**Minimum Multiplications to reach End: -**

Medium Accuracy: 48.94% Submissions: 47K+ Points: 4

Given **start**, **end** and an array **arr** of **n** numbers. At each step, **start** is multiplied with any number in the array and then mod operation with **100000** is done to get the new start.

Your task is to find the minimum steps in which **end** can be achieved starting from **start**. If it is not possible to reach **end**, then return **-1**.

**Example 1:**

**Input:**

arr[] = {2, 5, 7}

start = 3, end = 30

**Output:**

2

**Explanation:**

Step 1: 3\*2 = 6 % 100000 = 6

Step 2: 6\*5 = 30 % 100000 = 30

**Example 2:**

**Input:**

arr[] = {3, 4, 65}

start = 7, end = 66175

**Output:**

4

**Explanation:**

Step 1: 7\*3 = 21 % 100000 = 21

Step 2: 21\*3 = 63 % 100000 = 63

Step 3: 63\*65 = 4095 % 100000 = 4095

Step 4: 4095\*65 = 266175 % 100000 = 66175

**Your Task:**You don't need to print or input anything. Complete the function **minimumMultiplications()** which takes an integer array **arr**, an integer **start** and an integer**end** as the input parameters and returns an integer, denoting the minumum steps to reach in which **end** can be achieved starting from **start**.

**Expected Time Complexity:** O(105)  
**Expected Space Complexity:** O(105)

**Constraints:**

* 1 <= n <= 104
* 1 <= arr[i] <= 104
* 1 <= start, end < 105

**Code: -**

//{ Driver Code Starts

// Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// User function Template for C++

class Solution {

public:

int minimumMultiplications(vector<int>& arr, int start, int end) {

if(start == end)

return 0;

int mod = 100000;

vector<int> vis(mod, -1);

queue<int> q;

q.push(start);

vis[start] = 0;

while(q.size()){

int count = q.size();

while(count--){

int front = q.front();

q.pop();

for(auto &i : arr){

int newstart = (front \* i) % mod;

// end found immediately

if(newstart == end)

return vis[front]+1;

// newstart is not visited

if(vis[newstart] == -1){

q.push(newstart);

vis[newstart] = vis[front] + 1;

}

}

}

}

return -1;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

vector<int> arr(n);

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int start, end;

cin >> start >> end;

Solution obj;

cout << obj.minimumMultiplications(arr, start, end) << endl;

}

}

// } Driver Code Ends

**T.C: - O(105)**

**S.C: - O(105)**