* An *array* is a type of data structure that stores elements of the same type in a contiguous block of memory. In an array, A, of size N, each memory location has some unique index, i (where 0<=i<N), that can be referenced as A[i] or Ai.

Reverse an array of integers.

**Function Description:** Complete the function *reverseArray* in the editor below.

*reverseArray* has the following parameter(s):

* *int A[n]*: the array to reverse

**Returns:** *int[n]*: the reversed array

**Input Format:** The first line contains an integer, N, the number of integers in A.  
The second line contains N space-separated integers that make up A.

**Sample Input:** 4  
1 4 3 2

**Sample Output:**2 3 4 1

Ans:

import os

def reverseArray(a):

    return a[::-1]

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    arr\_count = int(input().strip())

    arr = list(map(int, input().rstrip().split()))

    res = reverseArray(arr)

    fptr.write(' '.join(map(str, res)))

    fptr.write('\n')

    fptr.close()

* Given a 6\*6 *2D Array*, arr:1 1 1 0 0 0

0 1 0 0 0 0

1 1 1 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

An hourglass in A is a subset of values with indices falling in this pattern in arr's graphical representation:a b c

d

e f g

There are 16 hourglasses in arr. An *hourglass sum* is the sum of an hourglass' values. Calculate the hourglass sum for every hourglass in arr, then print the *maximum* hourglass sum. The array will always be 6\*6.

**Function Description:** Complete the function *hourglassSum* in the editor below.

hourglassSum has the following parameter(s):

* *int arr[6][6]*: an array of integers

**Returns:** *int:* the maximum hourglass sum

**Input Format:** Each of the 6 lines of inputs arr[i] contains 6 space-separated integers arr[i][j].

**Output Format:** Print the largest (maximum) hourglass sum found in arr.

**Sample Input:** 1 1 1 0 0 0

0 1 0 0 0 0

1 1 1 0 0 0

0 0 2 4 4 0

0 0 0 2 0 0

0 0 1 2 4 0

**Sample Output:** 19

Ans:

import os

def hourglassSum(arr):

    return max(sum(arr[i][j:j+3]+[arr[i+1][j+1]]+arr[i+2][j:j+3]) for i in range(4) for j in range(4))

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    arr = []

    for \_ in range(6):

        arr.append(list(map(int, input().rstrip().split())))

    result = hourglassSum(arr)

    fptr.write(str(result) + '\n')

    fptr.close()

* A *left rotation* operation on an array of size n shifts each of the array's elements 1 unit to the left. Given an integer, d, rotate the array that many steps left and return the result.

**Example:** d=2

arr=[1,2,3,4,5]

After 2 rotations,arr’=[3,4,5,1,2] .

**Function Description:** Complete the *rotateLeft* function in the editor below.

*rotateLeft* has the following parameters:

* *int d:* the amount to rotate by
* *int arr[n]:* the array to rotate

**Returns:** *int[n]:* the rotated array

**Input Format:** The first line contains two space-separated integers that denote n, the number of integers, and ,d the number of left rotations to perform.  
The second line contains n space-separated integers that describe arr[].

**Sample Input:** 5 4

1 2 3 4 5

**Sample Output:** 5 1 2 3 4

Ans: import os

rotateLeft=lambda d,arr:arr[d:]+arr[:d]

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    first\_multiple\_input = input().rstrip().split()

    n = int(first\_multiple\_input[0])

    d = int(first\_multiple\_input[1])

    arr = list(map(int, input().rstrip().split()))

    result = rotateLeft(d, arr)

    fptr.write(' '.join(map(str, result)))

    fptr.write('\n')

    fptr.close()

* There is a collection of input strings and a collection of query strings. For each query string, determine how many times it occurs in the list of input strings. Return an array of the results.

**Function Description:** Complete the function *matchingStrings* in the editor below. The function must return an array of integers representing the frequency of occurrence of each query string in *stringList*.

matchingStrings has the following parameters:

* *string stringList[n]* - an array of strings to search
* *string queries[q]* - an array of query strings

**Returns:** *int[q]:* an array of results for each query

**Input Format:** The first line contains and integer n, the size of stringList[].  
Each of the next n lines contains a string stringList[i].  
The next line contains q, the size of queries[].  
Each of the next q lines contains a string queries[i].

**Sample Input:** 3  
def  
de  
fgh  
3  
de  
lmn  
fgh

**Sample Output:**1  
0  
1

Ans:

from collections import Counter

import os

def matchingStrings(stringList,queries):

    c=Counter(stringList)

    return [c[q] for q in queries]

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    stringList\_count = int(input().strip())

    stringList = []

    for \_ in range(stringList\_count):

        stringList\_item = input()

        stringList.append(stringList\_item)

    queries\_count = int(input().strip())

    queries = []

    for \_ in range(queries\_count):

        queries\_item = input()

        queries.append(queries\_item)

    res = matchingStrings(stringList, queries)

    fptr.write('\n'.join(map(str, res)))

    fptr.write('\n')

    fptr.close()