# PROBLEM STATEMENT

HackerRank Practice > Data Structures > Arrays > Dynamic Array

Problem | Submissions | Leaderboard | Discussions

- Create a list, $seqList$, of $n$ empty sequences, where each sequence is indexed from $0$ to $n-1$. The elements within each of the $n$ sequences also use $0$-indexing.
- Create an integer, $lastAnswer$, and initialize it to $0$.
- There are $2$ types of queries that can be performed on the list of sequences:
  1. Query: `1 x y`
     1. Find the sequence, $seq$, at index $( (x \oplus lastAnswer) \% n )$ in $seqList$.
     2. Append integer $y$ to sequence $seq$.
  2. Query: `2 x y`
     1. Find the sequence, $seq$, at index $( (x \oplus lastAnswer) \% n )$ in $seqList$.
     2. Find the value of element $y \% size$ in $seq$ (where $size$ is the size of $seq$) and assign it to $lastAnswer$.
     3. Print the new value of $lastAnswer$ on a new line

**Note:** $\oplus$ is the bitwise XOR operation, which corresponds to the `^` operator in most languages. Learn more about it on Wikipedia.  is the modulo operator.

**Function Description**

Complete the dynamicArray function below.

dynamicArray has the following parameters:
- int n: the number of empty sequences to initialize in $seqList$
- string queries[q]: an array of query strings

Editorial

**Returns**

- int[]: the results of each type 2 query in the order they are presented

**Input Format**

The first line contains two space-separated integers, $n$ (the number of sequences) and $q$ (the number of queries), respectively.
Each of the $q$ subsequent lines contains a query in the format defined above, $queries[i]$.

**Constraints**

- $1 \leq n, q \leq 10^5$
- $0 \leq x \leq 10^9$
- $0 \leq y \leq 10^9$
- It is guaranteed that query type $2$ will never query an empty sequence or index.

**Sample Input**

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

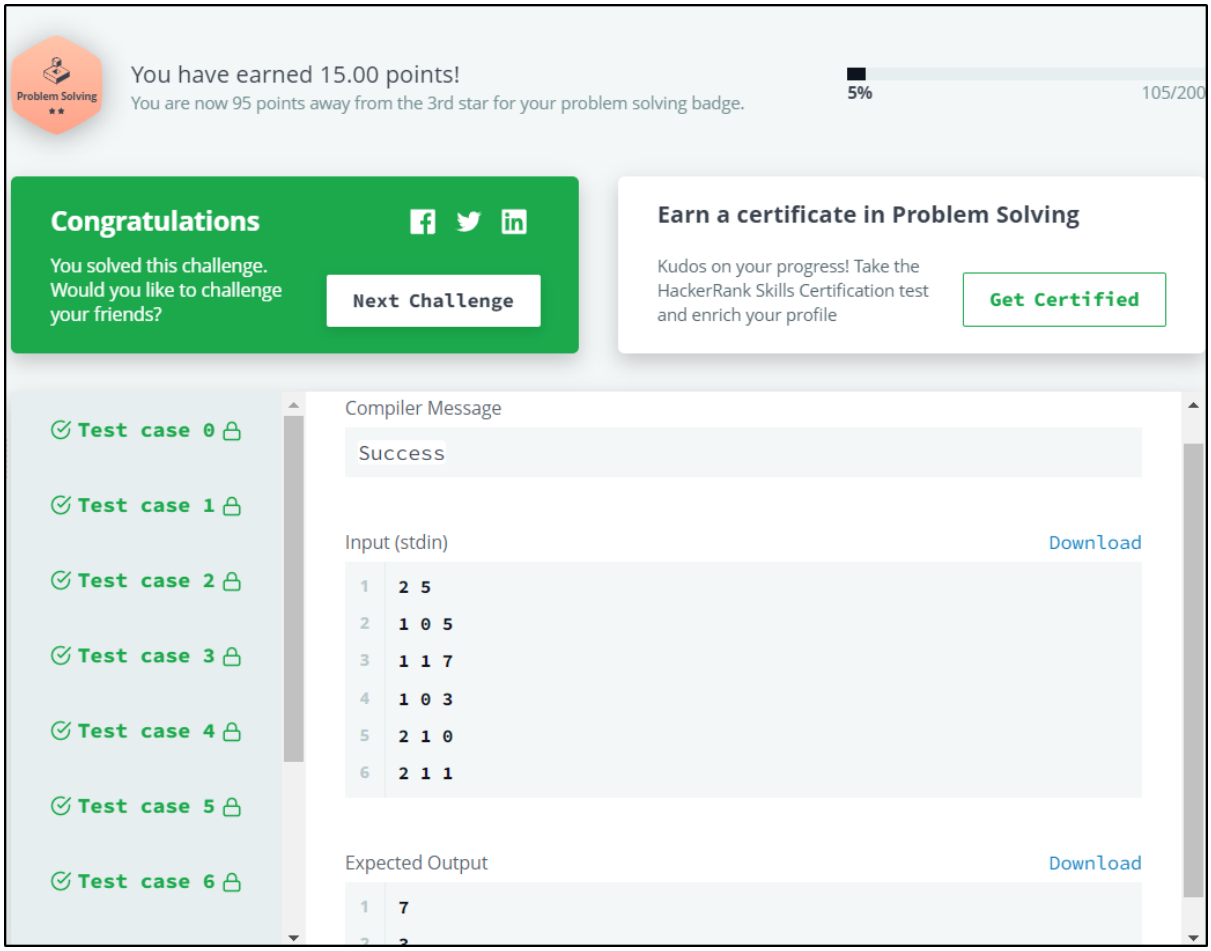
**Sample Output**

```
7
3
```

# PROGRAM USED TO SOLVE THE PROBLEM STATEMENT

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main(void)
{
    int N, Q, seq, lastAns = 0, x, y;
    scanf("%i %i",&N,&Q);

    int **s = calloc(100000, sizeof(int*));
    for (int i= 0; i < N; i++)
         s[i] = calloc(5, sizeof(int));

    int *count = calloc(100000, sizeof(int));

    for (int i = 0; i < Q; i++)
    {
    scanf("%i %i %i", &seq, &x, &y);
    int seqN = (x ^ lastAns) % N;
    if (seq == 1)
    {
        s[seqN][count[seqN]] = y;
        count[seqN]++;
    }
    else if (seq == 2)
    {
        lastAns = s[seqN][y % count[seqN]];
        printf("%i\n", lastAns);
    }
    }
    free(s);
    free(count);
    return 0;
}
```

# TEST CASES


You have earned 15.00 points!
You are now 95 points away from the 3rd star for your problem solving badge.
5%
105/200
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Earn a certificate in Problem Solving
Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile
Get Certified
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
2 5
1 0 5
1 1 7
1 0 3
2 1 0
2 1 1
Expected Output
Download
7