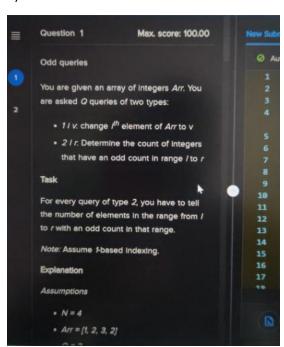
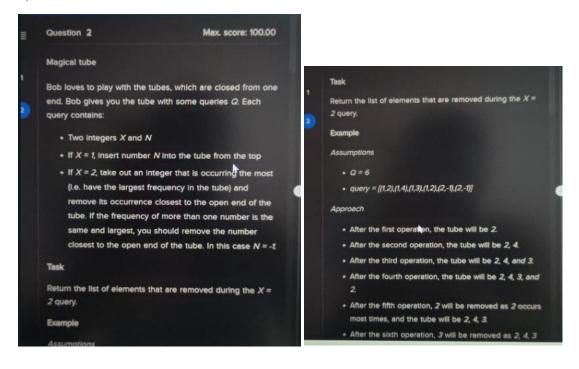
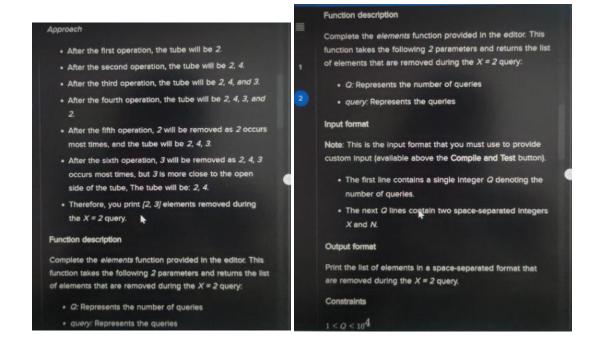
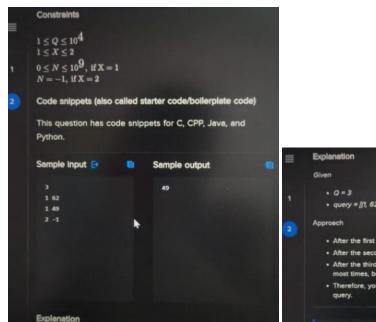
## **Amazon Questions**

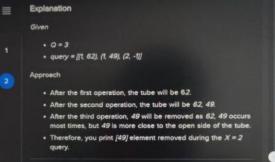
1)

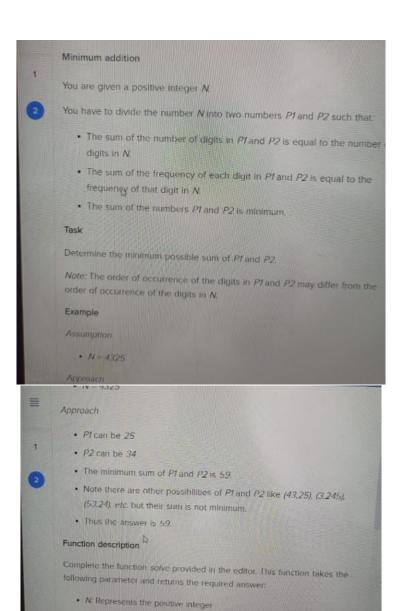












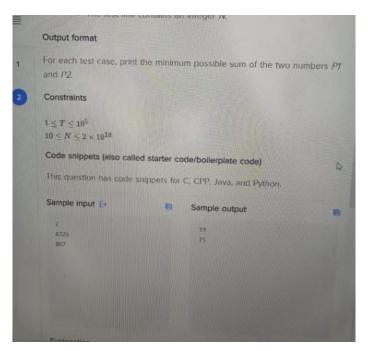
Input format Note: This is the input format that you must use to provide custom input (available above the Compile and Test button).

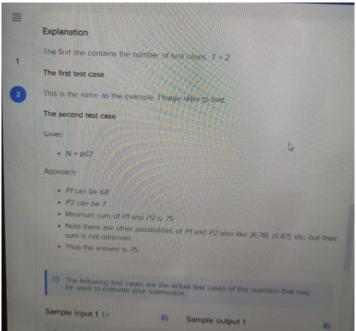
The first line contains 7 denoting the number of test cases. 7 also specifies the number of times you have to run the solve function on a

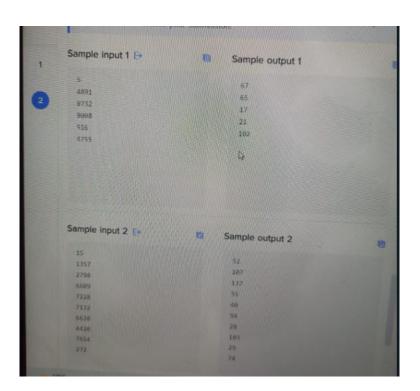
different set of inputs

• For each test case:

The first line contains an integer N.







## Solutions:

1)

```
int freq[10001];
     vector(int) solve (int N, int Q, vector(int) arr, vector(vector(int) > q) {
        vector(int>ans;
for(auto x:q){
    if(x[0] == 1){
                  arr[x[1]-1] = x[2];
             }else{
                  int 1 = x[1] - 1;
                  int r = x[2] - 1;
                  memset(freq,0,sizeof(freq));
                  for(int i=1;i<=r;i++){
13
                       freq[arr[i]]++;
14
15
                  int cnt = 0;
16
                  for(int i=0;i<10001;i++){
17
                       cnt += (freq[i]&1);
18
19
                  ans.push back(cnt);
```

```
int 1 = x[1] - 1;
int r = x[2] - 1;
memset(freq,0,sizeof(freq));
for(int i=1;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() {</pre>
```

```
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())</pre>
             if (i % 2 == 0)
                 n1 = n\overline{1 * 10 + s[i]} - '0';
             else
                n2 = n2 * 10 + s[i] - '0';
             i++;
        cout << n1 + n2 << endl;
    }
    return 0;
```