

Amazon Questions

1)

The screenshot shows a web-based interview question interface. At the top, it says 'Question 1' and 'Max. score: 100.00'. The title of the question is 'Odd queries'. The problem statement is: 'You are given an array of integers *Arr*. You are asked *Q* queries of two types:'. There are two types of queries listed: 1. *l* *v*: change *i*th element of *Arr* to *v*; 2. *l* *r*: Determine the count of integers that have an odd count in range *l* to *r*. Below this is the 'Task' section, which says: 'For every query of type 2, you have to tell the number of elements in the range from *l* to *r* with an odd count in that range.' A note specifies: 'Note: Assume 1-based Indexing.' There are sections for 'Explanation' and 'Assumptions'. Under 'Assumptions', it lists: *N* = 4 and *Arr* = [1, 2, 3, 2]. On the right side of the interface, there is a vertical list of numbers from 1 to 17, which likely represent different test cases or sub-questions. A 'New Sub' button is visible at the top right of the interface.

Question 1 Max. score: 100.00

Odd queries

You are given an array of integers *Arr*. You are asked *Q* queries of two types:

- 1 *l* *v*: change *i*th element of *Arr* to *v*
- 2 *l* *r*: Determine the count of integers that have an odd count in range *l* to *r*

Task

For every query of type 2, you have to tell the number of elements in the range from *l* to *r* with an odd count in that range.

Note: Assume 1-based Indexing.

Explanation

Assumptions

- N* = 4
- Arr* = [1, 2, 3, 2]

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

2)

Question 2 Max. score: 100.00

Magical tube

Bob loves to play with the tubes, which are closed from one end. Bob gives you the tube with some queries Q . Each query contains:

- Two integers X and N
- If $X = 1$, insert number N into the tube from the top
- If $X = 2$, take out an integer that is occurring the most (i.e. have the largest frequency in the tube) and remove its occurrence closest to the open end of the tube. If the frequency of more than one number is the same and largest, you should remove the number closest to the open end of the tube. In this case $N = -1$

Task

Return the list of elements that are removed during the $X = 2$ query.

Example

Assumptions

Task

Return the list of elements that are removed during the $X = 2$ query.

Example

Assumptions

- $Q = 6$
- query = $[(1,2),(1,4),(1,3),(1,2),(2,-1),(2,-1)]$

Approach

- After the first operation, the tube will be 2.
- After the second operation, the tube will be 2, 4.
- After the third operation, the tube will be 2, 4, and 3.
- After the fourth operation, the tube will be 2, 4, 3, and 2.
- After the fifth operation, 2 will be removed as 2 occurs most times, and the tube will be 2, 4, 3.
- After the sixth operation, 3 will be removed as 2, 4, 3.

Approach

- After the first operation, the tube will be 2.
- After the second operation, the tube will be 2, 4.
- After the third operation, the tube will be 2, 4, and 3.
- After the fourth operation, the tube will be 2, 4, 3, and 2.
- After the fifth operation, 2 will be removed as 2 occurs most times, and the tube will be 2, 4, 3.
- After the sixth operation, 3 will be removed as 2, 4, 3 occurs most times, but 3 is more close to the open side of the tube. The tube will be: 2, 4.
- Therefore, you print $[2, 3]$ elements removed during the $X = 2$ query.

Function description

Complete the `elements` function provided in the editor. This function takes the following 2 parameters and returns the list of elements that are removed during the $X = 2$ query:

- Q : Represents the number of queries
- query: Represents the queries

Function description

Complete the `elements` function provided in the editor. This function takes the following 2 parameters and returns the list of elements that are removed during the $X = 2$ query:

- Q : Represents the number of queries
- query: Represents the queries

Input format

Note: This is the input format that you must use to provide custom Input (available above the Compile and Test button).

- The first line contains a single integer Q denoting the number of queries.
- The next Q lines contain two space-separated integers X and N .

Output format

Print the list of elements in a space-separated format that are removed during the $X = 2$ query.

Constraints

$1 < Q < 10^4$



Constraints

$1 \leq Q \leq 10^4$
 $1 \leq X \leq 2$


1 $0 \leq N \leq 10^9$, if $X = 1$
 $N = -1$, if $X = 2$

2 Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input  

```
3
1 62
1 49
2 -1
```

Sample output 

```
49
```

Explanation

Explanation

Given

1 $Q = 3$
 $query = [(1, 62), (1, 49), (2, -1)]$

2 Approach

- After the first operation, the tube will be 62.
- After the second operation, the tube will be 62, 49.
- After the third operation, 49 will be removed as 62, 49 occurs most times, but 49 is more close to the open side of the tube.
- Therefore, you print [49] element removed during the $X = 2$ query.

3)

Minimum addition

1 You are given a positive integer N .

2 You have to divide the number N into two numbers $P1$ and $P2$ such that:

- The sum of the number of digits in $P1$ and $P2$ is equal to the number of digits in N .
- The sum of the frequency of each digit in $P1$ and $P2$ is equal to the frequency of that digit in N .
- The sum of the numbers $P1$ and $P2$ is minimum.

Task

Determine the minimum possible sum of $P1$ and $P2$.

Note: The order of occurrence of the digits in $P1$ and $P2$ may differ from the order of occurrence of the digits in N .

Example

Assumption

- $N = 4325$

Approach

- $N = 4325$

Approach

1

- $P1$ can be 25
- $P2$ can be 34

2

- The minimum sum of $P1$ and $P2$ is 59.
- Note there are other possibilities of $P1$ and $P2$ like (43,25), (3,245), (53,24), etc. but their sum is not minimum.
- Thus the answer is 59.

Function description

Complete the function `solve` provided in the editor. This function takes the following parameter and returns the required answer:

- N : Represents the positive integer

Input format Note: This is the input format that you must use to provide custom input (available above the **Compile and Test** button).

- The first line contains T denoting the number of test cases. T also specifies the number of times you have to run the `solve` function on a different set of inputs.
- For each test case:
 - The first line contains an integer N .

The first line contains an integer T .

Output format

1 For each test case, print the minimum possible sum of the two numbers $P1$ and $P2$.

2 **Constraints**

$$1 \leq T \leq 10^5$$
$$10 \leq N \leq 2 \times 10^{18}$$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input [↗](#)

```
2
4325
867
```

Sample output [↗](#)

```
59
75
```

Explanation

1 The first line contains the number of test cases, $T = 2$.

The first test case

2 This is the same as the example. Please refer to that.

The second test case

Given:

- $N = 867$

Approach:

- $P1$ can be 68
- $P2$ can be 7
- Minimum sum of $P1$ and $P2$ is 75.
- Note there are other possibilities of $P1$ and $P2$ also like (6,78), (6,87), etc. but their sum is not minimum.
- Thus the answer is 75.

[ⓘ](#) The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Sample input 1 [↗](#)

Sample output 1 [↗](#)

Sample input 1		Sample output 1	
1	5	67	
2	4891	65	
	8732	17	
	9008	21	
	516	102	
	4755		
Sample input 2		Sample output 2	
	15	52	
	1357	107	
	2798	137	
	6609	55	
	7228	40	
	7132	94	
	6628	28	
	4428	103	
	7654	29	
	272	24	

Solutions:

1)

```
3 int freq[10001];
4 vector<int> solve (int N, int Q, vector<int> arr, vector<vector<int> > q) {
5     vector<int> ans;
6     for(auto x:q){
7         if(x[0] == 1){
8             arr[x[1]-1] = x[2];
9         }else{
10             int l = x[1] - 1;
11             int r = x[2] - 1;
12             memset(freq,0,sizeof(freq));
13             for(int i=l;i<=r;i++){
14                 freq[arr[i]]++;
15             }
16             int cnt = 0;
17             for(int i=0;i<10001;i++){
18                 cnt += (freq[i]&1);
19             }
20             ans.push_back(cnt);

```

```
        int l = x[1] - 1;
        int r = x[2] - 1;
        memset(freq,0,sizeof(freq));
        for(int i=l;i<=r;i++){
            freq[arr[i]]++;
        }
        int cnt = 0;
        for(int i=0;i<10001;i++){
            cnt += (freq[i]&1);
        }
        ans.push_back(cnt);
    }
    return ans;
}

int main() {

```

2)

```
4  vector<int> elements (int q, vector<vector<int> > query) {  
5      map<int, int> frq;  
6      vector<int> ans, v;  
7      v.push_back(0);  
8      int maxi= -1;  
9      for(auto x: query){  
10         if(x[0]==1){  
11             v.push_back(x[1]);  
12             frq[x[1]]++;  
13             maxi= max(maxi, frq[x[1]]);  
14         }  
15         else{  
16             for(int i= v.size()-1; i; i--){  
17                 if(frq[v[i]]== maxi){  
18                     frq[v[i]]--;  
19                     ans.push_back(v[i]);  
20                     v[i]=-1;  
21                     break;  
22                 }  
23             }  
24             for(auto it: frq){  
25                 maxi= max(0, it.second);  
26             }  
27         }  
28     }  
29     return ans;  
30 }
```


3)

```
include<bits / stdc++.h>
using namespace std;

int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        string s;
        cin >> s;
        int n1 = 0;
        int n2 = 0;

        sort(s.begin(), s.end());

        int i = 0;

        while (i < s.size())
        {
            if (i % 2 == 0)
                n1 = n1 * 10 + s[i] - '0';

            else
                n2 = n2 * 10 + s[i] - '0';
            i++;
        }
        cout << n1 + n2 << endl;
    }

    return 0;
}
```