Given a 6×6 2D array, arr, an hourglass is a subset of values with indices falling in the following pattern:

```
a b c
  d
e f g
```

There are 16 hourglasses in a 6×6 array. The $hourglass\ sum$ is the sum of the values in an hourglass. Calculate the hourglass sum for every hourglass in arr, then print the $maximum$ hourglass sum.

Example

$arr =$

```
-9 -9 -9  1 1 1
 0 -9  0  4 3 2
-9 -9 -9  1 2 3
 0  0  8  6 6 0
 0  0  0 -2 0 0
 0  0  1  2 4 0
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

- Sample Test case 0
- Sample Test case 1
- Sample Test case 2

Input (stdin) Download

```
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
```

Your Output (stdout)

```
19
```

Expected Output Download

```
19
```

An array is a 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 $0 \leq i < N$), that can be referenced as $A[i]$ or A_i.

Your task is to reverse an array of integers.

Note: If you've already solved our C++ domain's Arrays Introduction challenge, you may want to skip this.

Example

$A = [1, 2, 3]$

Return $[3, 2, 1]$.

Function Description

Complete the function $reverseArray$ with 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.

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
4
1 4 3 2
```

Your Output (stdout)

```
2 3 4 1
```

Expected Output Download

```
2 3 4 1
```

- Declare a 2-dimensional array, arr, with n empty arrays, all zero-indexed.
- Declare an integer, $lastAnswer$, and initialize it to 0.

You need to process two types of queries:

1. Query: $1\ x\ y$
   - Compute $idx = (x \oplus lastAnswer)$.
   - Append the integer y to $arr[idx]$.
2. Query: $2\ x\ y$
   - Compute $idx = (x \oplus lastAnswer)$.
   - Set $lastAnswer = arr[idx][y\%size(arr[idx])]$.
   - Store the new value of $lastAnswer$ in an answers array.

Notes:

- $\oplus$ is the bitwise XOR operation, which corresponds to the ^ operator in most languages. Learn more about it on Wikipedia.
- $\%$ is the modulo operator.
- Finally, $size(arr[idx])$ is the number of elements in $arr[idx]$.

Function Description

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
2 5
1 0 5
1 1 7
1 0 3
2 1 0
2 1 1
```

Your Output (stdout)

```
7
3
```

A *left rotation* operation on a circular array shifts each of the array's elements 1 unit to the left. The elements that fall off the left end reappear at the right end. Given an integer d, rotate the array that many steps to the 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 with 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,

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
1  5 4
2  1 2 3 4 5
```

Your Output (stdout)

```
1  5 1 2 3 4
```

Expected Output Download

```
1  5 1 2 3 4
```

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.

Example

$stringList = ['ab', 'ab', 'abc']$

$queries = ['ab', 'abc', 'bc']$

There are **2** instances of 'ab', **1** of 'abc', and **0** of 'bc'. For each query, add an element to the return array: $results = [2, 1, 0]$.

Function Description

Complete the function $matchingStrings$ with the following parameters:

- $string\ stringList[n]$: an array of strings to search
- $string\ queries[q]$: an array of query strings

Returns

- $int[q]$: the results of each query

Input Format

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

- Sample Test case 0
- Sample Test case 1
- Sample Test case 2

Input (stdin) Download

```
4
aba
baba
aba
xzxb
3
aba
xzxb
ab
```

Your Output (stdout)

```
2
```