, N , the number of students.

The $2N$ subsequent lines describe each student over 2 lines.

- The first line contains a student's name.
- The second line contains their grade.

Constraints

- $2 \leq N \leq 5$
- There will always be one or more students having the second lowest grade.

Output Format

Print the name(s) of any student(s) having the second lowest grade in. If there are multiple students, order their names alphabetically and print each one on a new line.

Sample Input 0

```
5
Harry
37.21
Berry
37.21
Tina
37.2
Akriti
41
Harsh
39
```

Sample Output 0

<https://www.hackerrank.com/challenges/nested-list/problem>

Berry
Harry

Explanation 0

There are **5** students in this class whose names and grades are assembled to build the following list:

```
python students = [['Harry', 37.21], ['Berry', 37.21], ['Tina', 37.2], ['Akriti', 41], ['Harsh', 39]]
```

The lowest grade of **37.2** belongs to Tina. The second lowest grade of **37.21** belongs to both Harry and Berry, so we order their names alphabetically and print each name on a new line.

Change Theme Language Python 3 ▼ ↻ ⤶ ⤷ ⋮

```
1 if __name__ == '__main__':
2     score_list = []
3     for _ in range(int(input())):
4         name = input()
5         score = float(input())
6         score_list.append([name, score])
7 second_highest = sorted(set([score for name, score in score_list]))[1]
8 print('\n'.join(sorted([name for name, score in score_list if score == second_highest])))
9
```

Line: 9 Col: 1

Test against custom input



Finding the percentage ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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.

Constraints

- $2 \leq n \leq 10$
- $0 \leq \text{marks}[i] \leq 100$
- length of marks arrays = 3

Output Format

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

Sample Input 0

```
3
Krishna 67 68 69
Arjun 70 98 63
Malika 52 56 60
Malika
```

Sample Output 0

```
56.00
```

Explanation 0

Marks for Malika are {52, 56, 60} whose average is $\frac{52+56+60}{3} \Rightarrow 56$

Sample Input 1

```
2
Harsh 25 26.5 28
Anurag 26 28 30
Harsh
```

Sample Output 1

```
26.50
```

[Change Theme](#)

Language

Python 3



Line: 13 Col: 1

 [Upload Code as File](#)[Test against custom input](#)[Run Code](#)[Submit Code](#)



Lists ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

1. `insert i e`: Insert integer `e` at position `i`.
2. `print`: Print the list.
3. `remove e`: Delete the first occurrence of integer `e`.
4. `append e`: Insert integer `e` at the end of the list.
5. `sort`: Sort the list.
6. `pop`: Pop the last element from the list.
7. `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.

Example

`N = 4`

`append 1`

`append 2`

`insert 3 1`

`print`

- `append 1`: Append `1` to the list, `arr = [1]`.
- `append 2`: Append `2` to the list, `arr = [1, 2]`.
- `insert 3 1`: Insert `3` at index `1`, `arr = [1, 3, 2]`.
- `print`: Print the array.

Output:

`[1, 3, 2]`

Input Format

The first line contains an integer, `n`, denoting the number of commands.

Each line `i` of the `n` subsequent lines contains one of the commands described above.

Constraints

- The elements added to the list must be integers.

Output Format

For each command of type `print`, print the list on a new line.

Sample Input 0

```
12
insert 0 5
insert 1 10
insert 0 6
print
remove 6
append 9
append 1
sort
print
pop
reverse
print
```

Sample Output 0

```
[6, 5, 10]
[1, 5, 9, 10]
[9, 5, 1]
```





Tuples ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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 $\text{hash}(t)$.

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 $\text{hash}(t)$.

Sample Input 0

```
2
1 2
```

Sample Output 0

```
3713081631934410656
```





sWAP cASE ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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.

For Example:

Www.HackerRank.com → wWW.hACKERrANK.COM

Pythonist 2 → pYTHONIST 2

Function Description

Complete the swap_case function in the editor below.

swap_case has the following parameters:

- string s: the string to modify

Returns

- string: the modified string

Input Format

A single line containing a string **s**.

Constraints

$0 < \text{len}(s) \leq 1000$

Sample Input 0

HackerRank.com presents "Pythonist 2".

Sample Output 0

hACKERrANK.COM PRESENTS "pYTHONIST 2".





String Split and Join ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



In Python, a string can be split on a delimiter.

Example:

```
>>> a = "this is a string"
>>> a = a.split(" ") # a is converted to a list of strings.
>>> print a
['this', 'is', 'a', 'string']
```

Joining a string is simple:

```
>>> a = "-".join(a)
>>> print a
this-is-a-string
```

Task

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

Function Description

Complete the `split_and_join` function in the editor below.

`split_and_join` has the following parameters:

- `string line`: a string of space-separated words

Returns

- `string`: the resulting string

Input Format

The one line contains a string consisting of space separated words.

Sample Input

```
this is a string
```

Sample Output

```
this-is-a-string
```





[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



What's Your Name? ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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.
```

Function Description

Complete the print_full_name function in the editor below.

print_full_name has the following parameters:

- string first: the first name
- string last: the last name

Prints

- string: 'Hello **firstname lastname**! You just delved into python' where **firstname** and **lastname** are replaced with **first** and **last**.

Input Format

The first line contains the first name, and the second line contains the last name.

Constraints

The length of the first and last names are each ≤ 10 .

Sample Input 0

```
Ross
Taylor
```

Sample Output 0

```
Hello Ross Taylor! You just delved into python.
```

Explanation 0

The input read by the program is stored as a string data type. A string is a collection of characters.

[Change Theme](#)

Language

Pypy 2





Line: 1 Col: 1

[Upload Code as File](#)[Test against custom input](#)[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Mutations ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



We have seen that lists are mutable (they can be changed), and tuples are immutable (they cannot be changed).

Let's try to understand this with an example.

You are given an immutable string, and you want to make changes to it.

Example

```
>>> string = "abracadabra"
```

You can access an index by:

```
>>> print string[5]
a
```

What if you would like to assign a value?

```
>>> string[5] = 'k'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

How would you approach this?

- One solution is to convert the string to a list and then change the value.

Example

```
>>> string = "abracadabra"
>>> l = list(string)
>>> l[5] = 'k'
>>> string = ''.join(l)
>>> print string
abrackdabra
```

- Another approach is to slice the string and join it back.

Example

```
>>> string = string[:5] + "k" + string[6:]
>>> print string
abrackdabra
```

Task

Read a given string, change the character at a given index and then print the modified string.

Function Description

<https://www.hackerrank.com/challenges/python-mutations/problem>

Complete the `mutate_string` function in the editor below.

`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

Input Format

The first line contains a string, `string`.

The next line contains an integer `position`, the index location and a string `character`, separated by a space.

Sample Input

STDIN	Function
-----	-----
abracadabra	<code>s = 'abracadabra'</code>
5 k	<code>position = 5, character = 'k'</code>

Sample Output

abrackdabra

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 def mutate_string(string, position, character):
2     return
3
4 if __name__ == '__main__': ...
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Find a string ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



In this challenge, the 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.

NOTE: String letters are case-sensitive.

Input Format

The first line of input contains the original string. The next line contains the substring.

Constraints

$1 \leq \text{len(string)} \leq 200$

Each character in the string is an ascii character.

Output Format

Output the integer number indicating the total number of occurrences of the substring in the original string.

Sample Input

```
ABCD CDC  
CDC
```

Sample Output

```
2
```

Concept

Some string processing examples, [such as these](#), might be useful.

There are a couple of new concepts:

In Python, the length of a string is found by the function `len(s)`, where `s` is the string.

To traverse through the length of a string, use a for loop:

```
for i in range(0, len(s)):  
    print (s[i])
```

A range function is used to loop over some length:

```
range (0, 5)
```

Here, the range loops over **0** to **4**. **5** is excluded.



Line: 1 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



91/115 challenges solved

Rank: 10360 | Points: 1855 !



String Validators ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Python has built-in string validation methods for basic data. It can check if a string is composed of alphabetical characters, alphanumeric characters, digits, etc.

[str.isalnum\(\)](#)

This method checks if all the characters of a string are alphanumeric (a-z, A-Z and 0-9).

```
>>> print 'ab123'.isalnum()
True
>>> print 'ab123#'.isalnum()
False
```

[str.isalpha\(\)](#)

This method checks if all the characters of a string are alphabetical (a-z and A-Z).

```
>>> print 'abcD'.isalpha()
True
>>> print 'abcd1'.isalpha()
False
```

[str.isdigit\(\)](#)

This method checks if all the characters of a string are digits (0-9).

```
>>> print '1234'.isdigit()
True
>>> print '123edsd'.isdigit()
False
```

[str.islower\(\)](#)

This method checks if all the characters of a string are lowercase characters (a-z).

```
>>> print 'abcd123#'.islower()
True
>>> print 'Abcd123#'.islower()
False
```

[str.isupper\(\)](#)

This method checks if all the characters of a string are uppercase characters (A-Z).

```
>>> print 'ABCD123#'.isupper()
True
>>> print 'Abcd123#'.isupper()
```

```
False
```

Task

You are given a string S .

Your task is to find out if the string S contains: alphanumeric characters, alphabetical characters, digits, lowercase and uppercase characters.

Input Format

A single line containing a string S .

Constraints

$0 < \text{len}(S) < 1000$

Output Format

In the first line, print True if S has any alphanumeric characters. Otherwise, print False.

In the second line, print True if S has any alphabetical characters. Otherwise, print False.

In the third line, print True if S has any digits. Otherwise, print False.

In the fourth line, print True if S has any lowercase characters. Otherwise, print False.

In the fifth line, print True if S has any uppercase characters. Otherwise, print False.

Sample Input

```
qA2
```

Sample Output

```
True
True
True
True
True
```





91/115 challenges solved

Rank: 10360 | Points: 1855 !



Text Alignment ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



In Python, a string of text can be aligned left, right and center.

.ljust(width)

This method returns a left aligned string of length width.

```
>>> width = 20
>>> print 'HackerRank'.ljust(width,'-')
HackerRank-----
```

.center(width)

This method returns a centered string of length width.

```
>>> width = 20
>>> print 'HackerRank'.center(width,'-')
-----HackerRank-----
```

.rjust(width)

This method returns a right aligned string of length width.

```
>>> width = 20
>>> print 'HackerRank'.rjust(width,'-')
-----HackerRank
```

Task

You are given a partial code that is used for generating the HackerRank Logo of variable thickness.

Your task is to replace the blank (_____) with rjust, ljust or center.

Input Format

A single line containing the thickness value for the logo.

Constraints

The thickness must be an odd number.

0 < thickness < 50

Output Format

Output the desired logo.

Sample Input

Sample Output



Text Wrap ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

You are given a string S and width w .

Your task is to wrap the string into a paragraph of width w .

Function Description

Complete the wrap function in the editor below.

wrap has the following parameters:

- string string: a long string
- int max_width: the width to wrap to

Returns

- string: a single string with newline characters ('\n') where the breaks should be

Input Format

The first line contains a string, $string$.

The second line contains the width, max_width .

Constraints

- $0 < \text{len}(\text{string}) < 1000$
- $0 < \text{max_width} < \text{len}(\text{string})$

Sample Input 0

```
ABCDEFGHIJKLMNOQRSTUVWXYZ  
4
```

Sample Output 0

```
ABCD  
EFGH  
IJKL  
IMNO  
QRST  
UVWX  
YZ
```





[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Designer Door Mat ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Mr. Vincent works in a door mat manufacturing company. One day, he designed a new door mat with the following specifications:

- Mat size must be $N \times M$. (N is an odd natural number, and M is 3 times N .)
- The design should have 'WELCOME' written in the center.
- The design pattern should only use |, . and - characters.

Sample Designs

Size: 7 x 21

```
-----.|-----  
-----.|..|..|.----  
---.|...|..|...|...|.---  
-----.WELCOME-----  
---.|..|..|..|...|.---  
-----.|..|..|.----  
-----.|.-----
```

Size: 11 x 33

```
-----.|-----  
-----.|...|..|.----  
-----.|..|...|...|...|.----  
-----.|..|...|...|...|...|.----  
-----.|..|...|...|...|...|...|.----  
-----.|.WELCOME-----  
---.|...|..|...|...|...|...|.---  
-----.|...|..|...|...|...|.----  
-----.|..|...|...|...|.----  
-----.|...|..|.----  
-----.|.-----
```

Input Format

A single line containing the space separated values of N and M .

Constraints

- $5 < N < 101$
- $15 < M < 303$

Output Format

Output the design pattern.

Sample Input

9 27

Sample Output

<https://www.hackerrank.com/challenges/designer-door-mat/problem>

```
-----.|-----  
----.|..|...|.  
---.|..|..|..|..|..|.  
---.|..|..|..|..|..|..|.  
-----WELCOME-----  
---.|..|..|..|..|..|..|.  
---.|..|..|..|..|..|.  
-----|..|..|..|.  
-----|..|.
```

Change Theme Language Pypy 2 ▾



```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code



String Formatting ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Given an integer, **n**, print the following values for each integer **i** from 1 to **n**:

1. Decimal
2. Octal
3. Hexadecimal (capitalized)
4. Binary

Function Description

Complete the `print_formatted` function in the editor below.

`print_formatted` has the following parameters:

- int number: the maximum value to print

Prints

The four values must be printed on a single line in the order specified above for each **i** from 1 to **number**. Each value should be space-padded to match the width of the binary value of **number** and the values should be separated by a single space.

Input Format

A single integer denoting **n**.

Constraints

- $1 \leq n \leq 99$

Sample Input

17

Sample Output

```
1    1    1    1
2    2    2   10
3    3    3   11
4    4    4  100
5    5    5  101
6    6    6  110
7    7    7  111
8   10    8 1000
9   11    9 1001
10  12    A 1010
11  13    B 1011
12  14    C 1100
13  15    D 1101
14  16    E 1110
15  17    F 1111
16  20   10 10000
```

17 21 11 10001



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Alphabet Rangoli ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given an integer, N . Your task is to print an alphabet rangoli of size N . (Rangoli is a form of Indian folk art based on creation of patterns.)

Different sizes of alphabet rangoli are shown below:

```
#size 3
```

```
----c----  
--c-b-c--  
c-b-a-b-c  
--c-b-c--  
----c----
```

```
#size 5
```

```
-----e-----  
----e-d-e----  
---e-d-c-d-e---  
--e-d-c-b-c-d-e--  
e-d-c-b-a-b-c-d-e  
--e-d-c-b-c-d-e--  
---e-d-c-d-e---  
----e-d-e----  
-----e-----
```

```
#size 10
```

```
-----j-----  
----j-i-j----  
---j-i-h-i-j---  
--j-i-h-g-h-i-j---  
---j-i-h-g-f-g-h-i-j---  
----j-i-h-g-f-e-f-g-h-i-j---  
---j-i-h-g-f-e-d-e-f-g-h-i-j---  
----j-i-h-g-f-e-d-c-d-e-f-g-h-i-j---  
---j-i-h-g-f-e-d-c-b-a-b-c-d-e-f-g-h-i-j---  
--j-i-h-g-f-e-d-c-b-c-d-e-f-g-h-i-j---  
---j-i-h-g-f-e-d-c-d-e-f-g-h-i-j---  
----j-i-h-g-f-e-d-e-f-g-h-i-j---  
----j-i-h-g-f-e-f-g-h-i-j---  
-----j-i-h-g-f-g-h-i-j-----  
-----j-i-h-g-h-i-j-----  
-----j-i-h-i-j-----  
-----j-i-j-----
```

The center of the rangoli has the first alphabet letter a, and the boundary has the N^{th} alphabet letter (in alphabetical order).

Function Description

<https://www.hackerrank.com/challenges/alphabet-rangoli/problem>

Complete the rangoli function in the editor below.

rangoli has the following parameters:

- int size: the size of the rangoli

Returns

- string: a single string made up of each of the lines of the rangoli separated by a newline character (\n)

Input Format

Only one line of input containing **size**, the size of the rangoli.

Constraints

$$0 < \text{size} < 27$$

Sample Input

5

Sample Output

```
-----e-----
-----e-d-e-----
----e-d-c-d-e----
--e-d-c-b-c-d-e--
e-d-c-b-a-b-c-d-e
--e-d-c-b-c-d-e--
----e-d-c-d-e----
-----e-d-e-----
-----e-----
```

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 def print_rangoli(size):
2     # your code goes here
3
4 if __name__ == '__main__': ...
```

Line: 1 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

Prepare > Python > Strings > Capitalize!

Capitalize! ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)[Editorial](#)

You are asked to ensure that the first and last names of people begin with a capital letter in their passports. For example, alison heck should be capitalised correctly as Alison Heck.

alison heck ⇒ **Alison Heck**

Given a full name, your task is to capitalize the name appropriately.

Input Format

A single line of input containing the full name, S .

Constraints

- $0 < \text{len}(S) < 1000$
- The string consists of alphanumeric characters and spaces.

Note: in a word only the first character is capitalized. Example 12abc when capitalized remains 12abc.

Output Format

Print the capitalized string, S .

Sample Input

chris alan

Sample Output

Chris Alan

Author	[deleted]
Difficulty	Easy
Max Score	20
Submitted By	196283

NEED HELP?

[View discussions](#)

[View editorial](#)

[View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)





The Minion Game ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !



Problem

Submissions

Leaderboard

Editorial

RATE THIS CHALLENGE



Kevin and Stuart want to play the '**The Minion Game**'.

Game Rules

Both players are given the same string, S .

Both players have to make substrings using the letters of the string S .

Stuart has to make words starting with consonants.

Kevin has to make words starting with vowels.

The game ends when both players have made all possible substrings.

Scoring

A player gets +1 point for each occurrence of the substring in the string S .

For Example:

String S = BANANA

Kevin's vowel beginning word = ANA

Here, ANA occurs twice in BANANA. Hence, Kevin will get 2 Points.

For better understanding, see the image below:

BANANA			
STUART	KEVIN		
WORDS	SCORE	WORDS	SCORE
B	1	A	3
N	2	AN	2
BA	1	ANA	2
NA	2	ANAN	1
BAN	1	ANANA	1
NAN	1		
BANA	1		
NANA	1		
BANAN	1		
BANANA	1		
TOTAL	12	TOTAL	9

Your task is to determine the winner of the game and their score.

Function Description

<https://www.hackerrank.com/challenges/the-minion-game/problem>

Complete the `minion_game` in the editor below.

`minion_game` has the following parameters:

- string `string`: the string to analyze

Prints

- string: the winner's name and score, separated by a space on one line, or `Draw` if there is no winner

Input Format

A single line of input containing the string S .

Note: The string S will contain only uppercase letters: $[A - Z]$.

Constraints

$$0 < \text{len}(S) \leq 10^6$$

Sample Input

BANANA

Sample Output

Stuart 12

Note :

Vowels are only defined as `AEIOU`. In this problem, `Y` is not considered a vowel.





Merge the Tools! ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Consider the following:

- A string, s , of length n where $s = c_0c_1 \dots c_{n-1}$.
- An integer, k , where k is a factor of n .

We can split s into $\frac{n}{k}$ substrings where each substring, t_i , consists of a contiguous block of k characters in s . Then, use each t_i to create string u_i such that:

- The characters in u_i are a subsequence of the characters in t_i .
- Any repeat occurrence of a character is removed from the string such that each character in u_i occurs exactly once. In other words, if the character at some index j in t_i occurs at a previous index $< j$ in t_i , then do not include the character in string u_i .

Given s and k , print $\frac{n}{k}$ lines where each line i denotes string u_i .

Example

$s = \text{'AAABCADDE'}$

$k = 3$

There are three substrings of length 3 to consider: 'AAA', 'BCA' and 'DDE'. The first substring is all 'A' characters, so $u_1 = \text{'A'}$. The second substring has all distinct characters, so $u_2 = \text{'BCA'}$. The third substring has 2 different characters, so $u_3 = \text{'DE'}$. Note that a subsequence maintains the original order of characters encountered. The order of characters in each subsequence shown is important.

Function Description

Complete the merge_the_tools function in the editor below.

merge_the_tools has the following parameters:

- string s: the string to analyze
- int k: the size of substrings to analyze

Prints

Print each subsequence on a new line. There will be $\frac{n}{k}$ of them. No return value is expected.

Input Format

The first line contains a single string, s .

The second line contains an integer, k , the length of each substring.

Constraints

- $1 \leq n \leq 10^4$, where n is the length of s
- $1 \leq k \leq n$
- It is guaranteed that n is a multiple of k .

Sample Input

STDIN	Function
-----	-----
AABCAAADA	s = 'AABCAAADA'
3	k = 3

Sample Output

AB
CA
AD

Explanation

Split s into $\frac{n}{k} = \frac{9}{3} = 3$ equal parts of length $k = 3$. Convert each t_i to u_i by removing any subsequent occurrences of non-distinct characters in t_i :

1. $t_0 = "AAB" \rightarrow u_0 = "AB"$
2. $t_1 = "CAA" \rightarrow u_1 = "CA"$
3. $t_2 = "ADA" \rightarrow u_2 = "AD"$

Print each u_i on a new line.

Change Theme Language Pypy 2 ▾ ⚙️ ⚡ ⋮

```
1 def merge_the_tools(string, k):  
2     # your code goes here  
3  
4 if __name__ == '__main__': ...
```

Line: 1 Col: 1

Upload Code as File



Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Introduction to Sets ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



A set is an unordered collection of elements without duplicate entries.

When printed, iterated or converted into a sequence, its elements will appear in an arbitrary order.

Example

```
>>> print set()
set([])

>>> print set('HackerRank')
set(['a', 'c', 'e', 'H', 'k', 'n', 'r', 'R'])

>>> print set([1,2,1,2,3,4,5,6,0,9,12,22,3])
set([0, 1, 2, 3, 4, 5, 6, 9, 12, 22])

>>> print set((1,2,3,4,5,5))
set([1, 2, 3, 4, 5])

>>> print set(set(['H','a','c','k','e','r','r','a','n','k']))
set(['a', 'c', 'r', 'e', 'H', 'k', 'n'])

>>> print set({'Hacker' : 'DOSHI', 'Rank' : 616 })
set(['Hacker', 'Rank'])

>>> print set(enumerate(['H','a','c','k','e','r','r','a','n','k']))
set([(6, 'r'), (7, 'a'), (3, 'k'), (4, 'e'), (5, 'r'), (9, 'k'), (2, 'c'), (0, 'H'), (1, 'a'), (8, 'n')])
```

Basically, sets are used for membership testing and eliminating duplicate entries.

Task

Now, let's use our knowledge of sets and help Mickey.

Ms. Gabriel Williams is a botany professor at District College. One day, she asked her student Mickey to compute the average of all the plants with distinct heights in her greenhouse.

Formula used:

$$\text{Average} = \frac{\text{Sum of Distinct Heights}}{\text{Total Number of Distinct Heights}}$$

Function Description

Complete the average function in the editor below.

average has the following parameters:

- int arr: an array of integers

Returns

- float: the resulting float value rounded to 3 places after the decimal

Input Format

The first line contains the integer, N , the size of arr .

The second line contains the N space-separated integers, $\text{arr}[i]$.

Constraints

$$0 < N \leq 100$$

Sample Input

STDIN	Function
-----	-----
10	<code>arr[] size N = 10</code>
161 182 161 154 176 170 167 171 170 174	<code>arr = [161, 181, ..., 174]</code>

Sample Output

169.375

Explanation

Here, $\text{set}([154, 161, 167, 170, 171, 174, 176, 182])$ is the set containing the distinct heights. Using the `sum()` and `len()` functions, we can compute the average.

$$\text{Average} = \frac{1355}{8} = 169.375$$

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅ ⌆

```
1 from __future__ import division
2
3 def average(array):
4     # your code goes here
5
6 if __name__ == '__main__': ...
```

Upload Code as File Test against custom input**Run Code****Submit Code**

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



No Idea! ★

91/115 challenges solved

Rank: 10360 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



There is an array of n integers. There are also 2 disjoint sets, A and B , each containing m integers. You like all the integers in set A and dislike all the integers in set B . Your initial happiness is 0. For each i integer in the array, if $i \in A$, you add 1 to your happiness. If $i \in B$, you add -1 to your happiness. Otherwise, your happiness does not change. Output your final happiness at the end.

Note: Since A and B are sets, they have no repeated elements. However, the array might contain duplicate elements.

Constraints

$$1 \leq n \leq 10^5$$

$$1 \leq m \leq 10^5$$

$$1 \leq \text{Any integer in the input} \leq 10^9$$

Input Format

The first line contains integers n and m separated by a space.

The second line contains n integers, the elements of the array.

The third and fourth lines contain m integers, A and B , respectively.

Output Format

Output a single integer, your total happiness.

Sample Input

```
3 2
1 5 3
3 1
5 7
```

Sample Output

```
1
```

Explanation

You gain 1 unit of happiness for elements 3 and 1 in set A . You lose 1 unit for 5 in set B . The element 7 in set B does not exist in the array so it is not included in the calculation.

Hence, the total happiness is $2 - 1 = 1$.

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



91/115 challenges solved

Rank: 10361 | Points: 1855 !



Symmetric Difference ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Objective

Today, we're learning about a new data type: sets.

Concept

If the inputs are given on one line separated by a character (the delimiter), use `split()` to get the separate values in the form of a list. The delimiter is space (ascii 32) by default. To specify that comma is the delimiter, use `string.split(',')`. For this challenge, and in general on HackerRank, space will be the delimiter.

Usage:

```
>> a = raw_input()
5 4 3 2
>> lis = a.split()
>> print (lis)
['5', '4', '3', '2']
```

If the list values are all integer types, use the `map()` method to convert all the strings to integers.

```
>> newlis = list(map(int, lis))
>> print (newlis)
[5, 4, 3, 2]
```

Sets are an unordered collection of unique values. A single set contains values of any immutable data type.

CREATING SETS

```
>> myset = {1, 2} # Directly assigning values to a set
>> myset = set() # Initializing a set
>> myset = set(['a', 'b']) # Creating a set from a list
>> myset
{'a', 'b'}
```

MODIFYING SETS

Using the `add()` function:

```
>> myset.add('c')
>> myset
{'a', 'c', 'b'}
>> myset.add('a') # As 'a' already exists in the set, nothing happens
>> myset.add((5, 4))

>> myset
```

```
{'a', 'c', 'b', (5, 4)}
```

Using the update() function:

```
>> myset.update([1, 2, 3, 4]) # update() only works for iterable objects
>> myset
{'a', 1, 'c', 'b', 4, 2, (5, 4), 3}
>> myset.update({1, 7, 8})
>> myset
{'a', 1, 'c', 'b', 4, 7, 8, 2, (5, 4), 3}
>> myset.update({1, 6}, [5, 13])
>> myset
{'a', 1, 'c', 'b', 4, 5, 6, 7, 8, 2, (5, 4), 13, 3}
```

REMOVING ITEMS

Both the discard() and remove() functions take a single value as an argument and removes that value from the set. If that value is not present, discard() does nothing, but remove() will raise a KeyError exception.

```
>> myset.discard(10)
>> myset
{'a', 1, 'c', 'b', 4, 5, 7, 8, 2, 12, (5, 4), 13, 11, 3}
>> myset.remove(13)
>> myset
{'a', 1, 'c', 'b', 4, 5, 7, 8, 2, 12, (5, 4), 11, 3}
```

COMMON SET OPERATIONS

Using union(), intersection() and difference() functions.

```
>> a = {2, 4, 5, 9}
>> b = {2, 4, 11, 12}
>> a.union(b) # Values which exist in a or b
{2, 4, 5, 9, 11, 12}
>> a.intersection(b) # Values which exist in a and b
{2, 4}
>> a.difference(b) # Values which exist in a but not in b
{9, 5}
```

The union() and intersection() functions are symmetric methods:

```
>> a.union(b) == b.union(a)
True
>> a.intersection(b) == b.intersection(a)
True
>> a.difference(b) == b.difference(a)
False
```

These [other built-in data structures in Python](#) are also useful.

Task

Given **2** sets of integers, **M** and **N**, print their symmetric difference in ascending order. The term symmetric difference indicates those values that exist in either **M** or **N** but do not exist in both.

Input Format

<https://www.hackerrank.com/challenges/symmetric-difference/problem>

The first line of input contains an integer, M .

The second line contains M space-separated integers.

The third line contains an integer, N .

The fourth line contains N space-separated integers.

Output Format

Output the symmetric difference integers in ascending order, one per line.

Sample Input

STDIN	Function
-----	-----
4	set a size M = 4
2 4 5 9	a = {2, 4, 5, 9}
4	set b size N = 4
2 4 11 12	b = {2, 4, 11, 12}

Sample Output

5
9
11
12





Set .add() ★

91/115 challenges solved

Rank: 10361 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



If we want to add a single element to an existing set, we can use the `.add()` operation.

It adds the element to the set and returns 'None'.

Example

```
>>> s = set('HackerRank')
>>> s.add('H')
>>> print s
set(['a', 'c', 'e', 'H', 'k', 'n', 'r', 'R'])
>>> print s.add('HackerRank')
None
>>> print s
set(['a', 'c', 'e', 'HackerRank', 'H', 'k', 'n', 'r', 'R'])
```

Task

Apply your knowledge of the `.add()` operation to help your friend Rupal.

Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of N country stamps.

Find the total number of distinct country stamps.

Input Format

The first line contains an integer N , the total number of country stamps.

The next N lines contains the name of the country where the stamp is from.

Constraints

$0 < N < 1000$

Output Format

Output the total number of distinct country stamps on a single line.

Sample Input

```
7
UK
China
USA
France
New Zealand
```

```
UK
```

France

Sample Output

5

Explanation

UK and France repeat twice. Hence, the total number of distinct country stamps is **5** (five).

[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code



91/115 challenges solved

Rank: 10361 | Points: 1855 !



Set .discard(), .remove() & .pop() ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



.remove(x)

This operation removes element **x** from the set.

If element **x** does not exist, it raises a `KeyError`.

The `.remove(x)` operation returns `None`.

Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.remove(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.remove(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.remove(0)
KeyError: 0
```

.discard(x)

This operation also removes element **x** from the set.

If element **x** does not exist, it **does not** raise a `KeyError`.

The `.discard(x)` operation returns `None`.

Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.discard(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.discard(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.discard(0)
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
```

.pop()

This operation removes and return an arbitrary element from the set.

If there are no elements to remove, it raises a `KeyError`.

Example

```
>>> s = set([1])
>>> print s.pop()
1
>>> print s
set([])
>>> print s.pop()
KeyError: pop from an empty set
```

Task

You have a non-empty set s , and you have to execute N commands given in N lines.

The commands will be pop, remove and discard.

Input Format

The first line contains integer n , the number of elements in the set s .

The second line contains n space separated elements of set s . All of the elements are non-negative integers, less than or equal to 9.

The third line contains integer N , the number of commands.

The next N lines contains either pop, remove and/or discard commands followed by their associated value.

Constraints

$0 < n < 20$

$0 < N < 20$

Output Format

Print the sum of the elements of set s on a single line.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
10
pop
remove 9
discard 9
discard 8
remove 7
pop
discard 6
remove 5
pop
discard 5
```

Sample Output

```
4
```

Explanation

After completing these **10** operations on the set, we get set([4]). Hence, the sum is **4**.

Note: Convert the elements of set s to integers while you are assigning them. To ensure the proper input of the set, we have added the first two lines of code to the editor.

Line: 1 Col: 70

 [Upload Code as File](#) [Test against custom input](#)[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



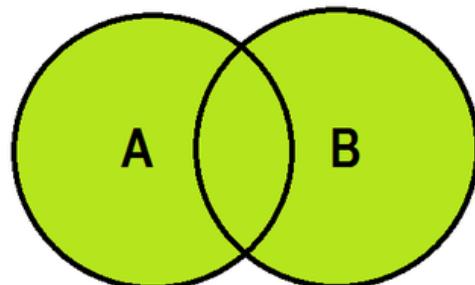
Set .union() Operation ★

91/115 challenges solved

Rank: 10361 | Points: 1855 ⓘ

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE

**A.union(B) or A|B**

BY DOSHI

.union()

The .union() operator returns the union of a set and the set of elements in an iterable.

Sometimes, the | operator is used in place of .union() operator, but it operates only on the set of elements in set.

Set is immutable to the .union() operation (or | operation).

Example

```
>>> s = set("Hacker")
>>> print s.union("Rank")
set(['a', 'R', 'c', 'r', 'e', 'H', 'k', 'n'])

>>> print s.union(set(['R', 'a', 'n', 'k']))
set(['a', 'R', 'c', 'r', 'e', 'H', 'k', 'n'])

>>> print s.union(['R', 'a', 'n', 'k'])
set(['a', 'R', 'c', 'r', 'e', 'H', 'k', 'n'])

>>> print s.union(enumerate(['R', 'a', 'n', 'k']))
set(['a', 'c', 'r', 'e', (1, 'a'), (2, 'n'), 'H', 'k', (3, 'k'), (0, 'R')])

>>> print s.union({"Rank":1})
set(['a', 'c', 'r', 'e', 'H', 'k', 'Rank'])

>>> s | set("Rank")
set(['a', 'R', 'c', 'r', 'e', 'H', 'k', 'n'])
```

Task

The students of District College have subscriptions to English and French newspapers. Some students have subscribed only to English, some have subscribed to only French and some have subscribed to both newspapers.

You are given two sets of student roll numbers. One set has subscribed to the English newspaper, and the other set is subscribed to the French newspaper. The same student could be in both sets. Your task is to find the total number of students who have subscribed to at least one newspaper.

Input Format

The first line contains an integer, **n**, the number of students who have subscribed to the English newspaper.

The second line contains **n** space separated roll numbers of those students.

The third line contains **b**, the number of students who have subscribed to the French newspaper.

The fourth line contains **b** space separated roll numbers of those students.

Constraints

0 < Total number of students in college < 1000

Output Format

Output the total number of students who have at least one subscription.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
9
10 1 2 3 11 21 55 6 8
```

Sample Output

```
13
```

Explanation

Roll numbers of students who have at least one subscription:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 21 and **55**. Roll numbers: **1, 2, 3, 6** and **8** are in both sets so they are only counted once.

Hence, the total is **13** students.

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

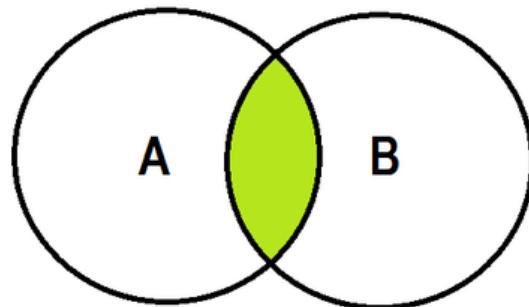
 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)

Set .intersection() Operation ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



A.intersection(B) or A&B

By DOSHI

[.intersection\(\)](#)

The .intersection() operator returns the intersection of a set and the set of elements in an iterable.

Sometimes, the & operator is used in place of the .intersection() operator, but it only operates on the set of elements in set.

The set is immutable to the .intersection() operation (or & operation).

```
>>> s = set("Hacker")
>>> print s.intersection("Rank")
set(['a', 'k'])

>>> print s.intersection(set(['R', 'a', 'n', 'k']))
set(['a', 'k'])

>>> print s.intersection(['R', 'a', 'n', 'k'])
set(['a', 'k'])

>>> print s.intersection(enumerate(['R', 'a', 'n', 'k']))
set([])

>>> print s.intersection({"Rank":1})
set([])

>>> s & set("Rank")
set(['a', 'k'])
```

Task

The students of District College have subscriptions to English and French newspapers. Some students have subscribed only to English, some have

<https://www.hackerrank.com/challenges/py-set-intersection-operation/problem>

subscribed only to French, and some have subscribed to both newspapers.

You are given two sets of student roll numbers. One set has subscribed to the English newspaper, one set has subscribed to the French newspaper. Your task is to find the total number of students who have subscribed to both newspapers.

Input Format

The first line contains **n**, the number of students who have subscribed to the English newspaper.

The second line contains **n** space separated roll numbers of those students.

The third line contains **b**, the number of students who have subscribed to the French newspaper.

The fourth line contains **b** space separated roll numbers of those students.

Constraints

0 < Total number of students in college < 1000

Output Format

Output the total number of students who have subscriptions to **both** English and French newspapers.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
9
10 1 2 3 11 21 55 6 8
```

Sample Output

```
5
```

Explanation

The roll numbers of students who have both subscriptions:

1, 2, 3, 6 and 8.

Hence, the total is **5** students.





91/115 challenges solved

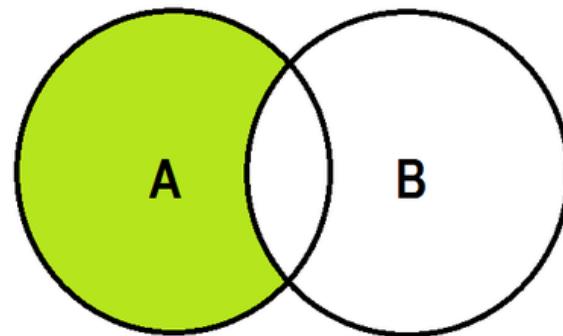
Rank: 10361 | Points: 1855



Set .difference() Operation ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE

**A.difference(B) or A - B**

By DOSHI

.difference()

The tool `.difference()` returns a set with all the elements from the set that are not in an iterable.

Sometimes the `-` operator is used in place of the `.difference()` tool, but it only operates on the set of elements in set.

Set is immutable to the `.difference()` operation (or the `-` operation).

```
>>> s = set("Hacker")
>>> print s.difference("Rank")
set(['c', 'r', 'e', 'H'])

>>> print s.difference(set(['R', 'a', 'n', 'k']))
set(['c', 'r', 'e', 'H'])

>>> print s.difference(['R', 'a', 'n', 'k'])
set(['c', 'r', 'e', 'H'])

>>> print s.difference(enumerate(['R', 'a', 'n', 'k']))
set(['a', 'c', 'r', 'e', 'H', 'k'])

>>> print s.difference({"Rank":1})
set(['a', 'c', 'e', 'H', 'k', 'r'])

>>> s - set("Rank")
set(['H', 'c', 'r', 'e'])
```

Task

Students of District College have a subscription to English and French newspapers. Some students have subscribed to only the English newspaper, some have subscribed to only the French newspaper, and some have subscribed to both newspapers.

You are given two sets of student roll numbers. One set has subscribed to the English newspaper, and one set has subscribed to the French newspaper. Your task is to find the total number of students who have subscribed to only English newspapers.

Input Format

The first line contains the number of students who have subscribed to the English newspaper.

The second line contains the space separated list of student roll numbers who have subscribed to the English newspaper.

The third line contains the number of students who have subscribed to the French newspaper.

The fourth line contains the space separated list of student roll numbers who have subscribed to the French newspaper.

Constraints

$0 < \text{Total number of students in college} < 1000$

Output Format

Output the total number of students who are subscribed to the English newspaper only.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
9
10 1 2 3 11 21 55 6 8
```

Sample Output

```
4
```

Explanation

The roll numbers of students who only have English newspaper subscriptions are:

4, 5, 7 and 9.

Hence, the total is **4** students.

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



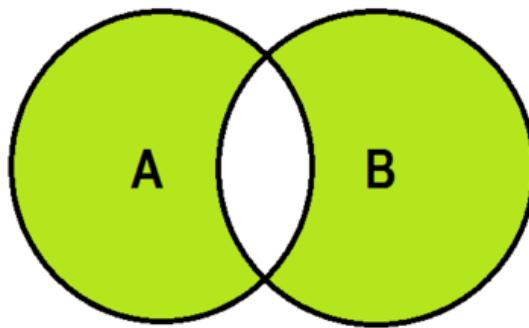
Set .symmetric_difference() Operation ★

91/115 challenges solved

Rank: 10361 | Points: 1855

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



A.symmetric_difference(B) or A^B

by DOSHI

.symmetric_difference()

The `.symmetric_difference()` operator returns a set with all the elements that are in the set and the iterable but not both.

Sometimes, a `^` operator is used in place of the `.symmetric_difference()` tool, but it only operates on the set of elements in set.

The set is immutable to the `.symmetric_difference()` operation (or `^` operation).

```
>>> s = set("Hacker")
>>> print s.symmetric_difference("Rank")
set(['c', 'e', 'H', 'n', 'R', 'r'])

>>> print s.symmetric_difference(set(['R', 'a', 'n', 'k']))
set(['c', 'e', 'H', 'n', 'R', 'r'])

>>> print s.symmetric_difference(['R', 'a', 'n', 'k'])
set(['c', 'e', 'H', 'n', 'R', 'r'])

>>> print s.symmetric_difference(enumerate(['R', 'a', 'n', 'k']))
set(['a', 'c', 'e', 'H', (0, 'R'), 'r', (2, 'n'), 'k', (1, 'a'), (3, 'k')])

>>> print s.symmetric_difference({"Rank":1})
set(['a', 'c', 'e', 'H', 'k', 'Rank', 'r'])

>>> s ^ set("Rank")
set(['c', 'e', 'H', 'n', 'R', 'r'])
```

Task

Students of District College have subscriptions to English and French newspapers. Some students have subscribed to English only, some have subscribed to French only, and some have subscribed to both newspapers.

You are given two sets of student roll numbers. One set has subscribed to the English newspaper, and one set has subscribed to the French newspaper. Your task is to find the total number of students who have subscribed to either the English or the French newspaper but not both.

Input Format

The first line contains the number of students who have subscribed to the English newspaper.

The second line contains the space separated list of student roll numbers who have subscribed to the English newspaper.

The third line contains the number of students who have subscribed to the French newspaper.

The fourth line contains the space separated list of student roll numbers who have subscribed to the French newspaper.

Constraints

0 < Total number of students in college < 1000

Output Format

Output total number of students who have subscriptions to the English or the French newspaper but not both.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
9
10 1 2 3 11 21 55 6 8
```

Sample Output

```
8
```

Explanation

The roll numbers of students who have subscriptions to English or French newspapers but not both are:

4, 5, 7, 9, 10, 11, 21 and **55**.

Hence, the total is **8** students.

[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



91/115 challenges solved

Rank: 10361 | Points: 1855 !



Set Mutations ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



We have seen the applications of union, intersection, difference and symmetric difference operations, but these operations do not make any changes or mutations to the set.

We can use the following operations to create mutations to a set:

.update() or |=

Update the set by adding elements from an iterable/another set.

```
>>> H = set("Hacker")
>>> R = set("Rank")
>>> H.update(R)
>>> print H
set(['a', 'c', 'e', 'H', 'k', 'n', 'r', 'R'])
```

.intersection_update() or &=

Update the set by keeping only the elements found in it and an iterable/another set.

```
>>> H = set("Hacker")
>>> R = set("Rank")
>>> H.intersection_update(R)
>>> print H
set(['a', 'k'])
```

.difference_update() or -=

Update the set by removing elements found in an iterable/another set.

```
>>> H = set("Hacker")
>>> R = set("Rank")
>>> H.difference_update(R)
>>> print H
set(['c', 'e', 'H', 'r'])
```

.symmetric_difference_update() or ^=

Update the set by only keeping the elements found in either set, but not in both.

```
>>> H = set("Hacker")
>>> R = set("Rank")
>>> H.symmetric_difference_update(R)
>>> print H
set(['c', 'e', 'H', 'n', 'r', 'R'])
```

TASK

You are given a set A and N number of other sets. These N number of sets have to perform some specific mutation operations on set A .

Your task is to execute those operations and print the sum of elements from set A .

Input Format

The first line contains the number of elements in set A .

The second line contains the space separated list of elements in set A .

The third line contains integer N , the number of other sets.

The next $2 * N$ lines are divided into N parts containing two lines each.

The first line of each part contains the space separated entries of the operation name and the length of the other set.

The second line of each part contains space separated list of elements in the other set.

$0 < \text{len}(\text{set}(A)) < 1000$

$0 < \text{len}(\text{otherSets}) < 100$

$0 < N < 100$

Output Format

Output the sum of elements in set A .

Sample Input

```
16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 24 52
4
intersection_update 10
2 3 5 6 8 9 1 4 7 11
update 2
55 66
symmetric_difference_update 5
22 7 35 62 58
difference_update 7
11 22 35 55 58 62 66
```

Sample Output

38

Explanation

After the first operation, (intersection_update operation), we get:

set $A = \text{set}([1, 2, 3, 4, 5, 6, 7, 8, 9, 11])$

After the second operation, (update operation), we get:

set $A = \text{set}([1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 55, 66])$

After the third operation, (symmetric_difference_update operation), we get:

set $A = \text{set}([1, 2, 3, 4, 5, 6, 8, 9, 11, 22, 35, 55, 58, 62, 66])$

After the fourth operation, (difference_update operation), we get:

set $A = \text{set}([1, 2, 3, 4, 5, 6, 8, 9])$

The sum of elements in set A after these operations is **38**.

Line: 1 Col: 70

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

The Captain's Room ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Mr. Anant Asankhya is the manager at the INFINITE hotel. The hotel has an infinite amount of rooms.

One fine day, a finite number of tourists come to stay at the hotel.

The tourists consist of:

- A Captain.
- An unknown group of families consisting of K members per group where $K \neq 1$.

The Captain was given a separate room, and the rest were given one room per group.

Mr. Anant has an unordered list of randomly arranged room entries. The list consists of the room numbers for all of the tourists. The room numbers will appear K times per group except for the Captain's room.

Mr. Anant needs you to help him find the Captain's room number.

The total number of tourists or the total number of groups of families is not known to you.

You only know the value of K and the room number list.

Input Format

The first line consists of an integer, K , the size of each group.

The second line contains the unordered elements of the room number list.

Constraints

$1 < K < 1000$

Output Format

Output the Captain's room number.

Sample Input

```
5
1 2 3 6 5 4 4 2 5 3 6 1 6 5 3 2 4 1 2 5 1 4 3 6 8 4 3 1 5 6 2
```

Sample Output

```
8
```

Explanation

The list of room numbers contains 31 elements. Since K is 5, there must be 6 groups of families. In the given list, all of the numbers repeat 5 times except for room number 8.

Hence, 8 is the Captain's room number.



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Check Subset ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given two sets, **A** and **B**.

Your job is to find whether set **A** is a subset of set **B**.

If set **A** is subset of set **B**, print **True**.

If set **A** is not a subset of set **B**, print **False**.

Input Format

The first line will contain the number of test cases, **T**.

The first line of each test case contains the number of elements in set **A**.

The second line of each test case contains the space separated elements of set **A**.

The third line of each test case contains the number of elements in set **B**.

The fourth line of each test case contains the space separated elements of set **B**.

Constraints

- $0 < T < 21$
- $0 < \text{Number of elements in each set} < 1001$

Output Format

Output **True** or **False** for each test case on separate lines.

Sample Input

```
3
5
1 2 3 5 6
9
9 8 5 6 3 2 1 4 7
1
2
5
3 6 5 4 1
7
1 2 3 5 6 8 9
3
9 8 2
```

Sample Output

```
True
False
False
```

Explanation

Test Case 01 Explanation

Set $A = \{1 2 3 5 6\}$

Set $B = \{9 8 5 6 3 2 1 4 7\}$

All the elements of set A are elements of set B .

Hence, set A is a subset of set B .

Change Theme Language Pypy 2



Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code



Check Strict Superset ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given a set A and n other sets.

Your job is to find whether set A is a strict superset of each of the N sets.

Print True, if A is a strict superset of each of the N sets. Otherwise, print False.

A strict superset has at least one element that does not exist in its subset.

Example

Set([1, 3, 4]) is a strict superset of set([1, 3]).

Set([1, 3, 4]) is not a strict superset of set([1, 3, 4]).

Set([1, 3, 4]) is not a strict superset of set([1, 3, 5]).

Input Format

The first line contains the space separated elements of set A .

The second line contains integer n , the number of other sets.

The next n lines contains the space separated elements of the other sets.

Constraints

- $0 < \text{len}(\text{set}(A)) < 501$
- $0 < N < 21$
- $0 < \text{len}(\text{otherSets}) < 101$

Output Format

Print True if set A is a strict superset of all other N sets. Otherwise, print False.

Sample Input 0

```
1 2 3 4 5 6 7 8 9 10 11 12 23 45 84 78
2
1 2 3 4 5
100 11 12
```

Sample Output 0

```
False
```

Explanation 0

Set A is the strict superset of the set([1, 2, 3, 4, 5]) but not of the set([100, 11, 12]) because 100 is not in set A .

Hence, the output is False.

[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

 [Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



collections.Counter()

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE

[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.

Input Format

The first line contains X , the number of shoes.

The second line contains the space separated list of all the shoe sizes in the shop.

The third line contains N , the number of customers.

The next N lines contain the space separated values of the **shoe size** desired by the customer and x_i , the price of the shoe.

Constraints

$0 < X < 10^3$

$0 < N \leq 10^3$

$20 < x_i < 100$

$2 < \text{shoe size} < 20$

Output Format

Print the amount of money earned by **Raghu**.

Sample Input

<https://www.hackerrank.com/challenges/collections-counter/problem>

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

Sample Output

200

Explanation

Customer 1: Purchased size 6 shoe for **\$55**.

Customer 2: Purchased size 6 shoe for **\$45**.

Customer 3: Size 6 no longer available, so no purchase.

Customer 4: Purchased size 4 shoe for **\$40**.

Customer 5: Purchased size 18 shoe for **\$60**.

Customer 6: Size 10 not available, so no purchase.

Total money earned = **55 + 45 + 40 + 60 = \$200**





91/115 challenges solved

Rank: 10362 | Points: 1855 !



DefaultDict Tutorial ★

Problem

Submissions

Leaderboard

Editorial

RATE THIS CHALLENGE



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**.

Example

Group A contains 'a', 'b', 'a' Group B contains 'a', 'c'

For the first word in group B, 'a', it appears at positions **1** and **3** in group A. The second word, 'c', does not appear in group A, so print **-1**.

Expected output:

```
1 3
-1
```

Input Format

The first line contains integers, ***n*** and ***m*** separated by a space.

The next ***n*** lines contains the words belonging to group **A**.

The next ***m*** lines contains the words belonging to group **B**.

Constraints

$1 \leq n \leq 10000$

$1 \leq m \leq 100$

$1 < \text{length of each word in the input} < 100$

Output Format

Output m lines.

The i^{th} line should contain the 1-indexed positions of the occurrences of the i^{th} word separated by spaces.

Sample Input

```
STDIN    Function
-----  -----
5 2      group A size n = 5, group B size m = 2
a        group A contains 'a', 'a', 'b', 'a', 'b'
a
b
a
b
a        group B contains 'a', 'b'
b
```

Sample Output

```
1 2 4
3 5
```

Explanation

'a' appeared 3 times in positions 1, 2 and 4.

'b' appeared 2 times in positions 3 and 5.

In the sample problem, if 'c' also appeared in word group B , you would print -1.





91/115 challenges solved

Rank: 10362 | Points: 1855

Collections.namedtuple()

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



[collections.namedtuple\(\)](#)

Basically, namedtuples are easy to create, lightweight object types.

They turn tuples into convenient containers for simple tasks.

With namedtuples, you don't have to use integer indices for accessing members of a tuple.

Example

Code 01

```
>>> from collections import namedtuple
>>> Point = namedtuple('Point','x,y')
>>> pt1 = Point(1,2)
>>> pt2 = Point(3,4)
>>> dot_product = ( pt1.x * pt2.x ) +( pt1.y * pt2.y )
>>> print dot_product
11
```

Code 02

```
>>> from collections import namedtuple
>>> Car = namedtuple('Car','Price Mileage Colour Class')
>>> xyz = Car(Price = 100000, Mileage = 30, Colour = 'Cyan', Class = 'Y')
>>> print xyz
Car(Price=100000, Mileage=30, Colour='Cyan', Class='Y')
>>> print xyz.Class
Y
```

Task

Dr. John Wesley has a spreadsheet containing a list of student's **IDs, marks, class** and **name**.

Your task is to help Dr. Wesley calculate the average marks of the students.

$$\text{Average} = \frac{\text{Sum of all marks}}{\text{Total Students}}$$

Note:

1. Columns can be in any order. IDs, marks, class and name can be written in any order in the spreadsheet.

2. Column names are ID, MARKS, CLASS and NAME. (The spelling and case type of these names won't change.)

Input Format

The first line contains an integer **N**, the total number of students.

The second line contains the names of the columns in any order.

The next **N** lines contains the **marks, IDs, name** and **class**, under their respective column names.

Constraints $0 < N \leq 100$ **Output Format**

Print the average marks of the list corrected to 2 decimal places.

Sample Input**TESTCASE 01**

5			
ID	MARKS	NAME	CLASS
1	97	Raymond	7
2	50	Steven	4
3	91	Adrian	9
4	72	Stewart	5
5	80	Peter	6

TESTCASE 02

5			
MARKS	CLASS	NAME	ID
92	2	Calum	1
82	5	Scott	2
94	2	Jason	3
55	8	Glenn	4
82	2	Fergus	5

Sample Output**TESTCASE 01**

78.00

TESTCASE 02

81.00

Explanation**TESTCASE 01**Average = $(97 + 50 + 91 + 72 + 80)/5$

Can you solve this challenge in 4 lines of code or less?

NOTE: There is no penalty for solutions that are correct but have more than 4 lines.[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

 Upload Code as File

Test against custom input

 Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Collections.OrderedDict() ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



collections.OrderedDict

An OrderedDict is a dictionary that remembers the order of the keys that were inserted first. If a new entry overwrites an existing entry, the original insertion position is left unchanged.

Example

Code

```
>>> from collections import OrderedDict
>>>
>>> ordinary_dictionary = {}
>>> ordinary_dictionary['a'] = 1
>>> ordinary_dictionary['b'] = 2
>>> ordinary_dictionary['c'] = 3
>>> ordinary_dictionary['d'] = 4
>>> ordinary_dictionary['e'] = 5
>>>
>>> print ordinary_dictionary
{'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4}
>>>
>>> ordered_dictionary = OrderedDict()
>>> ordered_dictionary['a'] = 1
>>> ordered_dictionary['b'] = 2
>>> ordered_dictionary['c'] = 3
>>> ordered_dictionary['d'] = 4
>>> ordered_dictionary['e'] = 5
>>>
>>> print ordered_dictionary
OrderedDict([('a', 1), ('b', 2), ('c', 3), ('d', 4), ('e', 5)])
```

Task

You are the manager of a supermarket.

You have a list of N items together with their prices that consumers bought on a particular day.

Your task is to print each item_name and net_price in order of its first occurrence.

item_name = Name of the item.

net_price = Quantity of the item sold multiplied by the price of each item.

Input Format

The first line contains the number of items, N .

The next N lines contains the item's name and price, separated by a space.

Constraints

$0 < N \leq 100$

Output Format

Print the item_name and net_price in order of its first occurrence.

Sample Input

```
9
BANANA FRIES 12
POTATO CHIPS 30
APPLE JUICE 10
CANDY 5
APPLE JUICE 10
CANDY 5
CANDY 5
CANDY 5
POTATO CHIPS 30
```

Sample Output

```
BANANA FRIES 12
POTATO CHIPS 60
APPLE JUICE 20
CANDY 20
```

Explanation

BANANA FRIES: Quantity bought: **1**, Price: **12**

Net Price: **12**

POTATO CHIPS: Quantity bought: **2**, Price: **30**

Net Price: **60**

APPLE JUICE: Quantity bought: **2**, Price: **10**

Net Price: **20**

CANDY: Quantity bought: **4**, Price: **5**

Net Price: **20**





Word Order ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given n words. Some words may repeat. For each word, output its number of occurrences. The output order should correspond with the input order of appearance of the word. See the sample input/output for clarification.

Note: Each input line ends with a "\n" character.

Constraints:

$$1 \leq n \leq 10^5$$

The sum of the lengths of all the words do not exceed 10^6

All the words are composed of lowercase English letters only.

Input Format

The first line contains the integer, n .

The next n lines each contain a word.

Output Format

Output **2** lines.

On the first line, output the number of distinct words from the input.

On the second line, output the number of occurrences for each distinct word according to their appearance in the input.

Sample Input

```
4
bcdef
abcdefg
bcde
bcdef
```

Sample Output

```
3
2 1 1
```

Explanation

There are **3** distinct words. Here, "**bcdef**" appears twice in the input at the first and last positions. The other words appear once each. The order of the first appearances are "**bcdef**", "**abcdefg**" and "**bcde**" which corresponds to the output.





[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Collections.deque()

91/115 challenges solved

Rank: 10362 | Points: 1855

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



[collections.deque\(\)](#)

A deque is a double-ended queue. It can be used to add or remove elements from both ends.

Deques support thread safe, memory efficient appends and pops from either side of the deque with approximately the same $O(1)$ performance in either direction.

Click on the link to learn more about **deque() methods**.

Click on the link to learn more about various approaches to working with deques: **Deque Recipes**.

Example

[Code](#)

```
>>> from collections import deque
>>> d = deque()
>>> d.append(1)
>>> print d
deque([1])
>>> d.appendleft(2)
>>> print d
deque([2, 1])
>>> d.clear()
>>> print d
deque([])
>>> d.extend('1')
>>> print d
deque(['1'])
>>> d.extendleft('234')
>>> print d
deque(['4', '3', '2', '1'])
>>> d.count('1')
1
>>> d.pop()
'1'
>>> print d
deque(['4', '3', '2'])
>>> d.popleft()
'4'
>>> print d
deque(['3', '2'])
>>> d.extend('7896')
>>> print d
deque(['3', '2', '7', '8', '9', '6'])
>>> d.remove('2')
>>> print d
deque(['3', '7', '8', '9', '6'])
>>> d.reverse()
>>> print d
deque(['6', '9', '8', '7', '3'])
```

```
>>> d.rotate(3)
>>> print d
deque(['8', '7', '3', '6', '9'])
```

Task

Perform append, pop, popleft and appendleft methods on an empty deque **d**.

Input Format

The first line contains an integer **N**, the number of operations.

The next **N** lines contains the space separated names of methods and their values.

Constraints

$0 < N \leq 100$

Output Format

Print the space separated elements of deque **d**.

Sample Input

```
6
append 1
append 2
append 3
appendleft 4
pop
popleft
```

Sample Output

```
1 2
```

[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)

[Prepare](#) > [Python](#) > [Collections](#) > [Company Logo](#)

Company Logo ★

[Problem](#)
[Submissions](#)
[Leaderboard](#)
[Discussions](#)
[Editorial](#)

A newly opened multinational brand has decided to base their company logo on the three most common characters in the company name. They are now trying out various combinations of company names and logos based on this condition. Given a string s , which is the company name in lowercase letters, your task is to find the top three most common characters in the string.

- Print the three most common characters along with their occurrence count.
- Sort in descending order of occurrence count.
- If the occurrence count is the same, sort the characters in alphabetical order.

For example, according to the conditions described above,

GOOGLE would have its logo with the letters **G, O, E**.

Input Format

A single line of input containing the string S .

Constraints

- $3 < \text{len}(S) \leq 10^4$
- S has at least 3 distinct characters

Output Format

Print the three most common characters along with their occurrence count each on a separate line.

Sort output in descending order of occurrence count.

If the occurrence count is the same, sort the characters in alphabetical order.

Sample Input 0

```
aabbccde
```

Sample Output 0

```
b 3
a 2
c 2
```

Explanation 0

aabbbccde****

Here, b occurs 3 times. It is printed first.

Both a and c occur 2 times. So, a is printed in the second line and c in the third line because a comes before c in the alphabet.

Note: The string S has at least 3 distinct characters.

Author [deleted]

Difficulty Medium

Max Score 30

Submitted By 57691

NEED HELP?

[View discussions](#)

[View editorial](#)

[View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Piling Up! ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



There is a horizontal row of **n** cubes. The length of each cube is given. You need to create a new vertical pile of cubes. The new pile should follow these directions: if **cube[i]** is on top of **cube[j]** then **sideLength[j] ≥ sideLength[i]**.

When stacking the cubes, you can only pick up either the leftmost or the rightmost cube each time. Print Yes if it is possible to stack the cubes. Otherwise, print No.

Example

blocks = [1, 2, 3, 8, 7]

Result: No

After choosing the rightmost element, **7**, choose the leftmost element, **1**. After than, the choices are **2** and **8**. These are both larger than the top block of size **1**.

blocks = [1, 2, 3, 7, 8]

Result: Yes

Choose blocks from right to left in order to successfully stack the blocks.

Input Format

The first line contains a single integer **T**, the number of test cases.

For each test case, there are **2** lines.

The first line of each test case contains **n**, the number of cubes.

The second line contains **n** space separated integers, denoting the sideLengths of each cube in that order.

Constraints

1 ≤ T ≤ 5

1 ≤ n ≤ 10⁵

1 ≤ sideLength < 2³¹

Output Format

For each test case, output a single line containing either Yes or No.

Sample Input

STDIN	Function
-----	-----
2	T = 2
6	blocks[] size n = 6
4 3 2 1 3 4	blocks = [4, 3, 2, 1, 3, 4]
3	blocks[] size n = 3
1 3 2	blocks = [1, 3, 2]

Sample Output

<https://www.hackerrank.com/challenges/piling-up/problem>

Yes
No

Explanation

In the first test case, pick in this order: **left - 4, right - 4, left - 3, right - 3, left - 2, right - 1**.

In the second test case, no order gives an appropriate arrangement of vertical cubes. **3** will always come after either **1** or **2**.



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Calendar Module ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !



Problem

Submissions

Leaderboard

Editorial

RATE THIS CHALLENGE



Calendar Module

The calendar module allows you to output calendars and provides additional useful functions for them.

```
class calendar.TextCalendar([firstweekday])
```

This class can be used to generate plain text calendars.

Sample Code

```
>>> import calendar
>>>
>>> print calendar.TextCalendar(firstweekday=6).formatyear(2015)
2015
```

January						
Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

February						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

March						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

April						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4		
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

May						
Su	Mo	Tu	We	Th	Fr	Sa
			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
	31					

June						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

July						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4		
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

August						
Su	Mo	Tu	We	Th	Fr	Sa
			1			
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

September						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

October						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

November						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

To learn more about different calendar functions, [click here](#).

Task

You are given a date. Your task is to find what the day is on that date.

Input Format

A single line of input containing the space separated month, day and year, respectively, in ***MM DD YYYY*** format.

Constraints

- ***2000 < year < 3000***

Output Format

Output the correct day in capital letters.

Sample Input

08 05 2015

Sample Output

WEDNESDAY

Explanation

The day on August 5th 2015 was WEDNESDAY.





Time Delta ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



When users post an update on social media, such as a URL, image, status update etc., other users in their network are able to view this new post on their news feed. Users can also see exactly when the post was published, i.e, how many hours, minutes or seconds ago.

Since sometimes posts are published and viewed in different time zones, this can be confusing. You are given two timestamps of one such post that a user can see on his newsfeed in the following format:

Day dd Mon yyyy hh:mm:ss +xxxx

Here +xxxx represents the time zone. Your task is to print the absolute difference (in seconds) between them.

Input Format

The first line contains **T**, the number of testcases.

Each testcase contains **2** lines, representing time **t₁** and time **t₂**.

Constraints

- Input contains only valid timestamps
- **year** ≤ 3000.

Output Format

Print the absolute difference (**t₁ – t₂**) in seconds.

Sample Input 0

```
2
Sun 10 May 2015 13:54:36 -0700
Sun 10 May 2015 13:54:36 -0000
Sat 02 May 2015 19:54:36 +0530
Fri 01 May 2015 13:54:36 -0000
```

Sample Output 0

```
25200
88200
```

Explanation 0

In the first query, when we compare the time in UTC for both the time stamps, we see a difference of 7 hours, which is **7 × 3600** seconds or **25200** seconds.

Similarly, in the second query, time difference is 5 hours and 30 minutes for time zone adjusting for that we have a difference of 1 day and 30 minutes. Or **24 × 3600 + 30 × 60 ⇒ 88200**



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Exceptions ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Exceptions

Errors detected during execution are called exceptions.

Examples:

ZeroDivisionError

This error is raised when the second argument of a division or modulo operation is zero.

```
>>> a = '1'  
>>> b = '0'  
>>> print int(a) / int(b)  
>>> ZeroDivisionError: integer division or modulo by zero
```

ValueError

This error is raised when a built-in operation or function receives an argument that has the right type but an inappropriate value.

```
>>> a = '1'  
>>> b = '#'  
>>> print int(a) / int(b)  
>>> ValueError: invalid literal for int() with base 10: '#'
```

To learn more about different built-in exceptions [click here](#).

Handling Exceptions

The statements try and except can be used to handle selected exceptions. A try statement may have more than one except clause to specify handlers for different exceptions.

```
#Code  
try:  
    print 1/0  
except ZeroDivisionError as e:  
    print "Error Code:",e
```

Output

Error Code: integer division or modulo by zero

Task

You are given two values **a** and **b**.

Perform integer division and print **a/b**.

Input Format

The first line contains **T**, the number of test cases.

The next T lines each contain the space separated values of a and b .

Constraints

- $0 < T < 10$

Output Format

Print the value of a/b .

In the case of ZeroDivisionError or ValueError, print the error code.

Sample Input

```
3
1 0
2 $
3 1
```

Sample Output

```
Error Code: integer division or modulo by zero
Error Code: invalid literal for int() with base 10: '$'
3
```

Note:

For integer division in **Python 3** use `/`.





Zipped! ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE

[zip\(\[iterable, ...\]\)](#)

This function returns a list of tuples. The i^{th} tuple contains the i^{th} element from each of the argument sequences or iterables.

If the argument sequences are of unequal lengths, then the returned list is truncated to the length of the shortest argument sequence.

Sample Code

```
>>> print zip([1,2,3,4,5,6],'Hacker')
[(1, 'H'), (2, 'a'), (3, 'c'), (4, 'k'), (5, 'e'), (6, 'r')]
>>>
>>> print zip([1,2,3,4,5,6],[0,9,8,7,6,5,4,3,2,1])
[(1, 0), (2, 9), (3, 8), (4, 7), (5, 6), (6, 5)]
>>>
>>> A = [1,2,3]
>>> B = [6,5,4]
>>> C = [7,8,9]
>>> X = [A] + [B] + [C]
>>>
>>> print zip(*X)
[(1, 6, 7), (2, 5, 8), (3, 4, 9)]
```

Task

The National University conducts an examination of N students in X subjects.

Your task is to compute the average scores of each student.

$$\text{Average score} = \frac{\text{Sum of scores obtained in all subjects by a student}}{\text{Total number of subjects}}$$

The format for the general mark sheet is:

Student ID →	1	2	3	4	5
Subject 1	89	90	78	93	80
Subject 2	90	91	85	88	86
Subject 3	91	92	83	89	90.5
Average -----	90	91	82	90	85.5

Input Format

The first line contains N and X separated by a space.

The next X lines contains the space separated marks obtained by students in a particular subject.

Constraints

$0 < N < 100$

<https://www.hackerrank.com/challenges/zipped/problem>

$0 < X \leq 100$

Output Format

Print the averages of all students on separate lines.

The averages must be correct up to **1** decimal place.

Sample Input

```
5 3
89 90 78 93 80
90 91 85 88 86
91 92 83 89 90.5
```

Sample Output

```
90.0
91.0
82.0
90.0
85.5
```

Explanation

Marks obtained by **student 1**: **89, 90, 91**

Average marks of **student 1**:

$$\mathbf{270/3 = 90}$$

Marks obtained by **student 2**: **90, 91, 92**

Average marks of **student 2**:

$$\mathbf{273/3 = 91}$$

Marks obtained by **student 3**: **78, 85, 83**

Average marks of **student 3**:

$$\mathbf{246/3 = 82}$$

Marks obtained by **student 4**: **93, 88, 89**

Average marks of **student 4**:

$$\mathbf{270/3 = 90}$$

Marks obtained by **student 5**: **80, 86, 90.5**

Average marks of **student 5**:

$$\mathbf{256.5/3 = 85.5}$$



Athlete Sort ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given a spreadsheet that contains a list of N athletes and their details (such as age, height, weight and so on). You are required to sort the data based on the K^{th} attribute and print the final resulting table. Follow the example given below for better understanding.

Rank	Age	Height (in cm)	Rank	Age	Height (in cm)	
1	32	190	5	24	176	
2	35	175	4	26	195	
3	41	188	—————>	1	32	190
4	26	195	i.e (age)	2	35	175
5	24	176		3	41	188

Note that K is indexed from 0 to $M - 1$, where M is the number of attributes.

Note: If two attributes are the same for different rows, for example, if two athletes are of the same age, print the row that appeared first in the input.

Input Format

The first line contains N and M separated by a space.

The next N lines each contain M elements.

The last line contains K .

Constraints

$1 \leq N, M \leq 1000$

$0 \leq K < M$

Each element ≤ 1000

Output Format

Print the N lines of the sorted table. Each line should contain the space separated elements. Check the sample below for clarity.

Sample Input 0

```
5 3
10 2 5
7 1 0
9 9 9
1 23 12
6 5 9
1
```

Sample Output 0

```
7 1 0
10 2 5
6 5 9
9 9 9
1 23 12
```

Explanation 0

The details are sorted based on the second attribute, since K is zero-indexed.



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

**ginorts ★**

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE

You are given a string **S**.**S** contains alphanumeric characters only.

Sorting

Your task is to sort the string **S** in the following manner:

- All sorted lowercase letters are ahead of uppercase letters.
- All sorted uppercase letters are ahead of digits.
- All sorted odd digits are ahead of sorted even digits.

Input Format

A single line of input contains the string **S**.

Constraints

- $0 < \text{len}(S) < 1000$

Output Format

Output the sorted string **S**.

Sample Input

Sorting1234

Sample Output

ginorts1324





Map and Lambda Function ★

91/115 challenges solved

Rank: 10362 | Points: 1855

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Let's learn some new Python concepts! You have to generate a list of the first **N** fibonacci numbers, **0** being the first number. Then, apply the map function and a lambda expression to cube each fibonacci number and print the list.

Concept

The `map()` function applies a function to every member of an iterable and returns the result. It takes two parameters: first, the function that is to be applied and secondly, the iterables.

Let's say you are given a list of names, and you have to print a list that contains the length of each name.

```
>> print (list(map(len, ['Tina', 'Raj', 'Tom'])))
[4, 3, 3]
```

Lambda is a single expression anonymous function often used as an inline function. In simple words, it is a function that has only one line in its body. It proves very handy in functional and GUI programming.

```
>> sum = lambda a, b, c: a + b + c
>> sum(1, 2, 3)
6
```

Note:

Lambda functions cannot use the `return` statement and can only have a single expression. Unlike `def`, which creates a function and assigns it a name, `lambda` creates a function and returns the function itself. Lambda can be used inside lists and dictionaries.

Input Format

One line of input: an integer **N**.

Constraints

$0 \leq N \leq 15$

Output Format

A list on a single line containing the cubes of the first **N** fibonacci numbers.

Sample Input

5

Sample Output

[0, 1, 1, 8, 27]

Explanation

<https://www.hackerrank.com/challenges/map-and-lambda-expression/problem>

The first **5** fibonacci numbers are [0, 1, 1, 2, 3], and their cubes are [0, 1, 1, 8, 27].

Change Theme Language Pypy 2

▼ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇



Line: 1 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Detect Floating Point Number ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

You are given a string N .

Your task is to verify that N is a floating point number.

In this task, a valid float number must satisfy all of the following requirements:

> Number can start with +, - or . symbol.

For example:

✓ +4.50

✓ -1.0

✓ .5

✓ -.7

✓ +.4

✗ -+4.5

> Number must contain at least 1 decimal value.

For example:

✗ 12.

✓ 12.0

> Number must have exactly one . symbol.

> Number must not give any exceptions when converted using $\text{float}(N)$.

Input Format

The first line contains an integer T , the number of test cases.

The next T line(s) contains a string N .

Constraints

- $0 < T < 10$

Output Format

Output True or False for each test case.

Sample Input 0

```
4
4.000
-1.00
+4.54
SomeRandomStuff
```

Sample Output 0

<https://www.hackerrank.com/challenges/introduction-to-regex/problem>

```
False  
True  
True  
False
```

Explanation 0

4.0O0: O is not a digit.

-1.00: is valid.

+4.54: is valid.

SomeRandomStuff: is not a number.



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Re.split()

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

RATE THIS CHALLENGE



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

You are given a string **s** consisting only of digits 0–9, commas , and dots .

Your task is to complete the `regex_pattern` defined below, which will be used to `re.split()` all of the , and . symbols in **s**.

It's guaranteed that every comma and every dot in **s** is preceded and followed by a digit.

Sample Input 0

```
100,000,000.000
```

Sample Output 0

```
100
000
000
000
```

[Change Theme](#) [Language](#) Pypy 2

Line: 1 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Group(), Groups() & Groupdict() ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



group()

A group() expression returns one or more subgroups of the match.

[Code](#)

```
>>> import re
>>> m = re.match(r'(\w+)@(\w+)\.(.\w+)', 'username@hackerrank.com')
>>> m.group(0)      # The entire match
'username@hackerrank.com'
>>> m.group(1)      # The first parenthesized subgroup.
'username'
>>> m.group(2)      # The second parenthesized subgroup.
'hackerrank'
>>> m.group(3)      # The third parenthesized subgroup.
'com'
>>> m.group(1,2,3)  # Multiple arguments give us a tuple.
('username', 'hackerrank', 'com')
```

groups()

A groups() expression returns a tuple containing all the subgroups of the match.

[Code](#)

```
>>> import re
>>> m = re.match(r'(\w+)@(\w+)\.(.\w+)', 'username@hackerrank.com')
>>> m.groups()
('username', 'hackerrank', 'com')
```

groupdict()

A groupdict() expression returns a dictionary containing all the named subgroups of the match, keyed by the subgroup name.

[Code](#)

```
>>> m = re.match(r'(?P<user>\w+)@(?P<website>\w+)\.(?P<extension>\w+)', 'myname@hackerrank.com')
>>> m.groupdict()
{'website': 'hackerrank', 'user': 'myname', 'extension': 'com'}
```

Task

You are given a string **S**.

Your task is to find the first occurrence of an alphanumeric character in **S** (read from left to right) that has consecutive repetitions.

Input Format

A single line of input containing the string S .

Constraints

$$0 < \text{len}(S) < 100$$

Output Format

Print the first occurrence of the repeating character. If there are no repeating characters, print -1.

Sample Input

```
..12345678910111213141516171820212223
```

Sample Output

```
1
```

Explanation

.. is the first repeating character, but it is not alphanumeric.

1 is the first (from left to right) alphanumeric repeating character of the string in the substring 111.

Change Theme Language Pypy 2



Line: 1 Col: 70

Upload Code as File



Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Re.findall() & Re.finditer() ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



re.findall()

The expression `re.findall()` returns all the non-overlapping matches of patterns in a string as a list of strings.

Code

```
>>> import re
>>> re.findall(r'\w','http://www.hackerrank.com/')
['h', 't', ' ', 'p', ' ', 'w', ' ', 'h', ' ', 'a', ' ', 'c', ' ', 'k', ' ', 'e', ' ', 'r', ' ', 'r', ' ', 'a', ' ', 'n', ' ', 'k', ' ', 'c', ' ', 'o', ' ', 'm']
```

re.finditer()

The expression `re.finditer()` returns an iterator yielding `MatchObject` instances over all non-overlapping matches for the `re` pattern in the string.

Code

```
>>> import re
>>> re.finditer(r'\w','http://www.hackerrank.com/')
<callable-iterator object at 0x0266C790>
>>> map(lambda x: x.group(),re.finditer(r'\w','http://www.hackerrank.com/'))
['h', 't', ' ', 'p', ' ', 'w', ' ', 'h', ' ', 'a', ' ', 'c', ' ', 'k', ' ', 'e', ' ', 'r', ' ', 'r', ' ', 'a', ' ', 'n', ' ', 'k', ' ', 'c', ' ', 'o', ' ', 'm']
```

Task

You are given a string S . It consists of alphanumeric characters, spaces and symbols(+,-).

Your task is to find all the substrings of S that contains **2** or more vowels.

Also, these substrings must lie in between **2** consonants and should contain vowels only.

Note :

Vowels are defined as: AEIOU and aeiou.

Consonants are defined as: QWRTYPSDFGHJKLZXCVBNM and qrvtypsdfghjklzxcvbnm.

Input Format

A single line of input containing string S .

Constraints

$0 < \text{len}(S) < 100$

Output Format

Print the matched substrings in their order of occurrence on separate lines.

If no match is found, print -1.

Sample Input

```
rabcddeefgyYhFjkIoomnp0eoreeeeet
```

Sample Output

```
ee  
Ioo  
Oeo  
eeeee
```

Explanation

ee is located between consonant **d** and **f**.

Ioo is located between consonant **k** and **m**.

Oeo is located between consonant **p** and **r**.

eeeeee is located between consonant **t** and **t**.

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⋮

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code

[Prepare](#) > [Python](#) > [Regex and Parsing](#) > [Re.start\(\) & Re.end\(\)](#)

Re.start() & Re.end() ★

[Problem](#)
[Submissions](#)
[Leaderboard](#)
[Discussions](#)
[Editorial](#)

start() & end()

These expressions return the indices of the start and end of the substring matched by the group.

Code

```
>>> import re
>>> m = re.search(r'\d+', '1234')
>>> m.end()
4
>>> m.start()
0
```

Task

You are given a string S .

Your task is to find the indices of the start and end of string k in S .

Input Format

The first line contains the string S .

The second line contains the string k .

Constraints

$0 < \text{len}(S) < 100$

$0 < \text{len}(k) < \text{len}(S)$

Output Format

Print the tuple in this format: (start_index, end_index).

If no match is found, print (-1, -1).

Sample Input

```
aaadaa
aa
```

Sample Output

```
(0, 1)
(1, 2)
(4, 5)
```

Author
[deleted]
Difficulty
Easy
Max Score
20
Submitted By
29160
NEED HELP?
🔗 [View discussions](#)
📖 [View editorial](#)
🏆 [View top submissions](#)
RATE THIS CHALLENGE

MORE DETAILS
⬇️ [Download problem statement](#)
⬇️ [Download sample test cases](#)
✍️ [Suggest Edits](#)


[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

[Prepare](#) > [Python](#) > [Regex and Parsing](#) > [Regex Substitution](#)

Regex Substitution ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)[Editorial](#)

The `re.sub()` tool (sub stands for substitution) evaluates a pattern and, for each valid match, it calls a method (or lambda).

The method is called for all matches and can be used to modify strings in different ways.

The `re.sub()` method returns the modified string as an output.

Learn more about [re. sub\(\)](#).

Transformation of Strings

[Code](#)

```
import re

#Squaring numbers
def square(match):
    number = int(match.group(0))
    return str(number**2)

print re.sub(r"\d+", square, "1 2 3 4 5 6 7 8 9")
```

[Output](#)

```
1 4 9 16 25 36 49 64 81
```

Replacements in Strings

[Code](#)

```
import re

html = """
<head>
<title>HTML</title>
</head>
<object type="application/x-flash"
  data="your-file.swf"
  width="0" height="0">
  <!-- <param name="movie" value="your-file.swf" /> -->
  <param name="quality" value="high"/>
</object>
"""

print re.sub("<!--.*?-->", "", html) #remove comment
```

[Output](#)

Author

[deleted]

Difficulty

Medium

Max Score

20

Submitted By

26383

NEED HELP?

[View discussions](#)
[View editorial](#)
[View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

[Download problem statement](#)
[Download sample test cases](#)
[Suggest Edits](#)


```
<head>
<title>HTML</title>
</head>
<object type="application/x-flash"
  data="your-file.swf"
  width="0" height="0">

  <param name="quality" value="high"/>
</object>
```

Task

You are given a text of N lines. The text contains `&&` and `||` symbols.

Your task is to modify those symbols to the following:

`&&` → `and`
`||` → `or`

Both `&&` and `||` should have a space " " on both sides.

Input Format

The first line contains the integer, N .

The next N lines each contain a line of the text.

Constraints

$0 < N < 100$

Neither `&&` nor `||` occur in the start or end of each line.

Output Format

Output the modified text.

Sample Input

```
11
a = 1;
b = input();

if a + b > 0 && a - b < 0:
    start()
elif a*b > 10 || a/b < 1:
    stop()
print set(list(a)) | set(list(b))
#Note do not change && or ||| or & or |
#Only change those '&&' which have space on both sides.
#Only change those '||' which have space on both sides.
```

Sample Output

```
a = 1;
b = input();

if a + b > 0 and a - b < 0:
    start()
elif a*b > 10 or a/b < 1:
    stop()
print set(list(a)) | set(list(b))
```

```
#Note do not change &&& or ||| or & or |
#Only change those '&&' which have space on both sides.
#Only change those '||' which have space on both sides.
```



Validating Roman Numerals ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given a string, and you have to validate whether it's a valid Roman numeral. If it is valid, print True. Otherwise, print False. Try to create a regular expression for a valid Roman numeral.

Input Format

A single line of input containing a string of Roman characters.

Output Format

Output a single line containing True or False according to the instructions above.

Constraints

The number will be between 1 and 3999 (both included).

Sample Input

CDXXI

Sample Output

True

References

Regular expressions are a key concept in any programming language. A quick explanation with Python examples is [available here](#). You could also go through the link below to read more about regular expressions in Python.

<https://developers.google.com/edu/python/regular-expressions>





Validating phone numbers ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Let's dive into the interesting topic of regular expressions! You are given some input, and you are required to check whether they are valid mobile numbers.

A valid mobile number is a ten digit number starting with a **7, 8 or 9**.

Concept

A valid mobile number is a ten digit number starting with a **7, 8 or 9**.

Regular expressions are a key concept in any programming language. A quick explanation with Python examples is [available here](#). You could also go through the link below to read more about regular expressions in Python.

<https://developers.google.com/edu/python/regular-expressions>

Input Format

The first line contains an integer **N**, the number of inputs.

N lines follow, each containing some string.

Constraints

1 ≤ N ≤ 10

2 ≤ len(Number) ≤ 15

Output Format

For every string listed, print "YES" if it is a valid mobile number and "NO" if it is not on separate lines. Do not print the quotes.

Sample Input

```
2
9587456281
1252478965
```

Sample Output

```
YES
NO
```



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Validating and Parsing Email Addresses ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



A valid email address meets the following criteria:

- It's composed of a username, domain name, and extension assembled in this format: `username@domain.extension`
- The username starts with an English alphabetical character, and any subsequent characters consist of one or more of the following: [alphanumeric characters](#), `-`, `,`, and `_`.
- The domain and extension contain only [English alphabetical characters](#).
- The extension is **1**, **2**, or **3** characters in length.

Given **n** pairs of names and email addresses as input, print each name and email address pair having a valid email address on a new line.

Hint: Try using [Email.utils\(\)](#) to complete this challenge. For example, this code:

```
import email.utils
print email.utils.parseaddr('DOSHI <DOSHI@hackerrank.com>')
print email.utils.formataddr(('DOSHI', 'DOSHI@hackerrank.com'))
```

produces this output:

```
('DOSHI', 'DOSHI@hackerrank.com')
DOSHI <DOSHI@hackerrank.com>
```

Input Format

The first line contains a single integer, **n**, denoting the number of email address.

Each line **i** of the **n** subsequent lines contains a name and an email address as two space-separated values following this format:

```
name <user@email.com>
```

Constraints

- $0 < n < 100$

Output Format

Print the space-separated name and email address pairs containing valid email addresses only. Each pair must be printed on a new line in the following format:

```
name <user@email.com>
```

You must print each valid email address in the same order as it was received as input.

Sample Input

```
2
DEXTER <dexter@hotmail.com>
VIRUS <virus!variable.:p>
```

Sample Output

```
DEXTER <dexter@hotmail.com>
```

Explanation

dexter@hotmail.com is a valid email address, so we print the name and email address pair received as input on a new line.
virus!@variable.:p is not a valid email address because the username contains an exclamation point (!) and the extension contains a colon (:).
As this email is not valid, we print nothing.

Change Theme Language Pypy 2



Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code



Hex Color Code ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



CSS colors are defined using a hexadecimal (HEX) notation for the combination of Red, Green, and Blue color values (RGB).

Specifications of HEX Color Code

- It must start with a '#' symbol.
- It can have **3** or **6** digits.
- Each digit is in the range of **0** to **F**. (**1, 2, 3, 4, 5, 6, 7, 8, 9, 0, A, B, C, D, E** and **F**).
- **A – F** letters can be lower case. (**a, b, c, d, e** and **f** are also valid digits).

Examples

Valid Hex Color Codes

```
#FFF  
#025  
#F0A1FB
```

Invalid Hex Color Codes

```
#ffffabg  
#abcf  
#12365erff
```

You are given **N** lines of CSS code. Your task is to print all valid Hex Color Codes, in order of their occurrence from top to bottom.

CSS Code Pattern

```
Selector
{
    Property: Value;
}
```

Input Format

The first line contains **N**, the number of code lines.

The next **N** lines contains CSS Codes.

Constraints

0 < N < 50

Output Format

Output the color codes with '#' symbols on separate lines.

Sample Input

```
11
#BED
{
```

```
color: #FfFdF8; background-color:#aef;
font-size: 123px;
background: -webkit-linear-gradient(top, #f9f9f9, #fff);
}
#Cab
{
    background-color: #ABC;
    border: 2px dashed #fff;
}
```

Sample Output

```
#FfFdF8
#aef
#f9f9f9
#fff
#ABC
#fff
```

Explanation

#BED and #Cab satisfy the Hex Color Code criteria, but they are used as selectors and not as color codes in the given CSS.

Hence, the valid color codes are:

```
#FfFdF8
#aef
#f9f9f9
#fff
#ABC
#fff
```

Note: There are no comments (// or /* */) in CSS Code.

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



HTML Parser - Part 1 ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



HTML

Hypertext Markup Language is a standard markup language used for creating World Wide Web pages.

Parsing

Parsing is the process of syntactic analysis of a string of symbols. It involves resolving a string into its component parts and describing their syntactic roles.

HTMLParser

An HTMLParser instance is fed HTML data and calls handler methods when start tags, end tags, text, comments, and other markup elements are encountered.

Example (based on the original Python documentation):

Code

```
from HTMLParser import HTMLParser

# create a subclass and override the handler methods
class MyHTMLParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print "Found a start tag :", tag
    def handle_endtag(self, tag):
        print "Found an end tag  :", tag
    def handle_startendtag(self, tag, attrs):
        print "Found an empty tag :", tag

# instantiate the parser and fed it some HTML
parser = MyHTMLParser()
parser.feed("<html><head><title>HTML Parser - I</title></head>" +
          "<body><h1>HackerRank</h1><br /></body></html>")
```

Output

```
Found a start tag : html
Found a start tag : head
Found a start tag : title
Found an end tag  : title
Found an end tag  : head
Found a start tag : body
Found a start tag : h1
Found an end tag  : h1
Found an empty tag : br
Found an end tag  : body
Found an end tag  : html
```

.handle_starttag(tag, attrs)

This method is called to handle the start tag of an element. (For example: <div class='marks'>)

The tag argument is the name of the tag converted to lowercase.

The attrs argument is a list of (name, value) pairs containing the attributes found inside the tag's <> brackets.

.handle_endtag(tag)

This method is called to handle the end tag of an element. (For example: </div>)

The tag argument is the name of the tag converted to lowercase.

.handle_startendtag(tag, attrs)

This method is called to handle the empty tag of an element. (For example:
)

The tag argument is the name of the tag converted to lowercase.

The attrs argument is a list of (name, value) pairs containing the attributes found inside the tag's <> brackets.

Task

You are given an HTML code snippet of N lines.

Your task is to print start tags, end tags and empty tags separately.

Format your results in the following way:

```
Start : Tag1
End   : Tag1
Start : Tag2
-> Attribute2[0] > Attribute_value2[0]
-> Attribute2[1] > Attribute_value2[1]
-> Attribute2[2] > Attribute_value2[2]
Start : Tag3
-> Attribute3[0] > None
Empty : Tag4
-> Attribute4[0] > Attribute_value4[0]
End   : Tag3
End   : Tag2
```

Here, the \rightarrow symbol indicates that the tag contains an attribute. It is immediately followed by the name of the attribute and the attribute value.

The $>$ symbol acts as a separator of the attribute and the attribute value.

If an HTML tag has no attribute then simply print the name of the tag.

If an attribute has no attribute value then simply print the name of the attribute value as None.

Note: Do not detect any HTML tag, attribute or attribute value inside the HTML comment tags ($<!--$ Comments $-->$).Comments can be multiline as well.

Input Format

The first line contains integer N , the number of lines in a HTML code snippet.

The next N lines contain HTML code.

Constraints

- $0 < N < 100$

Output Format

Print the HTML tags, attributes and attribute values in order of their occurrence from top to bottom in the given snippet.

Use proper formatting as explained in the problem statement.

Sample Input

```
2
<html><head><title>HTML Parser - I</title></head>
<body data-modal-target class='1'><h1>HackerRank</h1><br /></body></html>
```

Sample Output

```
Start : html
Start : head
Start : title
End   : title
End   : head
Start : body
-> data-modal-target > None
-> class > 1
Start : h1
End   : h1
Empty : br
End   : body
End   : html
```

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File



Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



HTML Parser - Part 2 ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



*This section assumes that you understand the basics discussed in [HTML Parser - Part 1](#)

[.handle_comment\(data\)](#)

This method is called when a comment is encountered (e.g. <!--comment-->).

The data argument is the content inside the comment tag:

```
from HTMLParser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_comment(self, data):
        print "Comment :", data
```

[.handle_data\(data\)](#)

This method is called to process arbitrary data (e.g. text nodes and the content of <script>...</script> and <style>...</style>).

The data argument is the text content of HTML.

```
from HTMLParser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_data(self, data):
        print "Data    :", data
```

Task

You are given an HTML code snippet of **N** lines.

Your task is to print the single-line comments, multi-line comments and the data.

Print the result in the following format:

```
>>> Single-line Comment
Comment
>>> Data
My Data
>>> Multi-line Comment
Comment_multiline[0]
Comment_multiline[1]
>>> Data
My Data
>>> Single-line Comment:
```

Note: Do not print data if data == '\n'.

Input Format

<https://www.hackerrank.com/challenges/html-parser-part-2/problem>

The first line contains integer N , the number of lines in the HTML code snippet.

The next N lines contains HTML code.

Constraints

$$0 < N < 100$$

Output Format

Print the single-line comments, multi-line comments and the data in order of their occurrence from top to bottom in the snippet.

Format the answers as explained in the problem statement.

Sample Input

```
4
<!--[if IE 9]>IE9-specific content
<![endif]-->
<div> Welcome to HackerRank</div>
<!--[if IE 9]>IE9-specific content<![endif]-->
```

Sample Output

```
>>> Multi-line Comment
[if IE 9]>IE9-specific content
<![endif]
>>> Data
    Welcome to HackerRank
>>> Single-line Comment
[if IE 9]>IE9-specific content<![endif]
```

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

 Upload Code as File Test against custom input**Run Code****Submit Code**

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



91/115 challenges solved

Rank: 10362 | Points: 1855



Detect HTML Tags, Attributes and Attribute Values

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given an HTML code snippet of N lines.

Your task is to detect and print all the HTML tags, attributes and attribute values.

Print the detected items in the following format:

```
Tag1
Tag2
-> Attribute2[0] > Attribute_value2[0]
-> Attribute2[1] > Attribute_value2[1]
-> Attribute2[2] > Attribute_value2[2]
Tag3
-> Attribute3[0] > Attribute_value3[0]
```

The \rightarrow symbol indicates that the tag contains an attribute. It is immediately followed by the name of the attribute and the attribute value.

The $>$ symbol acts as a separator of attributes and attribute values.

If an HTML tag has no attribute then simply print the name of the tag.

Note: Do not detect any HTML tag, attribute or attribute value inside the HTML comment tags ($<!--$ Comments $-->$). Comments can be multiline.

All attributes have an attribute value.

Input Format

The first line contains an integer N , the number of lines in the HTML code snippet.

The next N lines contain HTML code.

Constraints

$0 < N < 100$

Output Format

Print the HTML tags, attributes and attribute values in order of their occurrence from top to bottom in the snippet.

Format your answers as explained in the problem statement.

Sample Input

```
9
<head>
<title>HTML</title>
</head>
<object type="application/x-flash">

    data="your-file.swf"
```

```
width="0" height="0">
<!-- <param name="movie" value="your-file.swf" /> -->
<param name="quality" value="high"/>
</object>
```

Sample Output

```
head
title
object
-> type > application/x-flash
-> data > your-file.swf
-> width > 0
-> height > 0
param
-> name > quality
-> value > high
```

Explanation

- **head** tag: Print the head tag only because it has no attribute.
- **title** tag: Print the title tag only because it has no attribute.
- **object** tag: Print the object tag. In the next **4** lines, print the attributes type, data, width and height along with their respective values.
- **<!-- Comment -->** tag: Don't detect anything inside it.
- **param** tag: Print the param tag. In the next **2** lines, print the attributes name along with their respective values.

Change Theme Language Pypy 2 ▾



Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Validating UID ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



ABCXYZ company has up to **100** employees.

The company decides to create a unique identification number (UID) for each of its employees.

The company has assigned you the task of validating all the randomly generated UIDs.

A valid UID must follow the rules below:

- It must contain at least **2** uppercase English alphabet characters.
- It must contain at least **3** digits (**0 - 9**).
- It should only contain alphanumeric characters (**a - z, A - Z & 0 - 9**).
- No character should repeat.
- There must be exactly **10** characters in a valid UID.

Input Format

The first line contains an integer **T**, the number of test cases.

The next **T** lines contains an employee's UID.

Output Format

For each test case, print 'Valid' if the UID is valid. Otherwise, print 'Invalid', on separate lines. Do not print the quotation marks.

Sample Input

```
2
B1CD102354
B1CDEF2354
```

Sample Output

```
Invalid
Valid
```

Explanation

B1CD102354: **1** is repeating → Invalid

B1CDEF2354: Valid





[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



91/115 challenges solved

Rank: 10362 | Points: 1855 !



Validating Credit Card Numbers ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You and Fredrick are good friends. Yesterday, Fredrick received **N** credit cards from **ABCD Bank**. He wants to verify whether his credit card numbers are valid or not. You happen to be great at regex so he is asking for your help!

A valid credit card from **ABCD Bank** has the following characteristics:

- ▶ It must start with a **4**, **5** or **6**.
- ▶ It must contain exactly **16** digits.
- ▶ It must only consist of digits (**0-9**).
- ▶ It may have digits in groups of **4**, separated by one hyphen **"-"**.
- ▶ It must NOT use any other separator like **' '**, **'_'**, etc.
- ▶ It must NOT have **4** or more consecutive repeated digits.

Examples:

Valid Credit Card Numbers

```
4253625879615786  
4424424424442444  
5122-2368-7954-3214
```

Invalid Credit Card Numbers

```
42536258796157867      #17 digits in card number → Invalid  
4424444424442444      #Consecutive digits are repeating 4 or more times → Invalid  
5122-2368-7954 - 3214  #Separators other than '-' are used → Invalid  
44244x4424442444      #Contains non digit characters → Invalid  
0525362587961578      #Doesn't start with 4, 5 or 6 → Invalid
```

Input Format

The first line of input contains an integer **N**.

The next **N** lines contain credit card numbers.

Constraints

0 < N < 100

Output Format

Print 'Valid' if the credit card number is valid. Otherwise, print 'Invalid'. Do not print the quotes.

Sample Input

```
6  
4123456789123456
```

```
5123-4567-8912-3456
61234-567-8912-3456
4123356789123456
5133-3367-8912-3456
5123 - 3567 - 8912 - 3456
```

Sample Output

```
Valid
Valid
Invalid
Valid
Invalid
Invalid
```

Explanation

4123456789123456 : **Valid**

5123-4567-8912-3456 : **Valid**

61234-**567**-8912-3456 : **Invalid**, because the card number is not divided into equal groups of **4**.

4123356789123456 : **Valid**

51**33-33**67-8912-3456 : **Invalid**, consecutive digits **3333** is repeating **4** times.

5123 - 4567 - 8912 - 3456 : **Invalid**, because space ' ' and - are used as separators.

Change Theme Language Pypy 2

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```



Line: 1 Col: 70

[Upload Code as File](#) [Test against custom input](#)[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Validating Postal Codes ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



A valid postal code P have to fullfil both below requirements:

1. P must be a number in the range from **100000** to **999999** inclusive.
2. P must not contain more than one alternating repetitive digit pair.

Alternating repetitive digits are digits which repeat immediately after the next digit. In other words, an alternating repetitive digit pair is formed by two equal digits that have just a single digit between them.

For example:

```
121426 # Here, 1 is an alternating repetitive digit.  
523563 # Here, NO digit is an alternating repetitive digit.  
552523 # Here, both 2 and 5 are alternating repetitive digits.
```

Your task is to provide two regular expressions `regex_integer_in_range` and `regex_alternating_repetitive_digit_pair`. Where:

`regex_integer_in_range` should match only integers range from **100000** to **999999** inclusive

`regex_alternating_repetitive_digit_pair` should find alternating repetitive digits pairs in a given string.

Both these regular expressions will be used by the provided code template to check if the input string P is a valid postal code using the following expression:

```
(bool(re.match(regex_integer_in_range, P))  
and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```

Input Format

Locked stub code in the editor reads a single string denoting P from `stdin` and uses provided expression and your regular expressions to validate if P is a valid postal code.

Output Format

You are not responsible for printing anything to `stdout`. Locked stub code in the editor does that.

Sample Input 0

```
110000
```

Sample Output 0

```
False
```

Explanation 0

`1 1 0000` : (0, 0) and (0, 0) are two alternating digit pairs. Hence, it is an invalid postal code.

Note:

A score of **0** will be awarded for using 'if' conditions in your code.

You have to pass all the testcases to get a positive score.



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Matrix Script ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Neo has a complex matrix script. The matrix script is a $N \times M$ grid of strings. It consists of alphanumeric characters, spaces and symbols (!,@,#,\$,%,&).

Matrix Script

The grid contains the following characters:

T	s	i
h	%	x
i		#
s	M	
\$	a	
#	t	%
i	r	!

Matrix Decoded

This\$#is% Matrix# %!

To decode the script, Neo needs to read each column and select only the alphanumeric characters and connect them. Neo reads the column from top to bottom and starts reading from the leftmost column.

If there are symbols or spaces between two alphanumeric characters of the decoded script, then Neo replaces them with a single space ' ' for better readability.

Neo feels that there is no need to use 'if' conditions for decoding.

Alphanumeric characters consist of: [A-Z, a-z, and 0-9].

Input Format

The first line contains space-separated integers N (rows) and M (columns) respectively.

The next N lines contain the row elements of the matrix script.

Constraints

$0 < N, M < 100$

Note: A 0 score will be awarded for using 'if' conditions in your code.

Output Format

Print the decoded matrix script.

Sample Input 0

```
7 3
Tsi
h%x
i #
sM
$a
#t%
ir!
```

Sample Output 0

```
This is Matrix# %!
```

Explanation 0

The decoded script is:

```
This$is% Matrix# %!
```

Neo replaces the symbols or spaces between two alphanumeric characters with a single space ' ' for better readability.

So, the final decoded script is:

```
This is Matrix# %!
```





91/115 challenges solved

Rank: 10362 | Points: 1855 !



XML 1 - Find the Score ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given a valid XML document, and you have to print its score. The score is calculated by the sum of the score of each element. For any element, the score is equal to the number of attributes it has.

Input Format

The first line contains **N**, the number of lines in the XML document.

The next **N** lines follow containing the XML document.

Output Format

Output a single line, the integer score of the given XML document.

Sample Input

```
6
<feed xml:lang='en'>
    <title>HackerRank</title>
    <subtitle lang='en'>Programming challenges</subtitle>
    <link rel='alternate' type='text/html' href='http://hackerrank.com/'/>
    <updated>2013-12-25T12:00:00</updated>
</feed>
```

Sample Output

```
5
```

Explanation

The feed and subtitle tag have one attribute each - lang.

The title and updated tags have no attributes.

The link tag has three attributes - rel, type and href.

So, the total score is **1 + 1 + 3 = 5**.

There may be any level of nesting in the XML document. To learn about XML parsing, refer [here](#).

NOTE: In order to parse and generate an XML element tree, use the following code:

```
>> import xml.etree.ElementTree as etree
>> tree = etree.ElementTree(etree.fromstring(xml))
```

Here, XML is the variable containing the string.

Also, to find the number of keys in a dictionary, use the len function:

```
>> dicti = {'0': 'This is zero', '1': 'This is one'}
>> print (len(dicti))
```

2

[Change Theme](#)

Language

Pypy 2



Line: 3 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



XML2 - Find the Maximum Depth ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are given a valid XML document, and you have to print the maximum level of nesting in it. Take the depth of the root as **0**.

Input Format

The first line contains **N**, the number of lines in the XML document.

The next **N** lines follow containing the XML document.

Output Format

Output a single line, the integer value of the maximum level of nesting in the XML document.

Sample Input

```
6
<feed xml:lang='en'>
    <title>HackerRank</title>
    <subtitle lang='en'>Programming challenges</subtitle>
    <link rel='alternate' type='text/html' href='http://hackerrank.com/'/>
    <updated>2013-12-25T12:00:00</updated>
</feed>
```

Sample Output

```
1
```

Explanation

Here, the root is a feed tag, which has depth of **0**.

The tags title, subtitle, link and updated all have a depth of **1**.

Thus, the maximum depth is **1**.

[Change Theme](#)

Language

Pypy 2



```
1 import xml.etree.ElementTree as etree
2
3 maxdepth = 0
4 def depth(elem, level):
5     global maxdepth
6     # your code goes here
7
8 if __name__ == '__main__': ...
```

Line: 2 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Standardize Mobile Number Using Decorators ★

91/115 challenges solved
Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Let's dive into decorators! You are given **N** mobile numbers. Sort them in ascending order then print them in the standard format shown below:

```
+91 xxxx xxxx
```

The given mobile numbers may have **+91**, **91** or **0** written before the actual **10** digit number. Alternatively, there may not be any prefix at all.

Input Format

The first line of input contains an integer **N**, the number of mobile phone numbers.

N lines follow each containing a mobile number.

Output Format

Print **N** mobile numbers on separate lines in the required format.

Sample Input

```
3
07895462130
919875641230
9195969878
```

Sample Output

```
+91 78954 62130
+91 91959 69878
+91 98756 41230
```

Concept

Like most other programming languages, Python has the concept of closures. Extending these closures gives us decorators, which are an invaluable asset. You can learn about decorators in 12 easy steps [here](#).

To solve the above question, make a list of the mobile numbers and pass it to a function that sorts the array in ascending order. Make a decorator that standardizes the mobile numbers and apply it to the function.

Language: Python 2



[Change Theme](#)

```
1 def wrapper(f):
2     def fun(l):
3         # complete the function
4     return fun
5
6 @wrapper ...
```

Line: 1 Col: 1

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



91/115 challenges solved

Rank: 10362 | Points: 1855 !



Decorators 2 - Name Directory ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Let's use decorators to build a name directory! You are given some information about **N** people. Each person has a first name, last name, age and sex. Print their names in a specific format sorted by their age in ascending order i.e. the youngest person's name should be printed first. For two people of the same age, print them in the order of their input.

For Henry Davids, the output should be:

Mr. Henry Davids

For Mary George, the output should be:

Ms. Mary George

Input Format

The first line contains the integer **N**, the number of people.

N lines follow each containing the space separated values of the first name, last name, age and sex, respectively.

Constraints

1 ≤ N ≤ 10

Output Format

Output **N** names on separate lines in the format described above in ascending order of age.

Sample Input

```
3
Mike Thomson 20 M
Robert Bustle 32 M
Andria Bustle 30 F
```

Sample Output

```
Mr. Mike Thomson
Ms. Andria Bustle
Mr. Robert Bustle
```

Concept

For sorting a nested list based on some parameter, you can use the itemgetter library. You can read more about it [here](#).

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇

```
1 import operator
2
3 def person_lister(f):
4     def inner(people):
5         # complete the function
6         return inner
7
8 @person_lister...

```

Line: 2 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code



Arrays ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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`.

Input Format

A single line of input containing space separated numbers.

Output Format

Print the reverse NumPy array with type float.

Sample Input

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

Sample Output

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



Line: 2 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



Shape and Reshape ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



shape

The shape tool gives a tuple of array dimensions and can be used to change the dimensions of an array.

(a). Using shape to get array dimensions

```
import numpy

my_1D_array = numpy.array([1, 2, 3, 4, 5])
print my_1D_array.shape    #(5,) -> 1 row and 5 columns

my_2D_array = numpy.array([[1, 2],[3, 4],[6,5]])
print my_2D_array.shape    #(3, 2) -> 3 rows and 2 columns
```

(b). Using shape to change array dimensions

```
import numpy

change_array = numpy.array([1,2,3,4,5,6])
change_array.shape = (3, 2)
print change_array

#Output
[[1 2]
 [3 4]
 [5 6]]
```

reshape

The reshape tool gives a new shape to an array without changing its data. It creates a new array and does not modify the original array itself.

```
import numpy

my_array = numpy.array([1,2,3,4,5,6])
print numpy.reshape(my_array,(3,2))

#Output
[[1 2]
 [3 4]
 [5 6]]
```

Task

You are given a space separated list of nine integers. Your task is to convert this list into a **3X3** NumPy array.

Input Format

A single line of input containing **9** space separated integers.

<https://www.hackerrank.com/challenges/np-shape-reshape/problem>

Output Format

Print the **3X3** NumPy array.

Sample Input

```
1 2 3 4 5 6 7 8 9
```

Sample Output

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Change Theme Language Pypy 2



```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code



91/115 challenges solved

Rank: 10362 | Points: 1855 !



Transpose and Flatten ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Transpose

We can generate the transposition of an array using the tool `numpy.transpose`.

It will not affect the original array, but it will create a new array.

```
import numpy

my_array = numpy.array([[1,2,3],
                      [4,5,6]])
print numpy.transpose(my_array)

#Output
[[1 4]
 [2 5]
 [3 6]]
```

Flatten

The tool `flatten` creates a copy of the input array flattened to one dimension.

```
import numpy

my_array = numpy.array([[1,2,3],
                      [4,5,6]])
print my_array.flatten()

#Output
[1 2 3 4 5 6]
```

Task

You are given a $N \times M$ integer array matrix with space separated elements (N = rows and M = columns).

Your task is to print the transpose and flatten results.

Input Format

The first line contains the space separated values of N and M .

The next N lines contains the space separated elements of M columns.

Output Format

First, print the transpose array and then print the flatten.

Sample Input

```
2 2
1 2
3 4
```

Sample Output

```
[[1 3]
[2 4]]
[1 2 3 4]
```

Change Theme Language Pypy 2 ▾



```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code



Concatenate ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Concatenate

Two or more arrays can be concatenated together using the concatenate function with a tuple of the arrays to be joined:

```
import numpy

array_1 = numpy.array([1,2,3])
array_2 = numpy.array([4,5,6])
array_3 = numpy.array([7,8,9])

print numpy.concatenate((array_1, array_2, array_3))

#Output
[1 2 3 4 5 6 7 8 9]
```

If an array has more than one dimension, it is possible to specify the axis along which multiple arrays are concatenated. By default, it is along the first dimension.

```
import numpy

array_1 = numpy.array([[1,2,3],[0,0,0]])
array_2 = numpy.array([[0,0,0],[7,8,9]])

print numpy.concatenate((array_1, array_2), axis = 1)

#Output
[[1 2 3 0 0 0]
 [0 0 0 7 8 9]]
```

Task

You are given two integer arrays of size $N \times P$ and $M \times P$ (N & M are rows, and P is the column). Your task is to concatenate the arrays along axis **0**.

Input Format

The first line contains space separated integers N , M and P .

The next N lines contains the space separated elements of the P columns.

After that, the next M lines contains the space separated elements of the P columns.

Output Format

Print the concatenated array of size $(N + M) \times P$.

Sample Input

```
4 3 2
1 2
1 2
1 2
1 2
3 4
3 4
3 4
```

Sample Output

```
[[1 2]
[1 2]
[1 2]
[1 2]
[3 4]
[3 4]
[3 4]]
```

Change Theme Language Pypy 2 ▼

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Test against custom input

[Prepare](#) > [Python](#) > [Numpy](#) > [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`.

Input Format

A single line containing the space-separated integers.

Constraints

1 ≤ each integer ≤ 3

Output Format

First, print the array using the `numpy.zeros` tool and then print the array with the `numpy.ones` tool.

Sample Input 0

3 3 3

Sample Output 0

Author	[deleted]
Difficulty	Easy
Max Score	20
Submitted By	51836

NEED HELP?

↳ [View discussions](#)

↳ [View editorial](#)

🏆 [View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

⬇️ [Download problem statement](#)

⬇️ [Download sample test cases](#)

✍️ [Suggest Edits](#)



```
[[[0 0 0]
 [0 0 0]
 [0 0 0]]]
```

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]]
```

```
[[[0 0 0]
 [0 0 0]
 [0 0 0]]]
 [[[1 1 1]
 [1 1 1]
 [1 1 1]]]
```

```
[[[1 1 1]
 [1 1 1]
 [1 1 1]]]
```

```
[[[1 1 1]
 [1 1 1]
 [1 1 1]]]]
```

Explanation 0

Print the array built using `numpy.zeros` and `numpy.ones` tools and you get the result as shown.



91/115 challenges solved
Rank: 10362 | Points: 1855 !



Eye and Identity ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



identity

The identity tool returns an identity array. An identity array is a square matrix with all the main diagonal elements as **1** and the rest as **0**. The default type of elements is float.

```
import numpy
print numpy.identity(3) #3 is for dimension 3 X 3

#Output
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
```

eye

The eye tool returns a 2-D array with **1**'s as the diagonal and **0**'s elsewhere. The diagonal can be main, upper or lower depending on the optional parameter **k**. A positive **k** is for the upper diagonal, a negative **k** is for the lower, and a **0 k** (default) is for the main diagonal.

```
import numpy
print numpy.eye(8, 7, k = 1)    # 8 X 7 Dimensional array with first upper diagonal 1.

#Output
[[ 0.  1.  0.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.]]

print numpy.eye(8, 7, k = -2)   # 8 X 7 Dimensional array with second lower diagonal 1.
```

Task

Your task is to print an array of size **N** × **M** with its main diagonal elements as **1**'s and **0**'s everywhere else.

Note

In order to get alignment correct, please insert the line `numpy.set_printoptions(legacy='1.13')` below the numpy import.

Input Format

A single line containing the space separated values of **N** and **M**.

N denotes the rows.

M denotes the columns.

Output Format

Print the desired $N \times M$ array.

Sample Input

```
3 3
```

Sample Output

```
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
```

Change Theme Language Pypy 2 ▾



Line: 1 Col: 70

Upload Code as File



Test against custom input

Run Code

Submit Code



91/115 challenges solved

Rank: 10362 | Points: 1855 !



Array Mathematics ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



Basic mathematical functions operate element-wise on arrays. They are available both as operator overloads and as functions in the NumPy module.

```
import numpy

a = numpy.array([1,2,3,4], float)
b = numpy.array([5,6,7,8], float)

print a + b          #[ 6.  8. 10. 12.]
print numpy.add(a, b)#[ 6.  8. 10. 12.]

print a - b          #[-4. -4. -4. -4.]
print numpy.subtract(a, b)#[ -4. -4. -4. -4.]

print a * b          #[ 5. 12. 21. 32.]
print numpy.multiply(a, b)#[ 5. 12. 21. 32.]

print a / b          #[ 0.2      0.33333333 0.42857143 0.5      ]
print numpy.divide(a, b)#[ 0.2      0.33333333 0.42857143 0.5      ]

print a % b          #[ 1.  2.  3.  4.]
print numpy.mod(a, b)#[ 1.  2.  3.  4.]

print a**b           #[ 1.0000000e+00 6.4000000e+01 2.1870000e+03 6.5536000e+04]
print numpy.power(a, b)#[ 1.0000000e+00 6.4000000e+01 2.1870000e+03 6.5536000e+04]
```

Task

You are given two integer arrays, \mathbf{A} and \mathbf{B} of dimensions $N \times M$.

Your task is to perform the following operations:

1. Add ($\mathbf{A} + \mathbf{B}$)
2. Subtract ($\mathbf{A} - \mathbf{B}$)
3. Multiply ($\mathbf{A} * \mathbf{B}$)
4. Integer Division (\mathbf{A} / \mathbf{B})
5. Mod ($\mathbf{A} \% \mathbf{B}$)
6. Power ($\mathbf{A}^{**} \mathbf{B}$)

Note

There is a method `numpy.floor_divide()` that works like `numpy.divide()` except it performs a floor division.

Input Format

The first line contains two space separated integers, N and M .

The next N lines contains M space separated integers of array \mathbf{A} .

The following N lines contains M space separated integers of array \mathbf{B} .

Output Format

Print the result of each operation in the given order under **Task**.

Sample Input

```
1 4  
1 2 3 4  
5 6 7 8
```

Sample Output

```
[[ 6  8 10 12]]  
[[-4 -4 -4 -4]]  
[[ 5 12 21 32]]  
[[0 0 0 0]]  
[[1 2 3 4]]  
[[    1    64  2187 65536]]
```

Use // for division in Python 3.

[Change Theme](#)[Language](#)

Pypy 2



Line: 1 Col: 70

[Upload Code as File](#)[Test against custom input](#)[Run Code](#)[Submit Code](#)

Floor, Ceil and Rint ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



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 .

Note

In order to get the correct output format, add the line `numpy.set_printoptions(legacy='1.13')` below the numpy import.

Input Format

A single line of input containing the space separated elements of array A .

Output Format

On the first line, print the `floor` of A .

On the second line, print the `ceil` of A .

On the third line, print the `rint` of A .

Sample Input

```
1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 9.9
```

Sample Output

```
[ 1.  2.  3.  4.  5.  6.  7.  8.  9.]
[ 2.  3.  4.  5.  6.  7.  8.  9.  10.]
[ 1.  2.  3.  4.  6.  7.  8.  9.  10.]
```

Change Theme Language Pypy 2 ▾ ⌂ ⌃ ⌄ ⌅

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

[Upload Code as File](#)[Test against custom input](#)[Run Code](#)[Submit Code](#)



Sum and Prod ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



sum

The sum tool returns the sum of array elements over a given axis.

```
import numpy

my_array = numpy.array([ [1, 2], [3, 4] ])

print numpy.sum(my_array, axis = 0)      #Output : [4 6]
print numpy.sum(my_array, axis = 1)      #Output : [3 7]
print numpy.sum(my_array, axis = None)    #Output : 10
print numpy.sum(my_array)                #Output : 10
```

By default, the axis value is `None`. Therefore, it performs a sum over all the dimensions of the input array.

prod

The prod tool returns the product of array elements over a given axis.

```
import numpy

my_array = numpy.array([ [1, 2], [3, 4] ])

print numpy.prod(my_array, axis = 0)       #Output : [3 8]
print numpy.prod(my_array, axis = 1)       #Output : [ 2 12]
print numpy.prod(my_array, axis = None)     #Output : 24
print numpy.prod(my_array)                 #Output : 24
```

By default, the axis value is `None`. Therefore, it performs the product over all the dimensions of the input array.

Task

You are given a 2-D array with dimensions $N \times M$.

Your task is to perform the `sum` tool over axis `0` and then find the `product` of that result.

Input Format

The first line of input contains space separated values of N and M .

The next N lines contains M space separated integers.

Output Format

Compute the sum along axis `0`. Then, print the product of that sum.

Sample Input

```
2 2
1 2
3 4
```

Sample Output

24

Explanation

The sum along axis **0** = **[4 6]**

The product of this sum = **24**

[Change Theme](#)

Language

Pypy 2



Line: 1 Col: 70

[Upload Code as File](#)

Test against custom input

[Run Code](#)[Submit Code](#)



Min and Max ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



min

The tool min returns the minimum value along a given axis.

```
import numpy

my_array = numpy.array([[2, 5],
                      [3, 7],
                      [1, 3],
                      [4, 0]])

print numpy.min(my_array, axis = 0)      #Output : [1 0]
print numpy.min(my_array, axis = 1)      #Output : [2 3 1 0]
print numpy.min(my_array, axis = None)    #Output : 0
print numpy.min(my_array)                #Output : 0
```

By default, the axis value is None. Therefore, it finds the minimum over all the dimensions of the input array.

max

The tool max returns the maximum value along a given axis.

```
import numpy

my_array = numpy.array([[2, 5],
                      [3, 7],
                      [1, 3],
                      [4, 0]])

print numpy.max(my_array, axis = 0)       #Output : [4 7]
print numpy.max(my_array, axis = 1)       #Output : [5 7 3 4]
print numpy.max(my_array, axis = None)     #Output : 7
print numpy.max(my_array)                 #Output : 7
```

By default, the axis value is None. Therefore, it finds the maximum over all the dimensions of the input array.

Task

You are given a 2-D array with dimensions $N \times M$.

Your task is to perform the min function over axis 1 and then find the max of that.

Input Format

The first line of input contains the space separated values of N and M .

The next N lines contains M space separated integers.

Output Format

Compute the min along axis 1 and then print the max of that result.

Sample Input

```
4 2
2 5
3 7
1 3
4 0
```

Sample Output

```
3
```

Explanation

The min along axis **1** = **[2, 3, 1, 0]**

The max of **[2, 3, 1, 0]** = **3**





91/115 challenges solved

Rank: 10362 | Points: 1855



Mean, Var, and Std ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



mean

The mean tool computes the arithmetic mean along the specified axis.

```
import numpy

my_array = numpy.array([ [1, 2], [3, 4] ])

print numpy.mean(my_array, axis = 0)      #Output : [ 2.  3.]
print numpy.mean(my_array, axis = 1)      #Output : [ 1.5  3.5]
print numpy.mean(my_array, axis = None)    #Output : 2.5
print numpy.mean(my_array)                #Output : 2.5
```

By default, the axis is None. Therefore, it computes the mean of the flattened array.

var

The var tool computes the arithmetic variance along the specified axis.

```
import numpy

my_array = numpy.array([ [1, 2], [3, 4] ])

print numpy.var(my_array, axis = 0)        #Output : [ 1.  1.]
print numpy.var(my_array, axis = 1)        #Output : [ 0.25  0.25]
print numpy.var(my_array, axis = None)      #Output : 1.25
print numpy.var(my_array)                 #Output : 1.25
```

By default, the axis is None. Therefore, it computes the variance of the flattened array.

std

The std tool computes the arithmetic standard deviation along the specified axis.

```
import numpy

my_array = numpy.array([ [1, 2], [3, 4] ])

print numpy.std(my_array, axis = 0)         #Output : [ 1.  1.]
print numpy.std(my_array, axis = 1)         #Output : [ 0.5  0.5]
print numpy.std(my_array, axis = None)       #Output : 1.11803398875
print numpy.std(my_array)                  #Output : 1.11803398875
```

By default, the axis is None. Therefore, it computes the standard deviation of the flattened array.

Task

You are given a 2-D array of size $N \times M$.

<https://www.hackerrank.com/challenges/np-mean-var-and-std/problem>

Your task is to find:

1. The mean along axis **1**
2. The var along axis **0**
3. The std along axis **None**

Input Format

The first line contains the space separated values of **N** and **M**.

The next **N** lines contains **M** space separated integers.

Output Format

First, print the mean.

Second, print the var.

Third, print the std.

Sample Input

```
2 2
1 2
3 4
```

Sample Output

```
[ 1.5  3.5]
[ 1.  1.]
1.11803398875
```

Change Theme Language Pypy 2



Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Dot and Cross ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



dot

The dot tool returns the dot product of two arrays.

```
import numpy

A = numpy.array([ 1, 2 ])
B = numpy.array([ 3, 4 ])

print numpy.dot(A, B)      #Output : 11
```

cross

The cross tool returns the cross product of two arrays.

```
import numpy

A = numpy.array([ 1, 2 ])
B = numpy.array([ 3, 4 ])

print numpy.cross(A, B)    #Output : -2
```

Task

You are given two arrays \mathbf{A} and \mathbf{B} . Both have dimensions of $N \times N$.

Your task is to compute their **matrix product**.

Input Format

The first line contains the integer N .

The next N lines contains N space separated integers of array \mathbf{A} .

The following N lines contains N space separated integers of array \mathbf{B} .

Output Format

Print the matrix multiplication of \mathbf{A} and \mathbf{B} .

Sample Input

```
2
1 2
3 4
1 2
3 4
```

Sample Output

<https://www.hackerrank.com/challenges/np-dot-and-cross/problem>

```
[[ 7 10]
 [15 22]]
```

Change Theme Language Pypy 2 ▾



Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Inner and Outer ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



inner

The inner tool returns the [inner product](#) of two arrays.

```
import numpy

A = numpy.array([0, 1])
B = numpy.array([3, 4])

print numpy.inner(A, B)      #Output : 4
```

outer

The outer tool returns the [outer product](#) of two arrays.

```
import numpy

A = numpy.array([0, 1])
B = numpy.array([3, 4])

print numpy.outer(A, B)      #Output : [[0 0]
                            #                  [3 4]]
```

Task

You are given two arrays: **A** and **B**.

Your task is to compute their inner and outer product.

Input Format

The first line contains the space separated elements of array **A**.

The second line contains the space separated elements of array **B**.

Output Format

First, print the inner product.

Second, print the outer product.

Sample Input

```
0 1
2 3
```

Sample Output

```
3
[[0 0]
 [2 3]]
```

Change Theme Language Pypy 2



```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |
[Request a Feature](#)



91/115 challenges solved

Rank: 10362 | Points: 1855 !



Polynomials ★

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



poly

The poly tool returns the coefficients of a polynomial with the given sequence of roots.

```
print numpy.poly([-1, 1, 1, 10])      #Output : [ 1 -11  9  11 -10]
```

roots

The roots tool returns the roots of a polynomial with the given coefficients.

```
print numpy.roots([1, 0, -1])        #Output : [-1.  1.]
```

polyint

The polyint tool returns an antiderivative (indefinite integral) of a polynomial.

```
print numpy.polyint([1, 1, 1])       #Output : [ 0.33333333  0.5         1.          0.          ]
```

polyder

The polyder tool returns the derivative of the specified order of a polynomial.

```
print numpy.polyder([1, 1, 1, 1])    #Output : [3 2 1]
```

polyval

The polyval tool evaluates the polynomial at specific value.

```
print numpy.polyval([1, -2, 0, 2], 4)  #Output : 34
```

polyfit

The polyfit tool fits a polynomial of a specified order to a set of data using a least-squares approach.

```
print numpy.polyfit([0,1,-1, 2, -2], [0,1,1, 4, 4], 2)
#Output : [ 1.00000000e+00  0.00000000e+00 -3.97205465e-16]
```

The functions [polyadd](#), [polysub](#), [polymul](#), and [polydiv](#) also handle proper addition, subtraction, multiplication, and division of polynomial coefficients, respectively.

Task

You are given the coefficients of a polynomial P .

Your task is to find the value of P at point x .

Input Format

The first line contains the space separated value of the coefficients in P .

The second line contains the value of x .

Output Format

Print the desired value.

Sample Input

```
1.1 2 3  
0
```

Sample Output

```
3.0
```

Change Theme Language Pypy 2

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
```

Line: 1 Col: 70

Upload Code as File

Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Linear Algebra ★

91/115 challenges solved

Rank: 10362 | Points: 1855 !

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



The NumPy module also comes with a number of built-in routines for linear algebra calculations. These can be found in the sub-module linalg.

linalg.det

The linalg.det tool computes the determinant of an array.

```
print numpy.linalg.det([[1 , 2], [2, 1]])      #Output : -3.0
```

linalg.eig

The linalg.eig computes the eigenvalues and right eigenvectors of a square array.

```
vals, vecs = numpy.linalg.eig([[1 , 2], [2, 1]])
print vals
print vecs
```

#Output : [3. -1.]
#Output : [[0.70710678 -0.70710678]
[0.70710678 0.70710678]]

linalg.inv

The linalg.inv tool computes the (multiplicative) inverse of a matrix.

```
print numpy.linalg.inv([[1 , 2], [2, 1]])      #Output : [[-0.33333333  0.66666667]
#                  [ 0.66666667 -0.33333333]]
```

Other routines can be found [here](#)

Task

You are given a square matrix A with dimensions $N \times N$. Your task is to find the determinant. Note: Round the answer to 2 places after the decimal.

Input Format

The first line contains the integer N .

The next N lines contains the N space separated elements of array A .

Output Format

Print the determinant of A .

Sample Input

```
2
1.1 1.1
1.1 1.1
```

Sample Output

0.0

Change Theme Language Pypy 2



1 # Enter your code here. Read input from STDIN. Print output to STDOUT

Line: 1 Col: 70

 Upload Code as File Test against custom input Run Code Submit Code[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Birthday Cake Candles ★

75 more points to get your next star!

Rank: 1055287 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count how many candles are tallest.

Example

candles = [4, 4, 1, 3]

The maximum height candles are **4** units high. There are **2** of them, so return **2**.

Function Description

Complete the function `birthdayCakeCandles` in the editor below.

`birthdayCakeCandles` has the following parameter(s):

- int `candles[n]`: the candle heights

Returns

- int: the number of candles that are tallest

Input Format

The first line contains a single integer, **n**, the size of `candles[]`.

The second line contains **n** space-separated integers, where each integer **i** describes the height of `candles[i]`.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq \text{candles}[i] \leq 10^7$

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

Candle heights are **[3, 2, 1, 3]**. The tallest candles are **3** units, and there are **2** of them.

```
1  #!/bin/python
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9  #
10 # Complete the 'birthdayCakeCandles' function below.
11 #
12 # The function is expected to return an INTEGER.
13 # The function accepts INTEGER_ARRAY candles as parameter.
14 #
15
16 def birthdayCakeCandles(candles):
17     # Write your code here
18
19 if __name__ == '__main__':
20     fptr = open(os.environ['OUTPUT_PATH'], 'w')
21
22     candles_count = int(raw_input().strip())
23
24     candles = map(int, raw_input().rstrip().split())
```

Line: 31 Col: 1

 [Upload Code as File](#) [Test against custom input](#)[Run Code](#)[Submit Code](#)



Number Line Jumps ★

75 more points to get your next star!

Rank: 1055289 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction (i.e, toward positive infinity).

- The first kangaroo starts at location x_1 and moves at a rate of v_1 meters per jump.
- The second kangaroo starts at location x_2 and moves at a rate of v_2 meters per jump.

You have to figure out a way to get both kangaroos at the same location at the same time as part of the show. If it is possible, return YES, otherwise return NO.

Example

$x_1 = 2$

$v_1 = 1$

$x_2 = 1$

$v_2 = 2$

After one jump, they are both at $x = 3$, ($x_1 + v_1 = 2 + 1$, $x_2 + v_2 = 1 + 2$), so the answer is YES.

Function Description

Complete the function kangaroo in the editor below.

kangaroo has the following parameter(s):

- int x_1 , int v_1 : starting position and jump distance for kangaroo 1
- int x_2 , int v_2 : starting position and jump distance for kangaroo 2

Returns

- string: either YES or NO

Input Format

A single line of four space-separated integers denoting the respective values of x_1 , v_1 , x_2 , and v_2 .

Constraints

- $0 \leq x_1 < x_2 \leq 10000$
- $1 \leq v_1 \leq 10000$
- $1 \leq v_2 \leq 10000$

Sample Input 0

```
0 3 4 2
```

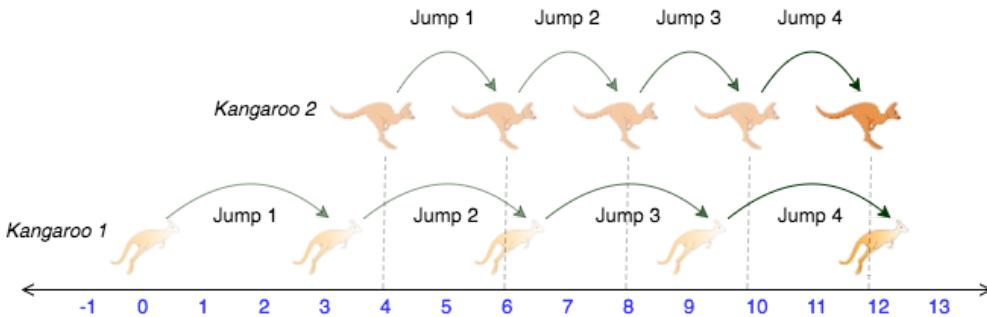
Sample Output 0

```
YES
```

Explanation 0

<https://www.hackerrank.com/challenges/kangaroo/problem>

The two kangaroos jump through the following sequence of locations:



From the image, it is clear that the kangaroos meet at the same location (number **12** on the number line) after same number of jumps (**4** jumps), and we print YES.

Sample Input 1

```
0 2 5 3
```

Sample Output 1

```
NO
```

Explanation 1

The second kangaroo has a starting location that is ahead (further to the right) of the first kangaroo's starting location (i.e., $x_2 > x_1$). Because the second kangaroo moves at a faster rate (meaning $v_2 > v_1$) and is already ahead of the first kangaroo, the first kangaroo will never be able to catch up. Thus, we print NO.





Viral Advertising ★

75 more points to get your next star!

Rank: 1055289 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



HackerLand Enterprise is adopting a new viral advertising strategy. When they launch a new product, they advertise it to exactly **5** people on social media.

On the first day, half of those **5** people (i.e., $\text{floor}(\frac{5}{2}) = 2$) like the advertisement and each shares it with **3** of their friends. At the beginning of the second day, $\text{floor}(\frac{5}{2}) \times 3 = 2 \times 3 = 6$ people receive the advertisement.

Each day, $\text{floor}(\frac{\text{recipients}}{2})$ of the recipients like the advertisement and will share it with **3** friends on the following day. Assuming nobody receives the advertisement twice, determine how many people have liked the ad by the end of a given day, beginning with launch day as day **1**.

Example

n = 5.

Day	Shared	Liked	Cumulative
1	5	2	2
2	6	3	5
3	9	4	9
4	12	6	15
5	18	9	24

The progression is shown above. The cumulative number of likes on the **5th** day is **24**.

Function Description

Complete the viralAdvertising function in the editor below.

viralAdvertising has the following parameter(s):

- int n: the day number to report

Returns

- int: the cumulative likes at that day

Input Format

A single integer, **n**, the day number.

Constraints

- $1 \leq n \leq 50$

Sample Input

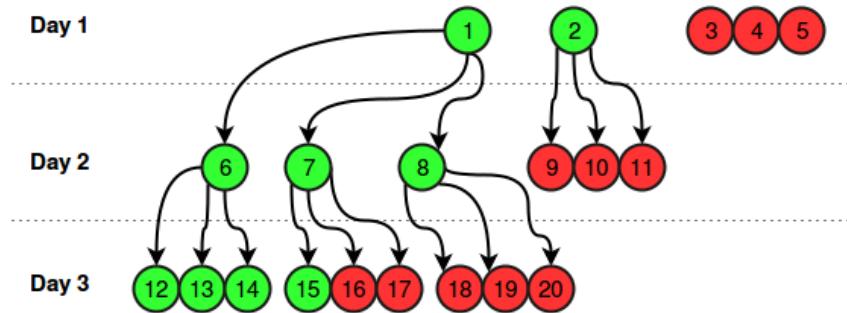
3

Sample Output

9

Explanation

This example is depicted in the following diagram:



2 people liked the advertisement on the first day, **3** people liked the advertisement on the second day and **4** people liked the advertisement on the third day, so the answer is **$2 + 3 + 4 = 9$** .





Recursive Digit Sum ★

75 more points to get your next star!

Rank: 1055289 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

RATE THIS CHALLENGE



We define super digit of an integer \mathbf{x} using the following rules:

Given an integer, we need to find the super digit of the integer.

- If \mathbf{x} has only 1 digit, then its super digit is \mathbf{x} .
- Otherwise, the super digit of \mathbf{x} is equal to the super digit of the sum of the digits of \mathbf{x} .

For example, the super digit of **9875** will be calculated as:

```
super_digit(9875)      9+8+7+5 = 29
super_digit(29)        2 + 9 = 11
super_digit(11)        1 + 1 = 2
super_digit(2)          = 2
```

Example

$\mathbf{n} = '9875'$

$\mathbf{k} = 4$

The number \mathbf{p} is created by concatenating the string \mathbf{n} \mathbf{k} times so the initial $\mathbf{p} = \mathbf{9875987598759875}$.

```
superDigit(p) = superDigit(9875987598759875)
                9+8+7+5+9+8+7+5+9+8+7+5+9+8+7+5 = 116
superDigit(p) = superDigit(116)
                1+1+6 = 8
superDigit(p) = superDigit(8)
```

All of the digits of \mathbf{p} sum to **116**. The digits of **116** sum to **8**. **8** is only one digit, so it is the super digit.

Function Description

Complete the function `superDigit` in the editor below. It must return the calculated super digit as an integer.

`superDigit` has the following parameter(s):

- string n : a string representation of an integer
- int k : the times to concatenate \mathbf{n} to make \mathbf{p}

Returns

- int: the super digit of \mathbf{n} repeated \mathbf{k} times

Input Format

The first line contains two space separated integers, \mathbf{n} and \mathbf{k} .

Constraints

- $1 \leq n < 10^{100000}$
- $1 \leq k \leq 10^5$

Sample Input 0

<https://www.hackerrank.com/challenges/recursive-digit-sum/problem>

148 3

Sample Output 0

3

Explanation 0

Here $n = 148$ and $k = 3$, so $p = 148148148$.

```
super_digit(P) = super_digit(148148148)
    = super_digit(1+4+8+1+4+8+1+4+8)
    = super_digit(39)
    = super_digit(3+9)
    = super_digit(12)
    = super_digit(1+2)
    = super_digit(3)
    = 3
```

Sample Input 1

9875 4

Sample Output 1

8

Sample Input 2

123 3

Sample Output 2

9

Explanation 2

Here $n = 123$ and $k = 3$, so $p = 123123123$.

```
super_digit(P) = super_digit(123123123)
    = super_digit(1+2+3+1+2+3+1+2+3)
    = super_digit(18)
    = super_digit(1+8)
    = super_digit(9)
    = 9
```





[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Insertion Sort - Part 1 ★

75 more points to get your next star!

Rank: 1055290 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)

RATE THIS CHALLENGE



Sorting

One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms.

Insertion Sort

These challenges will cover Insertion Sort, a simple and intuitive sorting algorithm. We will first start with a nearly sorted list.

Insert element into sorted list

Given a sorted list with an unsorted number **e** in the rightmost cell, can you write some simple code to insert **e** into the array so that it remains sorted?

Since this is a learning exercise, it won't be the most efficient way of performing the insertion. It will instead demonstrate the brute-force method in detail.

Assume you are given the array **arr = [1, 2, 4, 5, 3]** indexed **0 . . . 4**. Store the value of **arr[4]**. Now test lower index values successively from **3** to **0** until you reach a value that is lower than **arr[4]**, at **arr[1]** in this case. Each time your test fails, copy the value at the lower index to the current index and print your array. When the next lower indexed value is smaller than **arr[4]**, insert the stored value at the current index and print the entire array.

Example

n = 5

arr = [1, 2, 4, 5, 3]

Start at the rightmost index. Store the value of **arr[4] = 3**. Compare this to each element to the left until a smaller value is reached. Here are the results as described:

```
1 2 4 5 5
1 2 4 4 5
1 2 3 4 5
```

Function Description

Complete the insertionSort1 function in the editor below.

insertionSort1 has the following parameter(s):

- **n**: an integer, the size of **arr**
- **arr**: an array of integers to sort

Returns

- None: Print the interim and final arrays, each on a new line. No return value is expected.

Input Format

The first line contains the integer **n**, the size of the array **arr**.

The next line contains **n** space-separated integers **arr[0] . . . arr[n - 1]**.

Constraints

<https://www.hackerrank.com/challenges/insertionsort1/problem>

$1 \leq n \leq 1000$

$-10000 \leq arr[i] \leq 10000$

Output Format

Print the array as a row of space-separated integers each time there is a shift or insertion.

Sample Input

```
5
2 4 6 8 3
```

Sample Output

```
2 4 6 8 8
2 4 6 6 8
2 4 4 6 8
2 3 4 6 8
```

Explanation

3 is removed from the end of the array.

In the **1st** line **8 > 3**, so **8** is shifted one cell to the right.

In the **2nd** line **6 > 3**, so **6** is shifted one cell to the right.

In the **3rd** line **4 > 3**, so **4** is shifted one cell to the right.

In the **4th** line **2 < 3**, so **3** is placed at position **1**.

Next Challenge

In the [next Challenge](#), we will complete the insertion sort.

Change Theme Language Pypy 2



```
1 #!/bin/python
2
3 import math
4 import os
5 import random
6 import re
7 import sys
8
9 #
10 # Complete the 'insertionSort1' function below.
11 #
12 # The function accepts following parameters:
13 #   1. INTEGER n
14 #   2. INTEGER_ARRAY arr
15 #
16
17 def insertionSort1(n, arr):
18     # Write your code here
19
20 if __name__ == '__main__':
```

```
21     n = int(raw_input().strip())
22
23     arr = map(int, raw_input().rstrip().split())
24
```

Line: 26 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)



Insertion Sort - Part 2 ★

75 more points to get your next star!

Rank: 1055291 | Points: 125/200

[Problem](#)[Submissions](#)[Leaderboard](#)

RATE THIS CHALLENGE



In Insertion Sort Part 1, you inserted one element into an array at its correct sorted position. Using the same approach repeatedly, can you sort an entire array?

Guideline: You already can place an element into a sorted array. How can you use that code to build up a sorted array, one element at a time?

Note that in the first step, when you consider an array with just the first element, it is already sorted since there's nothing to compare it to.

In this challenge, print the array after each iteration of the insertion sort, i.e., whenever the next element has been inserted at its correct position. Since the array composed of just the first element is already sorted, begin printing after placing the second element.

Example.

$n = 7$

$arr = [3, 4, 7, 5, 6, 2, 1]$

Working from left to right, we get the following output:

```
3 4 7 5 6 2 1
3 4 7 5 6 2 1
3 4 5 7 6 2 1
3 4 5 6 7 2 1
2 3 4 5 6 7 1
1 2 3 4 5 6 7
```

Function Description

Complete the insertionSort2 function in the editor below.

insertionSort2 has the following parameter(s):

- int n: the length of arr
- int arr[n]: an array of integers

Prints

At each iteration, print the array as space-separated integers on its own line.

Input Format

The first line contains an integer, n , the size of arr .

The next line contains n space-separated integers $arr[i]$.

Constraints

$1 \leq n \leq 1000$

$-10000 \leq arr[i] \leq 10000, 0 \leq i < n$

Output Format

Print the entire array on a new line at every iteration.

Sample Input

<https://www.hackerrank.com/challenges/insertionsort2/problem>

STDIN	Function
-----	-----
6	n = 6
1 4 3 5 6 2	arr = [1, 4, 3, 5, 6, 2]

Sample Output

```
1 4 3 5 6 2
1 3 4 5 6 2
1 3 4 5 6 2
1 3 4 5 6 2
1 2 3 4 5 6
```

Explanation

Skip testing **1** against itself at position **0**. It is sorted.

Test position **1** against position **0**: **4 > 1**, no more to check, no change.

Print **arr**

Test position **2** against positions **1** and **0**:

- **3 < 4**, new position may be **1**. Keep checking.
- **3 > 1**, so insert **3** at position **1** and move others to the right.

Print **arr**

Test position **3** against positions **2, 1, 0** (as necessary): no change.

Print **arr**

Test position **4** against positions **3, 2, 1, 0**: no change.

Print **arr**

Test position **5** against positions **4, 3, 2, 1, 0**, insert **2** at position **1** and move others to the right.

Print **arr**

