LAB PROBLEMS

Problem 1

Agastya is currently standing at stair 0 and he wants to reach stair numbered X. Agastya can climb either Y steps or 1 step in one move. Find the minimum number of moves required by him to reach exactly the stair numbered X.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of a single line of input containing two space separated integers X and Y denoting the number of stairs Agastya wants to reach and the number of stairs he can climb in one move.

Output Format

For each test case, output the minimum number of moves required by him to reach exactly the stair numbered X.

Constraints

- 1≤≤T≤500
- 1≤X, Y≤100

Sample

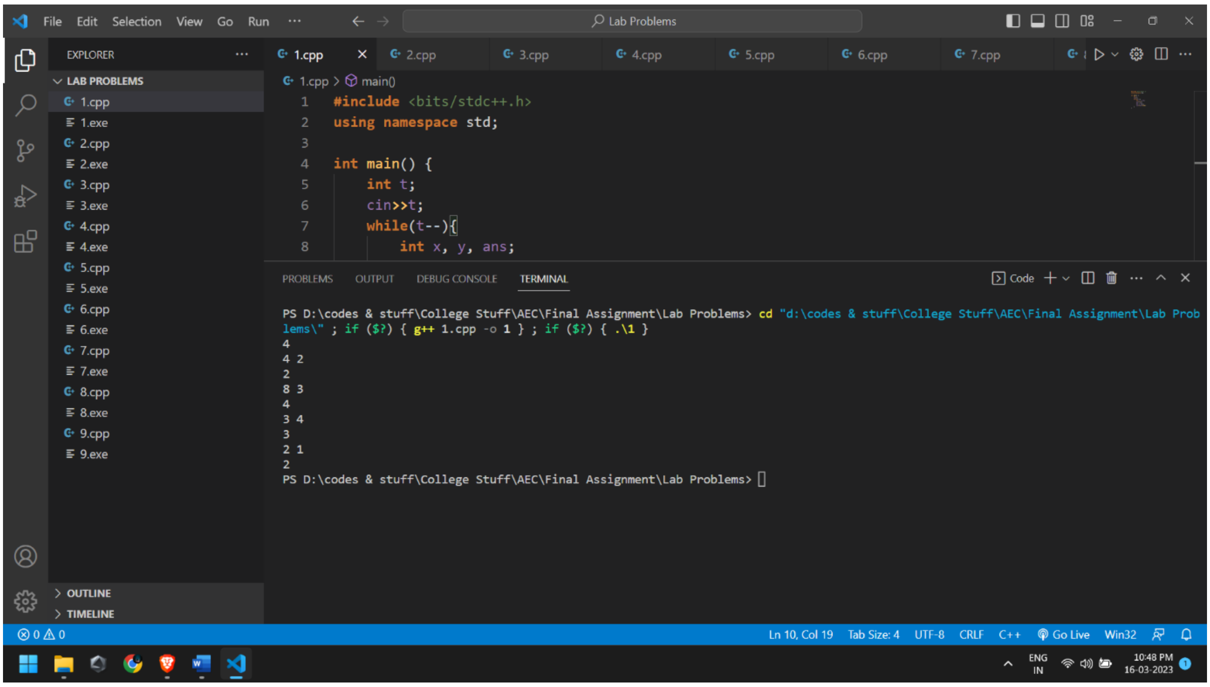
Input	**Output**

4
4 2
8 3
3 4
2 1
	2
	4
	3
	2

Solution

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

int main() {
    int testcases;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                         //Entering the number of Test Cases
    while(testcases--){
        int x, y, ans;
        cout<<"Enter two Integers: ";
        cin>>x>>y;                          //Entering the stairs to be reached & the stairs he can climb per move
        ans = x/y;
        ans += (x%y);
        cout<<ans<<endl;
    }
}
```



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

int main() {
    int t;
    cin>>t;
    while(t--){
        int x, y, ans;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?) { .\1 }
4
4 2
2
8 3
4
3 4
3
2 1
2
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 2

You are given the height H (in metres) and mass M (in kilograms). The Body Mass Index (BMI) of a person is computed as **M/H^2**. Report the category into which a person falls, based on his BMI:

• Category 1: Underweight if BMI ≤ 18

• Category 2: Normal weight if BMI ∈ {19, 20, … , 24}

• Category 3: Overweight if BMI ∈ {25, 26, … , 29}

• Category 4: Obesity if BMI ≥ 30

Input:

- The first line of input will contain an integer, TT, which denotes the number of testcases. Then the testcases follow.
- Each testcase contains a single line of input, with two space separated integers, M, H which denote the mass and height of Agastya respectively.

Output:

For each testcase, output in a single line, 1, 2, 3 or 4, based on the category in which Agastya falls.

Constraints

- 1 ≤ T≤ 2 ∗ 10^4
- 1 ≤M ≤ 10^4
- 1 ≤H ≤ 10^2
- It is guaranteed that H^2 divides M.

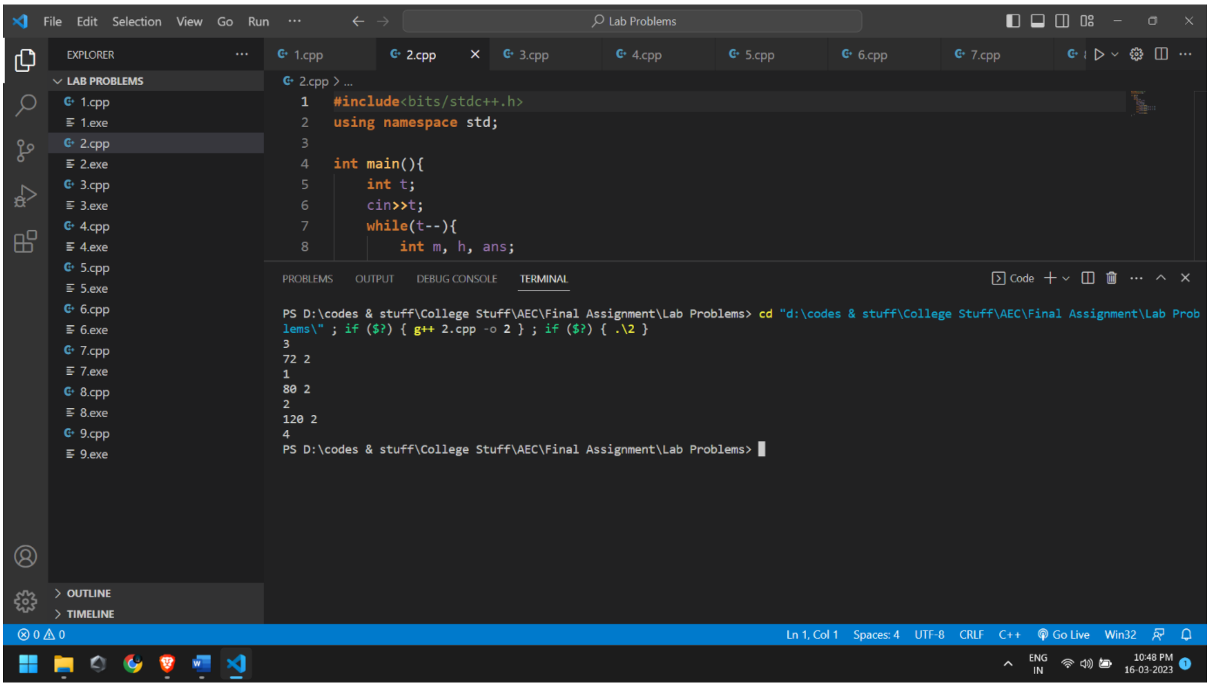
Sample

Input	Output
3	1
72 2	2
80 2	4
120 2	

Solution

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

int main(){
    int testcases;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                         //Entering the number of
Test Cases
    while(testcases--){
        int m, h, ans;
        cout<<"Enter the mass and height: ";
        cin>>m>>h;
        ans = m/(h*h);                      //Calculating the Body-
Mass Index
        if(ans <= 18)
            cout<<1<<endl;
        else if(ans >= 19 && ans <= 24)     //Categorising the BMI
            cout<<2<<endl;
        else if(ans >= 25 && ans <= 29)
            cout<<3<<endl;
        else
            cout<<4<<endl;
    }
}
```



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

int main(){
    int t;
    cin>>t;
    while(t--){
        int m, h, ans;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }
3
72 2
1
80 2
2
120 2
4
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 3

You will be given an array of N integers and you have to print the integers in the reverse order.

Input Format

The first line of the input contains N, where N is the number of integers. The next line contains N space-separated integers.

Constraints

- 1≤N≤1000
- 1≤A[i]≤10000 where A[i] is the ith integer in the array.
- **Sample**

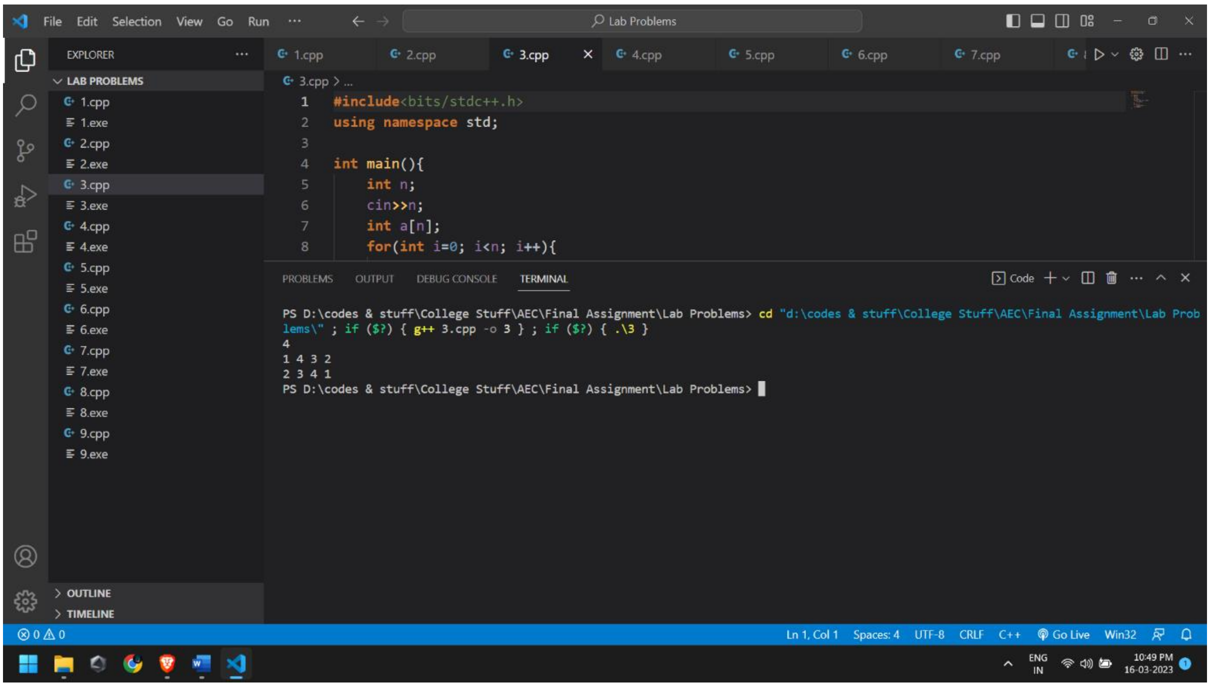
Input	**Output**
4 1 4 3 2	2 3 4 1

Solution

```cpp
#include<bits/stdc++.h>
```

```
using namespace std;

int main(){
    int n;
    cout<<"Enter the number of Integers: ";
    cin>>n;                                     //Entering the number of
Integers
    int a[n];
    for(int i=0; i<n; i++){
        cin>>a[i];                              //Entering the Integers
into an Array
    }
    while(n--)
        cout<<a[n]<<" ";                        //Printing the Integers
in the reverse order
    cout<<endl;
}
```



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

int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0; i<n; i++){
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 3.cpp -o 3 } ; if ($?) { .\3 }
4
1 4 3 2
2 3 4 1
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 4

Bhuvan has a string S with him. Bhuvan is happy if the string contains a contiguous substring of length strictly greater than 2 in which all its characters are vowels. Determine whether Bhuvan is happy or not. Note that, in English alphabet, vowels are a, e, i, o, and u.

Input Format

- First line will contain T, number of test cases. Then the test cases follow.
- Each test case contains of a single line of input, a string S.

Output Format

- For each test case, if Bhuvan is happy, print HAPPY else print SAD.
- You may print each character of the string in uppercase or lowercase (for example, the strings hAppY, Happy, haPpY, and HAPPY will all be treated as identical).

Constraints

- $1 \leq S \leq 1000$
- $3 \leq |S| \leq 1000$, where |S| is the length of S.

S will only contain lowercase English letters.

Sample

Input	**Output**
4 aeiou abxy aebcdefghij abcdeeafg	Happy Sad Sad Happy

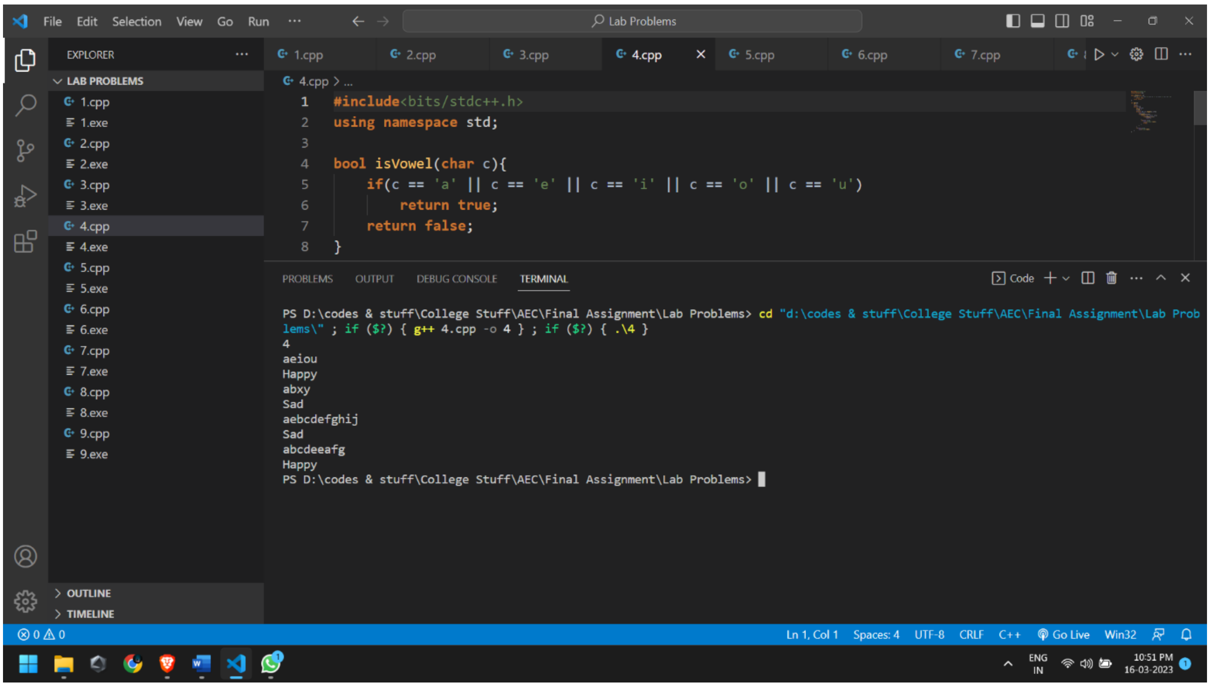
Solution

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

```

```
bool isVowel(char c){                       //Function to check
whether a given character is a Vowel or not
    if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
        return true;
    return false;
}

int main(){
    int testcases;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                         //Entering the number of
test cases
    while(testcases--){
        int count=0;
        string s;                           //Entering the String
        cout<<"Enter the String: ";
        cin>>s;
        for(int i=0; i<s.length(); i++){
            count = 0;
            if(isVowel(s[i])){
                for(int j=i; j<i+3; j++){
                    if(isVowel(s[j]))
                        count++;            //Counting the number of
vowels
                }
                if(count > 2){
                    cout<<"Happy"<<endl;
                    break;
                }
            }                               //If the number of
Vowels is greater than 2, Bhuvan is HAPPY or else SAD
        }
        if(count == 0)
            cout<<"Sad"<<endl;
    }
}
```



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

bool isVowel(char c){
    if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
        return true;
    return false;
}
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 4.cpp -o 4 } ; if ($?) { .\4 }
4
aeiou
Happy
abxy
Sad
aebcdefghij
Sad
abcdeeafg
Happy
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 5

There is a group of N friends who wish to enrol in a course together. The course has a maximum capacity of M students that can register for it. If there are K other students who have already enrolled in the course, determine if it will still be possible for all the N friends to do so or not.

Input Format

- The first line contains a single integer T - the number of test cases. Then the test cases follow.
- Each test case consists of a single line containing three integers N, M and K - the size of the friend group, the capacity of the course and the number of students already registered for the course.

Output Format

- For each test case, output Yes if it will be possible for all the N friends to register for the course. Otherwise output No.

- You may print each character of Yes and No in uppercase or lowercase (for example, yes, yEs, YES will be considered identical).

Constraints

- 1 ≤ M ≤ 1000
- 1 ≤ K≤ N ≤ 100
- 0 ≤ K ≤ N

Sample

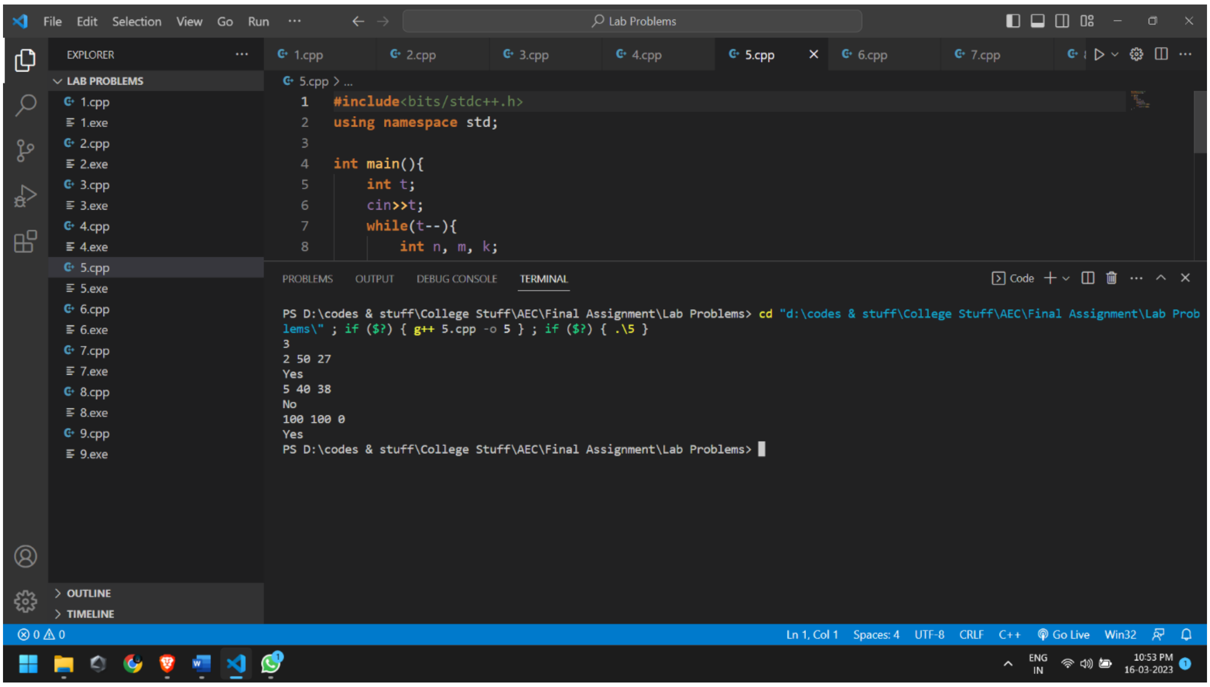
Input	Output

3
2 50 27
5 40 38
100 100 0
	Yes
	No
	Yes

Solution

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

int main(){
    int testcases;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                             //Entering number of
test cases
    while(testcases--){
        int n, m, k;
        cin>>n>>m>>k;                           //Entering number of
friends, maximum capacity in the course & enrolled students
        if((n+k) <= m)
            cout<<"Yes"<<endl;
        else
            cout<<"No"<<endl;
    }
}
```



File Edit Selection View Go Run ···
Lab Problems
EXPLORER
LAB PROBLEMS
1.cpp
1.exe
2.cpp
2.exe
3.cpp
3.exe
4.cpp
4.exe
5.cpp
5.exe
6.cpp
6.exe
7.cpp
7.exe
8.cpp
8.exe
9.cpp
9.exe
OUTLINE
TIMELINE
1.cpp 2.cpp 3.cpp 4.cpp 5.cpp 6.cpp 7.cpp
5.cpp > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Go Live Win32
ENG IN 10:53 PM 16-03-2023

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

int main(){
    int t;
    cin>>t;
    while(t--){
        int n, m, k;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 5.cpp -o 5 } ; if ($?) { .\5 }
3
2 50 27
Yes
5 40 38
No
100 100 0
Yes
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 6

Chetan is playing a videogame, and is getting close to the end. He decides to finish the rest of the game in a single session. There are X levels remaining in the game, and each level takes Chetan Y minutes to complete. To protect against eye strain, Chetan also decides that every time he completes 3 levels, he will take a Z minute break from playing. Note that there is no need to take this break if the game has been completed. How much time (in minutes) will it take Chetan to complete the game?

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- The first and only line of input will contain three space-separated integers X, Y, and Z.

Output Format

For each test case, output on a new line the answer — the length of Chetan’s gaming session.

Constraints

- 1≤T≤100
- 1≤X≤100
- 5≤Y≤100

Sample

Input	Output
4	24
2 12 10	36
3 12 10	156
7 20 8	1185
24 45 15	

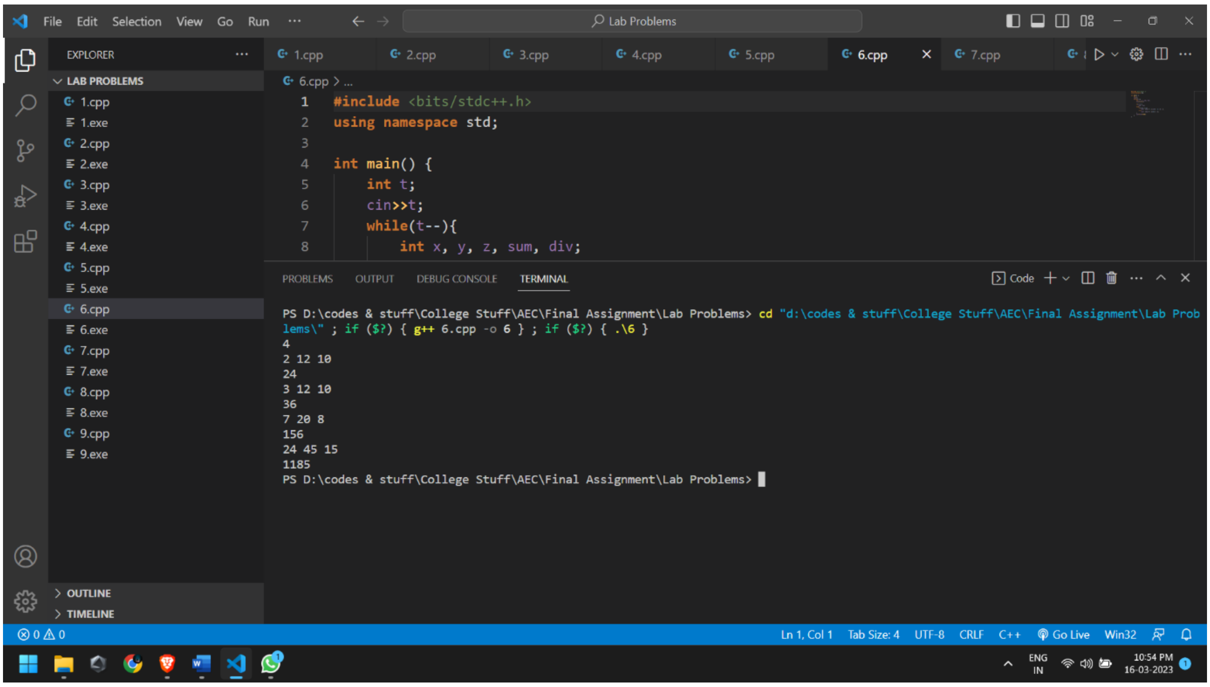
Solution

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

int main() {
    int testcases;
    cin>>testcases;                         //Entering the number of
test cases
    while(testcases--){
        int x, y, z, sum, div;
        cin>>x>>y>>z;                       //Entering the total
levels, minutes to complete & break-time

        if(x <= 3)
            sum = x*y;
        else{
            if((x%3) == 0)
                sum = (x*y) + ( ( (x/3) - 1 ) * z );        //If
number of levels are divisible by 3, then the last break-time is not
taken by Chetan
```

```
            else
                sum = (x*y) + ( (x/3) * z );                //Else
calculate the total time taken
        }
        cout<<sum<<endl;
    }
}
```



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

int main() {
    int t;
    cin>>t;
    while(t--){
        int x, y, z, sum, div;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 6.cpp -o 6 } ; if ($?) { .\6 }
4
2 12 10
24
3 12 10
36
7 20 8
156
24 45 15
1185
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 7

Dhruv is very fond of horses. He enjoys watching them race. As expected, he has a stable full of horses. He, along with his friends, goes to his stable during the weekends to watch a few of these horse race. Agastya wants his friends to enjoy the race and so he wants the race to be close. This can happen only if the horses are comparable on their skill i.e., the difference in their skills is less.

There are N horses in the stable. The skill of the horse i is represented by an integer S[i]. Dhruv needs to pick 2 horses for the race such that the difference in their skills is minimum. This way, he would be able to host a very interesting race. Your task is to help him do this and report the minimum difference that is possible between 2 horses in the race.

Input:

First line of the input file contains a single integer T, the number of test cases. Every test case starts with a line containing the integer N. The next line contains N space separated integers where the i^{th} integer is S[i].

Output:

For each test case, output a single line containing the minimum difference that is possible.

Constraints:

- $1 \leq T \leq 10$
- $2 \leq N \leq 5000$
- $1 \leq S[i] \leq 1000000000$

Sample

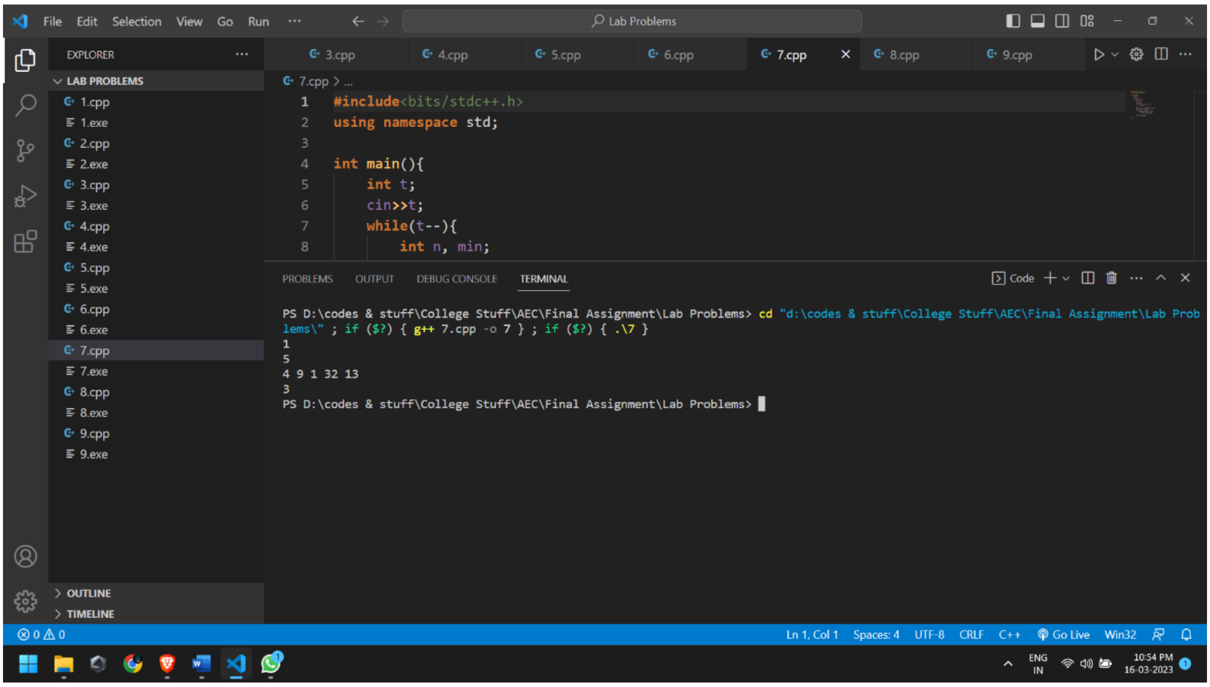
Input	Output

1
5
4 9 1 32 13
	3

Solution

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

int main(){
    int testcases;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                         //Entering the number of Test Cases
    while(testcases--){
        int n, min;
        cin>>n;                                 //Entering the number of
Horses in the Stable
        int s[n];
        for(int i=0; i<n; i++){
            cin>>s[i];                          //Entering the skills of
each horse
        }
        sort(s, s + n, greater<int>());         //Sorting the array in
decending order
        min = s[0] - s[1];
        for(int i=1; i<(n-1); i++){
            if((s[i]-s[i+1]) < min)           //Calculating the minimum
difference between each of the skills (array numbers)
                min = s[i]-s[i+1];
        }
        cout<<min<<endl;
    }
}
```



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

int main(){
    int t;
    cin>>t;
    while(t--){
        int n, min;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 7.cpp -o 7 } ; if ($?) { .\7 }
1
5
4 9 1 32 13
3
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 8

There is a frog initially placed at the origin of the coordinate plane. In exactly 1 second, the frog can either move up 1 unit, move right 1 unit, or stay still. In other words, from position (x, y), the frog can spend 1 second to move to:

- (x+1, y)
- (x, y+1)
- (x, y)

After T seconds, a villager who sees the frog reports that the frog lies on or inside a square of side-length s with coordinates (X, Y), (X+s, Y), (X, Y+s), (X+s, Y+s). Calculate how many points with integer coordinates on or inside this square could be the frog's position after exactly T seconds

Input Format:

The first and only line of input contains four space-separated integers: X, Y, s, and T.

Output Format:

Print number of points with integer coordinates that could be the frog's position after T seconds.

Constraints:

- 0≤X, Y≤100
- 1≤s≤100
- 0≤T≤400

Note that the Expected Output Feature for Custom Invocation is not supported for this contest.

Sample

Input	**Output**
2 2 3 6	6

The points shown are the points on or in the specified square, and those that are white are the points that the frog could be at after 6 seconds.

Solution

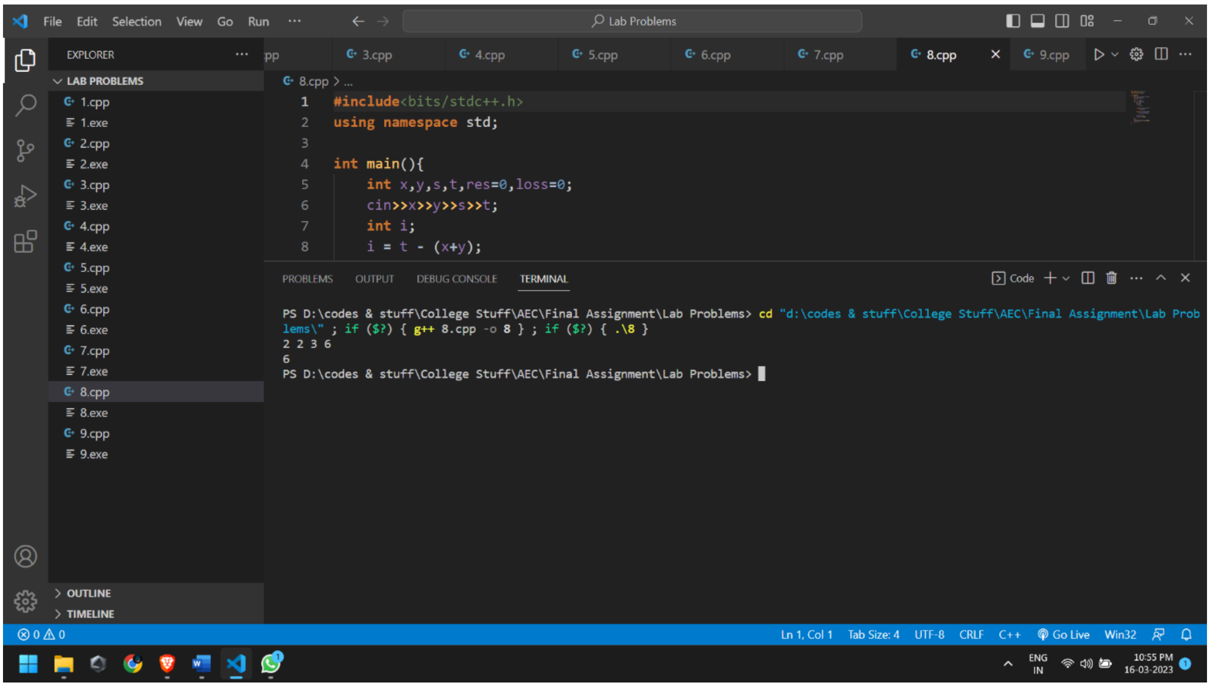
```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int x,y,s,t,res=0,loss=0;
    cout<<"Enter the Position of the frog (x, y), Side-length of the
square & Time: ";
    cin>>x>>y>>s>>t;
    int remaining;
    remaining = t - (x+y);
    if(remaining <= (2*s)+1){           //To check if the remaining
steps are within the square or not
        int j=0;
        if(remaining > s)
            j = remaining - s;          //If the remaining steps is
greater than the diagonal of the square then assign the number of
steps to be taken after the diagonal

        for(int k=remaining+1; k>=0; k--)
            res += k;                   //Loop to add the number of
points before the diagonal with the remaining steps

        for(int k=1; k<=j; k++)
            loss += (k * 2);            //If the steps go beyond the
diagonal, add the number of lost point with increment of lost points
of 2 with each step

        res = res - loss;
        cout<<res<<endl;
    }
    else
    cout<<((s+1)*(s+1))<<endl;          //If remaining steps go
outside the square then print maximum number of points present in
the square
}
```



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

int main(){
    int x,y,s,t,res=0,loss=0;
    cin>>x>>y>>s>>t;
    int i;
    i = t - (x+y);
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 8.cpp -o 8 } ; if ($?) { .\8 }
2 2 3 6
6
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> 
```

Problem 9

There are N students in a class, where the ith student has a score of Ai. The ith student will boast if and only if the number of students scoring less than or equal to Ai is greater than the number of students scoring greater than Ai. Find the number of students who will boast.

Input Format

- The first line contains T - the number of test cases. Then the test cases follow.
- The first line of each test case contains a single integer N - the number of students.
- The second line of each test case contains N integers 1,2,…,A1,A2,…,AN - the scores of the students.

Output Format

For each test case, output in a single line the number of students who will boast.

Constraints

- 1≤T≤1000
- 1≤N≤100
- 0≤Ai≤100

Sample

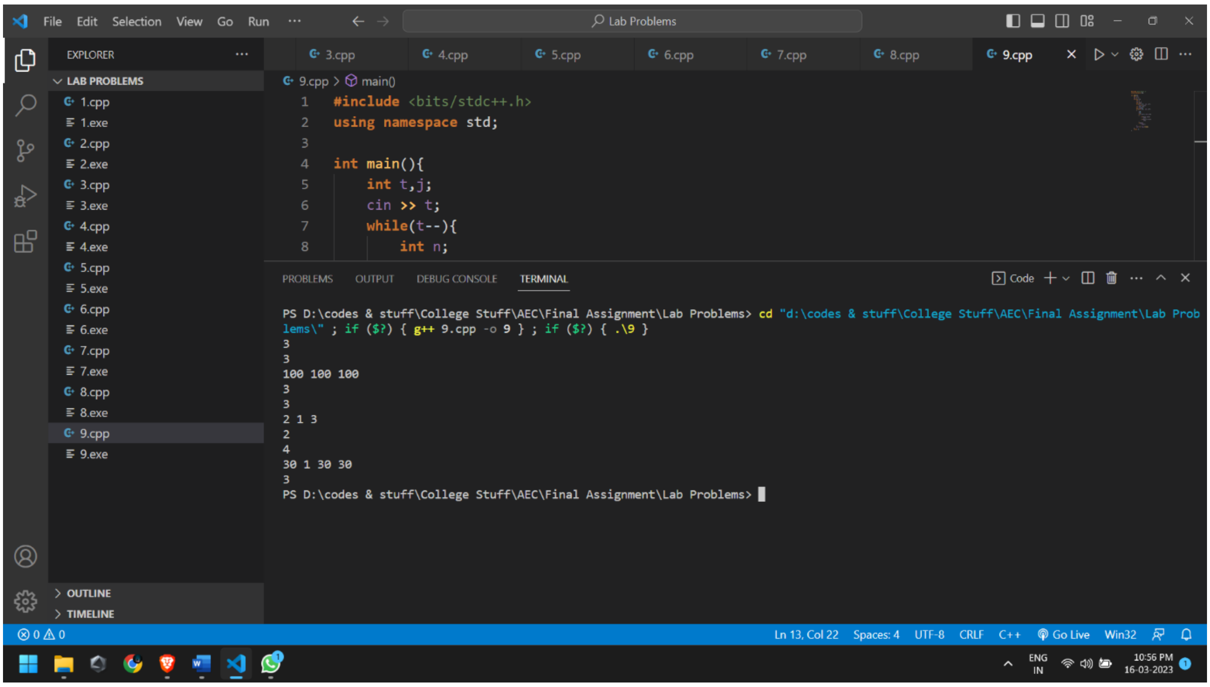
Input	Output

3
3
100 100 100
3
2 1 3
4
30 1 30 30
	3
	2
	3

Solution

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

int main(){
    int testcases,j;
    cout<<"Enter number of Test Cases: ";
    cin>>testcases;                               //Entering the number of Test Cases
    while(testcases--){
        int n;
        cin>>n;                                   //Entering the number of students in the class
        int a[n];
        for(int i=0 ; i<n ;i++)
            cin >> a[i];                          //Entering the marks of each student
        int lower=0 ,higher=0;
        int flag=0;
        for(int i=0 ; i<n ;i++)             //Outer Loop to give the
number of students boasting
        {
            lower=0;
            higher=0;
            for( j=0 ; j<n ;j++)            //Inner Loop to iterate
among each student
            {
                if(a[j]> a[i])
                    lower++;                //Counting the number of
students with lower marks
                if(a[j] <= a[i])
                    higher++;               //Counting the number of
students with higher marks
            }
            if(higher>lower)
                flag++;
        }
        cout << flag <<endl;
    }
    return 0;
}
```



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

int main(){
    int t,j;
    cin >> t;
    while(t--){
        int n;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Prob
lems\" ; if ($?) { g++ 9.cpp -o 9 } ; if ($?) { .\9 }
3
3
100 100 100
3
3
2 1 3
2
4
30 1 30 30
3
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Lab Problems>
```

PRACTICE PROBLEMS

Problem 10

Team RCB has earned X points in the games it has played so far in this year's IPL. To qualify for the playoffs, they must earn at least a total of Y points. They currently have Z games left, in each game they earn 2 points for a win, 1 point for a draw, and no points for a loss.

Is it possible for RCB to qualify for the playoffs this year?

Input Format

- First line will contain T, number of testcases. Then the testcases follow.
- Each testcase contains of a single line of input, three integers, X, Y, Z.

Output Format

- For each test case, output in one line YES if it is possible for RCB to qualify for the playoffs, or NO if it is not possible to do so.
- Output is case insensitive, which means that "yes", "Yes", "YEs", "no", "nO" - all such strings will be acceptable.

Constraints

- 1≤T≤5000
- 0≤X, Y, Z≤1000

Sample

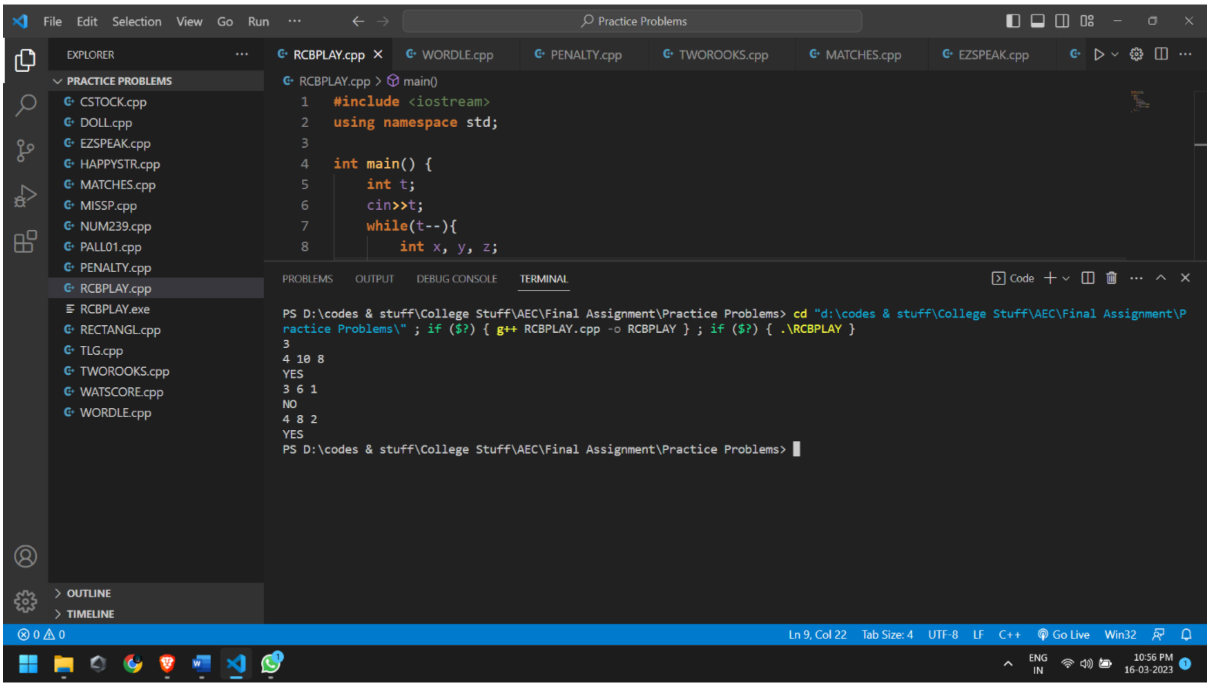
Input	Output

3
4 10 8
3 6 1
4 8 2
	YES
	NO
	YES

Solution

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int x, y, z;
        cin>>x>>y>>z;
        if(x+(z*2)>=y){
            cout<<"YES"<<endl;
        }else {
            cout<<"NO"<<endl;
        }
    }
    return 0;
}
```



```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int x, y, z;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems\" ; if ($?) { g++ RCBPLAY.cpp -o RCBPLAY } ; if ($?) { .\RCBPLAY }
3
4 10 8
YES
3 6 1
NO
4 8 2
YES
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 11

Chef invented a modified wordle. There is a hidden word S and a guess word T, both of length 5. Chef defines a string M to determine the correctness of the guess word. For the ith index:

- If the guess at the i^{th} index is correct, the ith character of M is G.
- If the guess at the ith index is wrong, the ith character of M is B.

Input Format

- First line will contain T, number of test cases. Then the test cases follow.
- Each test case contains of two lines of input.
- First line contains the string S - the hidden word.
- Second line contains the string T - the guess word.

Output Format

For each test case, print the value of string M.

You may print each character of the string in uppercase or lowercase (for example, the strings BgBgB, BGBGB, bgbGB and bgbgb will all be treated as identical).

Constraints

- 1≤T≤1000
- |S|=|T|=5
- S,T contain uppercase English alphabets only.

Sample

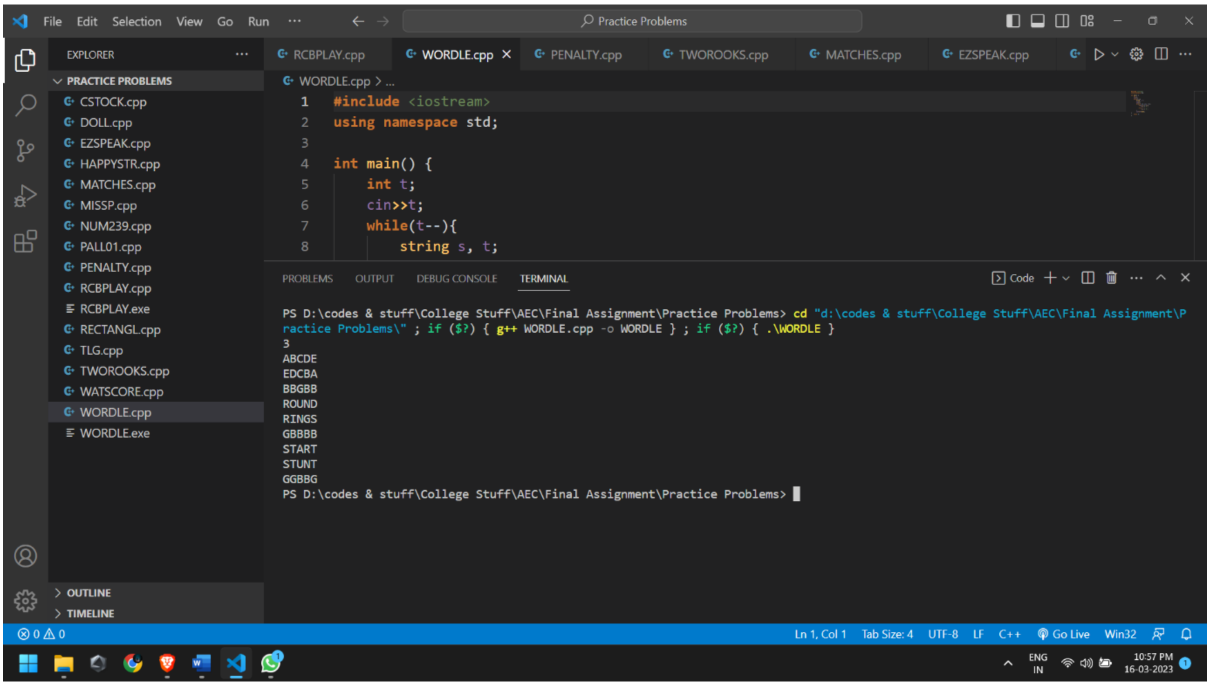
Input	Output
3 ABCDE EDCBA ROUND RINGS	BBGBB GBBBB GGBBG

START STUNT	

Solution

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        string s, t;
        cin>>s;
        cin>>t;
        for(int i=0; i<5; i++){
            if(s[i] == t[i])
                t[i] = 'G';
            else
                t[i] = 'B';
        }
        cout<<t<<endl;
}
    return 0;
}
```



```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        string s, t;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems\" ; if ($?) { g++ WORDLE.cpp -o WORDLE } ; if ($?) { .\WORDLE }
3
ABCDE
EDCBA
BBGBB
ROUND
RINGS
GBBBB
START
STUNT
GGBBG
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 12

It's the soccer match finals in Chef land and as always it has reached the penalty shootouts. Each team is given 5 shots to make and the team scoring a goal on the maximum number of shots wins the game. If both the teams' scores are equal, then the game is considered a draw and we would have 2 champions.

Given ten integers A1, A2…, A10, where the odd indexed integers (A1, A3, A5, A7, A9) represent the outcome of the shots made by team 11 and even indexed integers (A2,A4,A6,A8 ,A10) represent the outcome of the shots made by team 22 (here Ai=1 indicates that it's a goal and Ai=0 indicates a miss), determine the winner or find if the game ends in a draw.

Input Format

- The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains ten space-separated integers A1, A2, …, A10.

Output Format

For each test case, print a single line containing one integer - 00 if the game ends in a draw or 11 if the first team wins or 22 if the second team wins.

Constraints

- 1≤T≤1024
- 0≤Ai≤1

Sample

Input	Output

4
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1
 	
1 0 1 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0
	0
	2
	1
	1

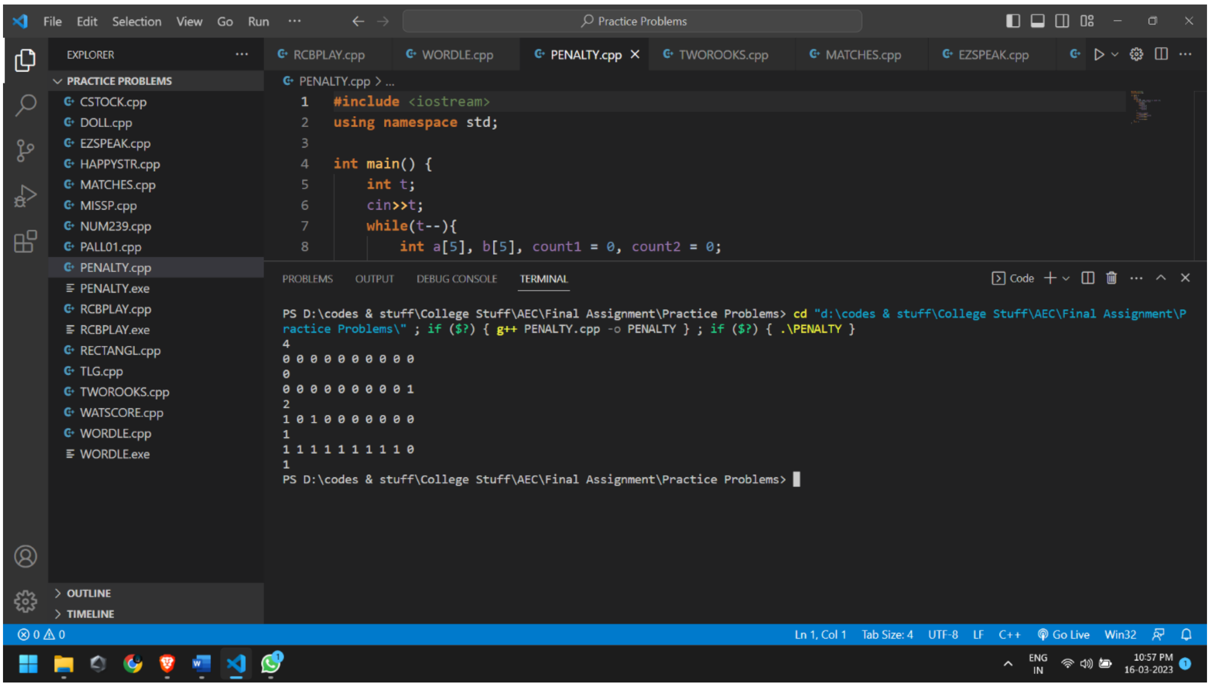
Solution

```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int a[5], b[5], count1 = 0, count2 = 0;
        for(int i=0; i<5; i++){
            cin>>a[i];
            cin>>b[i];
            if(a[i] == 1)
                count1++;
            if(b[i] == 1)
                count2++;
        }

        if(count1 > count2)
            cout<<1<<endl;
        else if(count1 < count2)
            cout<<2<<endl;
        else
```

```
            cout<<0<<endl;
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int a[5], b[5], count1 = 0, count2 = 0;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ PENALTY.cpp -o PENALTY } ; if ($?) { .\PENALTY }
4
0 0 0 0 0 0 0 0 0 0
0
0 0 0 0 0 0 0 0 0 1
2
1 0 1 0 0 0 0 0 0 0
1
1 1 1 1 1 1 1 1 1 0
1
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 13

You are given a standard 8×8 chessboard which has exactly 2 rooks on it and no other pieces. The rows are numbered 1 to 8 from bottom to top, and the columns are numbered 1 to 8 from left to right. The cell at the intersection of the *i*-th column and *j*-th row is denoted(*i*,*j*).

Given the initial positions of the rooks in the form of coordinates (*X*1,*Y*1) and (*X*2,*Y*2), you need to tell whether the 22 rooks currently attack each other or not. Assume, each square can contain at most one piece.

Rooks can only travel in straight lines along the row or column they are placed at, and can't jump over other pieces. For a more detailed explanation of the moves of rooks, along with images, please click here.

Input Format

- The first line contains T - the number of test cases. Then the test cases follow.
- The first line of each test case contain four space-separated integers each $X1$, $Y1$, $X2$, $Y2$ - $(X1,Y1)$ is the position of the first rook and $(X2,Y2)$ is the position of the second rook.

Output Format

For each test case, output on a single line YES (without quotes) if the rooks attack each other, and NO otherwise.
You may print each character of the string in uppercase or lowercase (for example, the strings YeS, YEs, yes and yeS will all be treated as identical).

Constraints

- $1 \leq \leq X1,X2,Y1,Y2 \leq 8$
- $(X1,Y1) \neq (X2,Y2)$

Sample

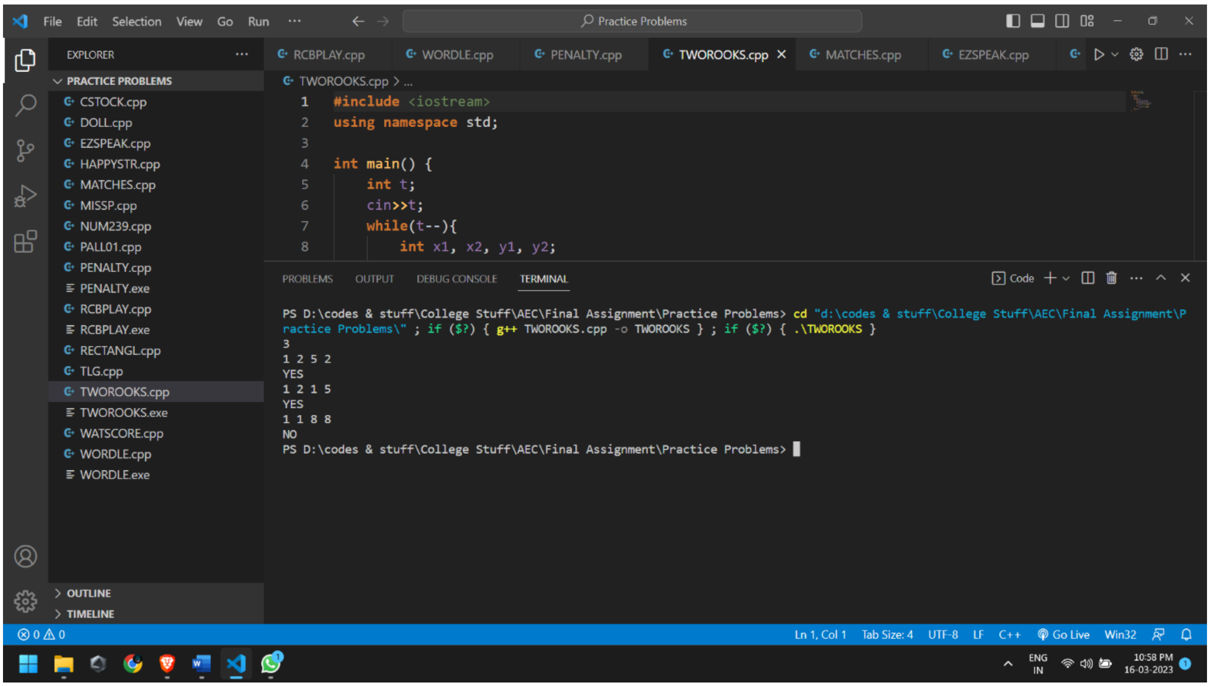
Input	Output

3
1 2 5 2
1 2 1 5
1 1 8 8
	YES
	YES
	NO

Solution

```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int x1, x2, y1, y2;
        cin>>x1>>y1>>x2>>y2;
        if(x1 == x2 || y1 == y2)
            cout<<"YES"<<endl;
        else
            cout<<"NO"<<endl;
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int x1, x2, y1, y2;
```

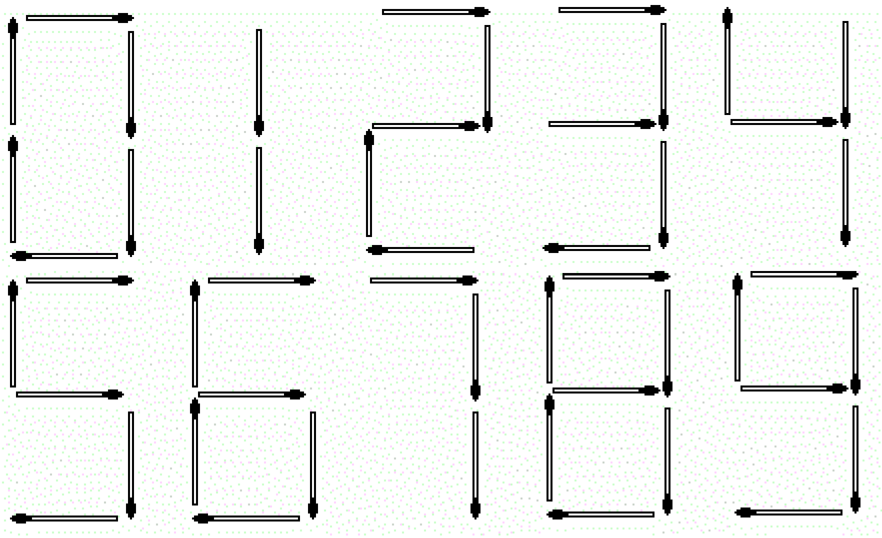
```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ TWOROOKS.cpp -o TWOROOKS } ; if ($?) { .\TWOROOKS }
3
1 2 5 2
YES
1 2 1 5
YES
1 1 8 8
NO
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 14

Chef's son Chefu found some matches in the kitchen and he immediately starting playing with them.

The first thing Chefu wanted to do was to calculate the result of his homework — the sum of *A* and *B*, and write it using matches. Help Chefu and tell him the number of matches needed to write the result.

Digits are formed using matches in the following way:



Input

The first line of the input contains a single integer *T* denoting the number of test cases. The description of *T* test cases follows.

The first and only line of each test case contains two space-separated integers *A* and *B*.

Output

For each test case, print a single line containing one integer — the number of matches needed to write the result (*A*+*B*).

Constraints

- $1 \le T \le 1{,}000$
- $1 \le \le A, B \le 10^6$

Sample

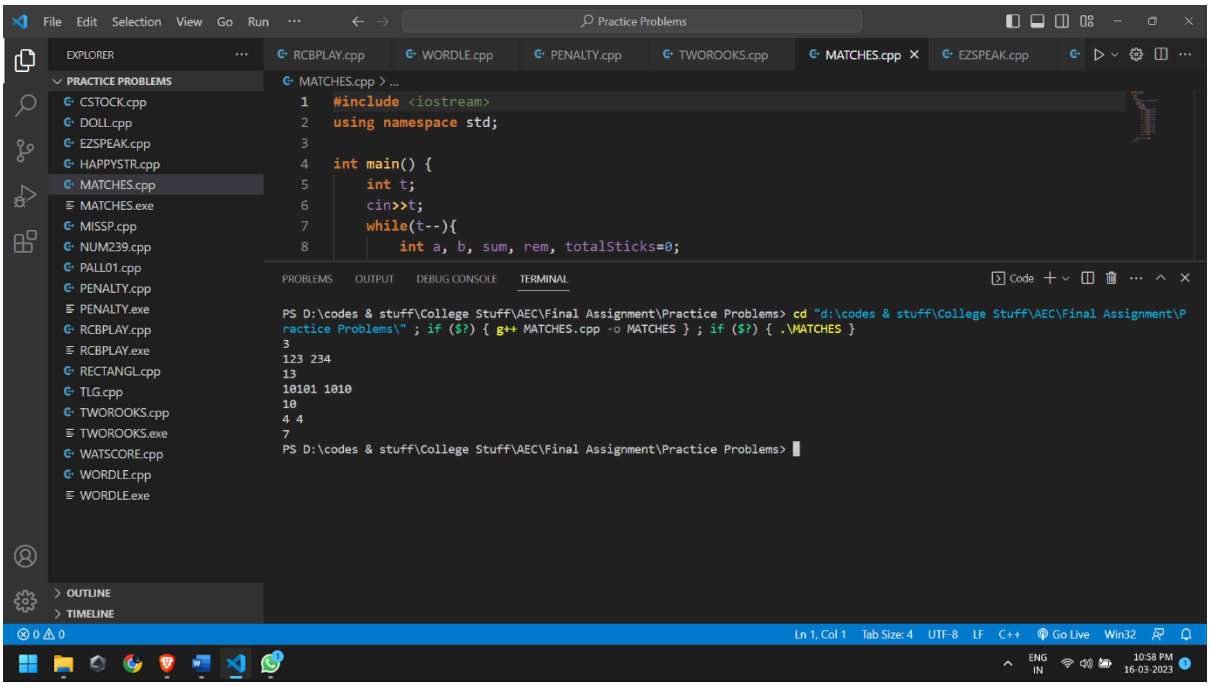
Input	Output
3	13
123 234	10
10101 1010	7

4 4	

Solution

```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int a, b, sum, rem, totalSticks=0;
        cin>>a>>b;
        sum=a+b;
        while(sum>0){
            rem=sum%10;
            sum=sum/10;
            switch(rem){
                case 0: totalSticks+=6;
                        break;
                case 1: totalSticks+=2;
                        break;
                case 2: totalSticks+=5;
                        break;
                case 3: totalSticks+=5;
                        break;
                case 4: totalSticks+=4;
                        break;
                case 5: totalSticks+=5;
                        break;
                case 6: totalSticks+=6;
                        break;
                case 7: totalSticks+=3;
                        break;
                case 8: totalSticks+=7;
                        break;
                case 9: totalSticks+=6;
                        break;
                default: break;
            }
        }
        cout<<totalSticks<<endl;
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int a, b, sum, rem, totalSticks=0;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ MATCHES.cpp -o MATCHES } ; if ($?) { .\MATCHES }
3
123 234
13
10101 1010
10
4 4
7
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 15

Words that contain many consecutive consonants, like "schtschurowskia", are generally considered somewhat hard to pronounce.

We say that a word is hard to pronounce if it contains 44 or more consonants in a row; otherwise, it is easy to pronounce. For example, "apple" and "polish" are easy to pronounce, but "schtschurowskia" is hard to pronounce.

You are given a string S consisting of N lowercase Latin characters. Determine whether it is easy to pronounce or not based on the rule above — print YES if it is easy to pronounce and NO otherwise.

For the purposes of this problem, the vowels are the characters {a, e, i, o, u} and the consonants are the other 2121 characters.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.

  - The first line of each test case contains a single integer N, the length of string S.
  - The second line of each test case contains the string S.

Output Format

For each test case, output on a new line the answer — YES if S is easy to pronounce, and NO otherwise.

Each character of the output may be printed in either uppercase or lowercase. For example, the strings YES, yeS, yes, and YeS will all be treated as identical.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- S contains only lowercase Latin characters, i.e., the characters {a, b, c,…,z}

Sample

Input	**Output**
5 5 apple 15 Schtschurowskia 6 Apple 5 Tryst 3 cry	YES NO YES NO YES

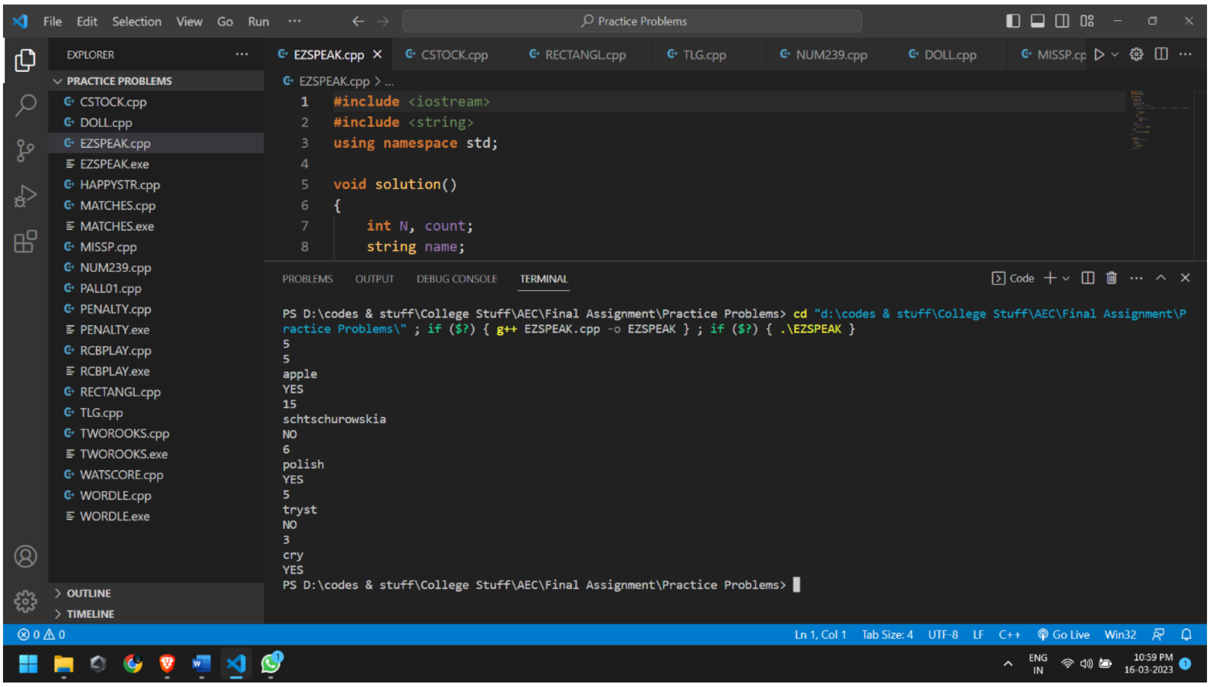
Solution

```
#include <iostream>
#include <string>
using namespace std;

void solution()
{
    int N, count;
    string name;

    cin >> N >> name;
    count = 0;
    for(int i = 0; i < N; i++)
    {
        if(name[i] == 'a' || name[i] == 'e' || name[i] == 'i' ||
name[i] == 'o' || name[i] == 'u')
        {
            count = 0;
            continue;
        }
        else
        {
            count++;
            if(count == 4)
            break;
        }
    }
    if(count < 4)
    {
        cout << "YES" << endl;
    }
    else
    {
        cout << "NO" << endl;
    }
}


int main() {
    int T;
    cin >> T;
    while(T--)
    {
        solution();
    }
    return 0;
}
```



```
#include <iostream>
#include <string>
using namespace std;

void solution()
{
    int N, count;
    string name;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ EZSPEAK.cpp -o EZSPEAK } ; if ($?) { .\EZSPEAK }
5
5
apple
YES
15
schtschurowskia
NO
6
polish
YES
5
tryst
NO
3
cry
YES
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 16

Chef wants to buy a stock whose price was S rupees when the market opened. He will buy the stock if and only if its price is in the range [A,B]. The price of the stock has changed by C% by the time he was trying to buy the stock. Will he be able to buy the stock?

Input Format

- First line of the input contains T, the number of testcases. Then the test cases follow.
- Each test case contains 44 space-separated integers S,A,B,C in a single line.

Output Format

For each test case, if Chef buys the stock print YES, otherwise print NO.

You may print each character of the string in uppercase or lowercase (for example, the strings yEs, yes, Yes, and YES will all be treated as identical).

Constraints

- 1≤≤T≤1000
- 0≤S≤106
- $0 \leq A \leq B \leq 10^6$
- −100≤C≤100

Sample

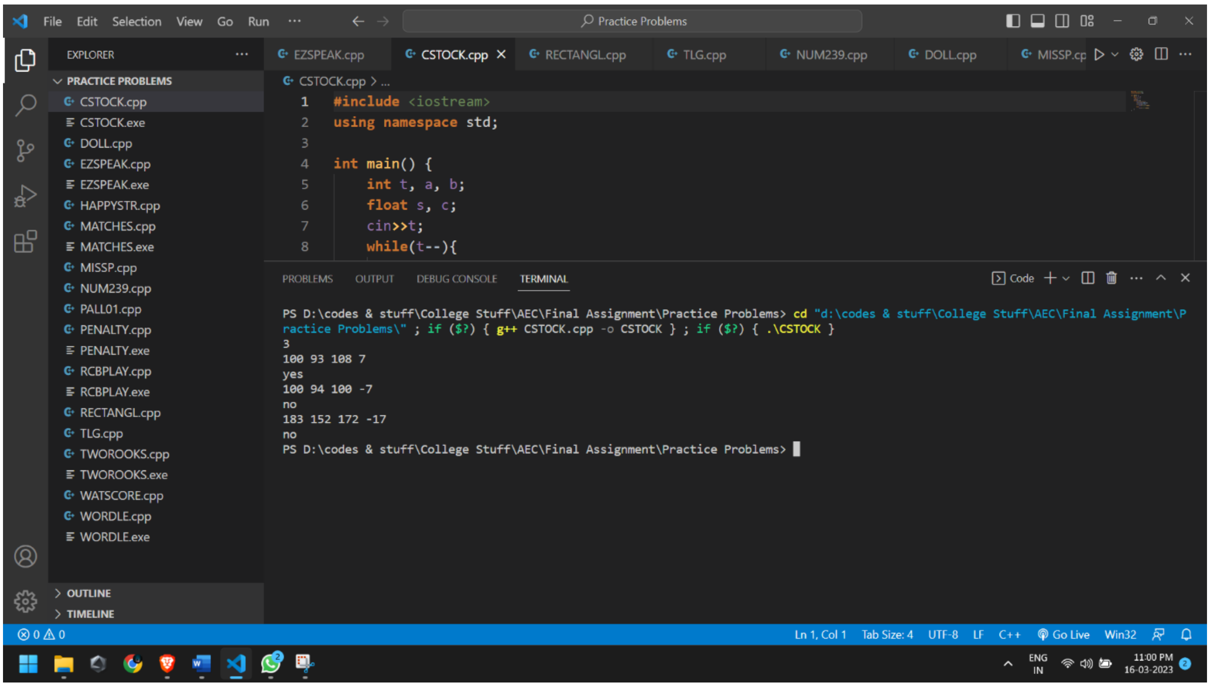
Input	Output

3
100 93 108 7
100 94 100 -7
183 152 172 -17
	Yes
	No
	No

Solution

```cpp
#include <iostream>
using namespace std;

int main() {
    int t, a, b;
    float s, c;
    cin>>t;
    while(t--){
        cin>>s>>a>>b>>c;
        s += (s*(c/100.0));
        if(s>=a && s<=b)
            cout<<"yes"<<endl;
        else
            cout<<"no"<<endl;
    }
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t, a, b;
    float s, c;
    cin>>t;
    while(t--){
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ CSTOCK.cpp -o CSTOCK } ; if ($?) { .\CSTOCK }
3
100 93 108 7
yes
100 94 100 -7
no
183 152 172 -17
no
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 17

You are given four integers a, b, c and d. Determine if there's a rectangle such that the lengths of its sides are a, b, c and d (in any order).

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains four space-separated integers a, b, c and d.

Output

For each test case, print a single line containing one string "YES" or "NO".

Constraints

- $1 \leq T \leq 1{,}000$
- $1 \leq a, b, c, d \leq 10{,}000$

Subtasks

Subtask #1 (100 points): original constraints

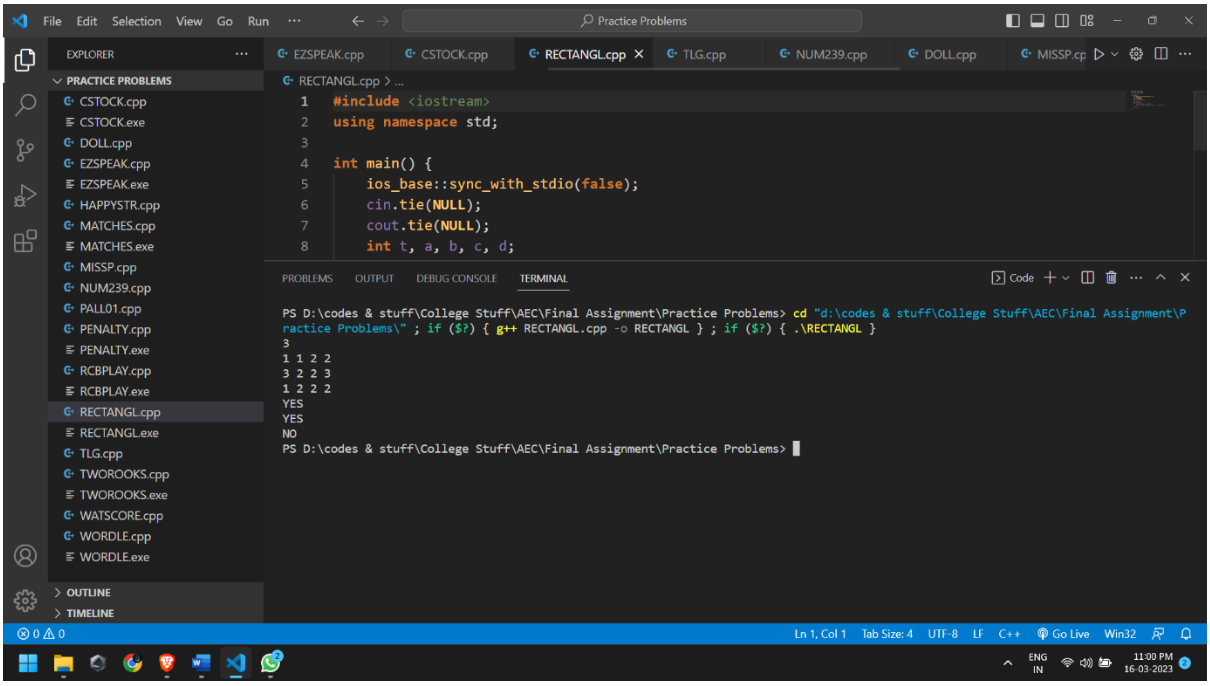
Sample

Input	Output
3	YES
1 1 2 2	YES
3 2 2 3	NO
1 2 2 2	

Solution

```cpp
#include <iostream>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t, a, b, c, d;
    cin >> t;
    while (t--) {
        cin >> a >> b >> c >> d;
        cout << ((a^b^c^d) == 0 ? "YES" : "NO") << '\n';
    }
    return 0;
}
```



```cpp
#include <iostream>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t, a, b, c, d;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ RECTANGL.cpp -o RECTANGL } ; if ($?) { .\RECTANGL }
3
1 1 2 2
3 2 2 3
1 2 2 2
YES
YES
NO
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 18

The game of billiards involves two players knocking 3 balls around on a green baize table. Well, there is more to it, but for our purposes this is sufficient.

The game consists of several rounds and in each round both players obtain a score, based on how well they played. Once all the rounds have been played, the total score of each player is determined by adding up the scores in all the rounds and the player with the higher total score is declared the winner.

The Siruseri Sports Club organizes an annual billiards game where the top two players of Siruseri play against each other. The Manager of Siruseri Sports Club decided to add his own twist to the game by changing the rules for determining the winner. In his version, at the end of each round, the **cumulative score** for each player is calculated, and the leader and her current lead are found. Once all the rounds are over the player who had the maximum lead at the end of any round in the game is declared the winner.

Consider the following score sheet for a game with 5 rounds:

Round	Player 1	Player 2
1	140	82
2	89	134

3	90	110
4	112	106
5	88	90

The total scores of both players, the leader and the lead after each round for this game is given below:

Round	Player 1	Player 2	Leader	Lead
1	140	82	Player 1	58
2	229	216	Player 1	13
3	319	326	Player 2	7
4	431	432	Player 2	1
5	519	522	Player 2	3

Note that the above table contains the cumulative scores.

The winner of this game is Player 1 as he had the maximum lead (58 at the end of round 1) during the game.

Your task is to help the Manager find the winner and the winning lead. You may assume that the scores will be such that there will always be a single winner. That is, there are no ties.

Input

The first line of the input will contain a single integer N (N ≤ 10000) indicating the number of rounds in the game. Lines 2,3,...,N+1 describe the scores of the two players in the N rounds. Line i+1 contains two integer Si and Ti, the scores of the Player 1 and 2 respectively, in round i. You may assume that 1 ≤ Si ≤ 1000 and 1 ≤ Ti ≤ 1000.

Output

Your output must consist of a single line containing two integers W and L, where W is 1 or 2 and indicates the winner and L is the maximum lead attained by the winner.

Sample

Input	Output

5
140 82
89 134
90 110
112 106
88 90
	1 58

Solution

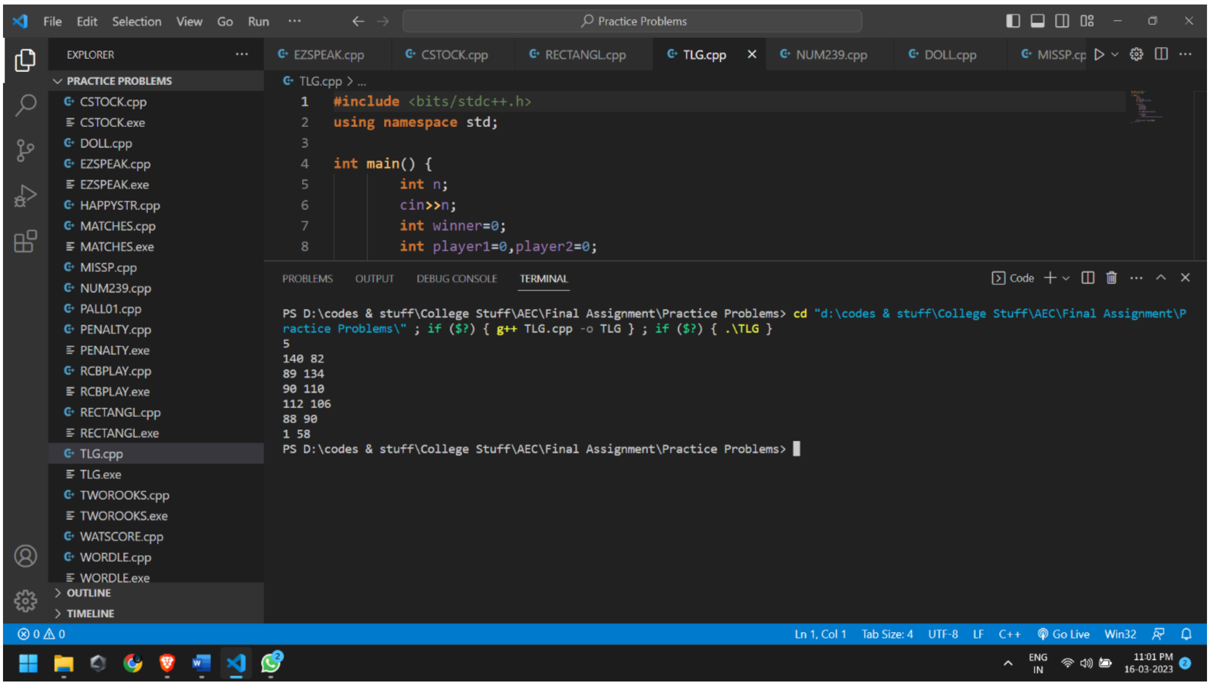
```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
        int n;
        cin>>n;
        int winner=0;
        int player1=0,player2=0;
        int lead=0;

        while(n--){
            int p1,p2;
            cin>>p1>>p2;
            player1+=p1;
            player2+=p2;

            int x=abs(player1-player2);

            if(x>lead){
                lead=x;
                player1>player2?winner=1:winner=2;
            }
        }
        cout<<winner<<" "<<lead<<endl;
    return 0;
}
```



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

int main() {
        int n;
        cin>>n;
        int winner=0;
        int player1=0,player2=0;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ TLG.cpp -o TLG } ; if ($?) { .\TLG }
5
140 82
89 134
90 110
112 106
88 90
1 58
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 19

Vasya likes the number 239. Therefore, he considers a number *pretty* if its last digit is 2, 3 or 9. Vasya wants to watch the numbers between L and R (both inclusive), so he asked you to determine how many pretty numbers are in this range. Can you help him?

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains two space-separated integers L and R.

Output

For each test case, print a single line containing one integer — the number of pretty numbers between L and R.

Constraints

- $1 \le T \le 100$
- $1 \le L \le R \le 105$

Subtasks

Subtask #1 (100 points): original constraints

Sample

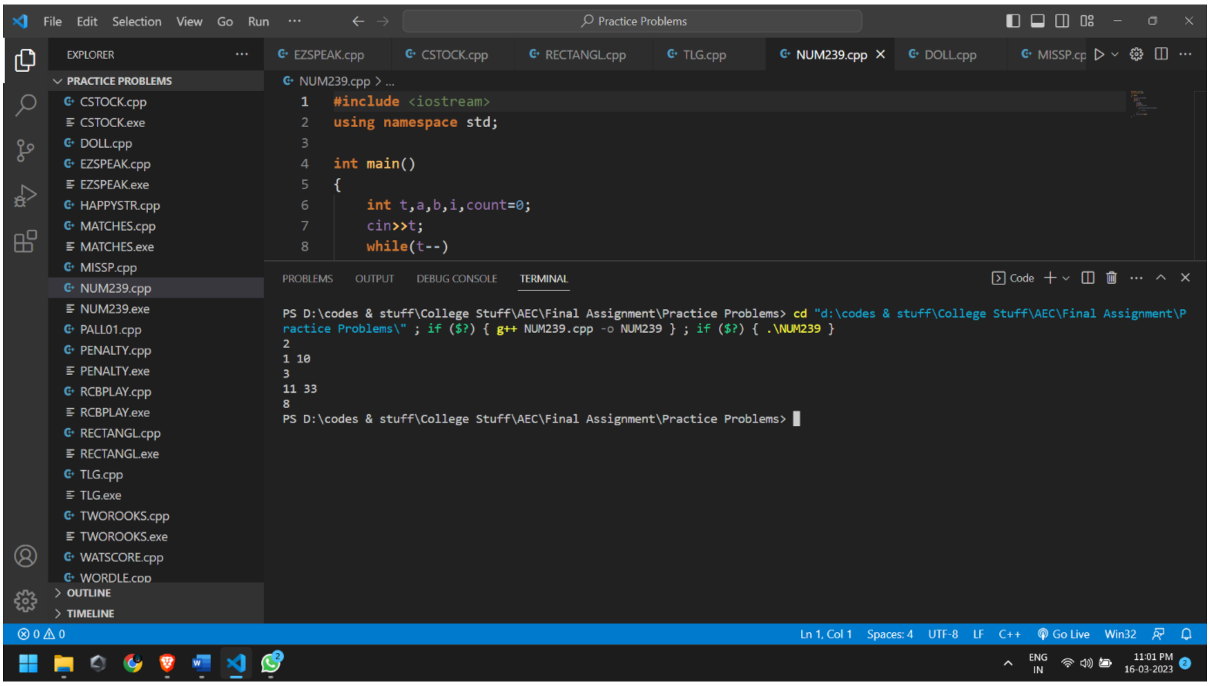
Input	Output

2
1 10
11 33
	3
	8

Solution

```cpp
#include <iostream>
using namespace std;

int main()
{
    int t,a,b,i,count=0;
    cin>>t;
    while(t--)
    {
        count=0;
        cin>>a>>b;
        for(i=a;i<=b;i++)
        {
            if(i%10==2||i%10==3||i%10==9)
            {
                count++;
            }
        }
        cout<<count<<endl;
    }
}
```



```cpp
#include <iostream>
using namespace std;

int main()
{
    int t,a,b,i,count=0;
    cin>>t;
    while(t--)
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ NUM239.cpp -o NUM239 } ; if ($?) { .\NUM239 }
2
1 10
3
11 33
8
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 20

You won't get caught if you hide behind someone."

Sang-Woo advises Gi-Hun to hide behind someone to avoid getting shot.

Gi-Hun follows Sang-Woo's advice and hides behind Ali, who saved his life earlier. Gi-Hun and Ali both have the same height, K. Many players saw this trick and also started hiding behind Ali.

Now, there are N players standing *between* Gi-Hun and Ali in a straight line, with the ith player having height Hi. Gi-Hun wants to know the minimum number of players who need to get shot so that Ali is visible in his line of sight.

Note:

- Line of sight is a straight line drawn between the topmost point of two objects. Ali is visible to Gi-Hun if nobody between them crosses this line.
- Even if there are some players who have the same height as that of Gi-Hun and Ali, Ali will be visible in Gi-Hun's line of sight.
- Gi-Hun and Ali have the same height.

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.

- The first line of each test case contains two space-separated integers N and K, denoting the total number of players between Gi-Hun and Ali and the height of both of them respectively.
- The second line of each test case contains N space-separated integers, denoting the heights of the players between Gi-Hun and Ali.

Output Format

For each test case, output in a single line the minimum number of players who need to get shot so that Ali is visible in Gi-Hun's line of sight.

Constraints

- $1 \leq T \leq 105$
- $1 \leq N \leq 105$
- $1 \leq K \leq 106$
- $1 \leq Hi \leq 106$ for every $1 \leq i \leq N$.

Sample

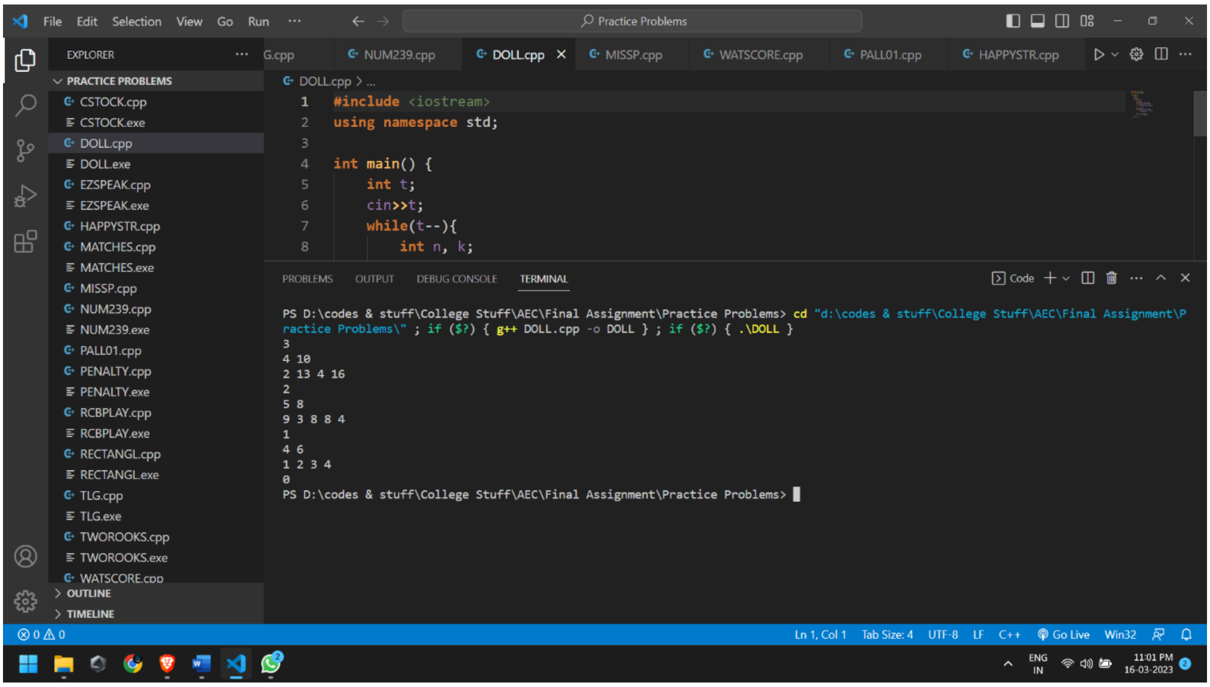
Input	Output

3
4 10
2 13 4 16
5 8
9 3 8 8 4
4 6
1 2 3 4
	2
	1
	0

Solution

```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int n, k;
        cin>>n>>k;
        int player_heights[n];
        for(int i=0; i<n; i++){
            cin>>player_heights[i];
        }
        int count=0;
        for(int i=0; i<n; i++){
            if(player_heights[i]>k)
                count++;
        }
        cout<<count<<endl;
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int n, k;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ DOLL.cpp -o DOLL } ; if ($?) { .\DOLL }
3
4 10
2 13 4 16
2
5 8
9 3 8 8 4
1
4 6
1 2 3 4
0
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```

Problem 21

Chef is fan of pairs and he likes all things that come in pairs. He even has a doll collection in which the dolls come in pairs. One day while going through his collection he found that there are odd number of dolls. Someone had stolen a doll!!!

Help chef find which type of doll is missing.

Input

- The first line contains an integer **T**, the number of test cases.

- The first line of each test case contains an integer **N**, the number of dolls. The next **N** lines are the types of dolls that are left.

Output

For each test case, display the type of doll that doesn't have a pair, in a new line.

Constraints

- 1<=T<=10
- 1<=N<=100000
- 0<=type<=100000

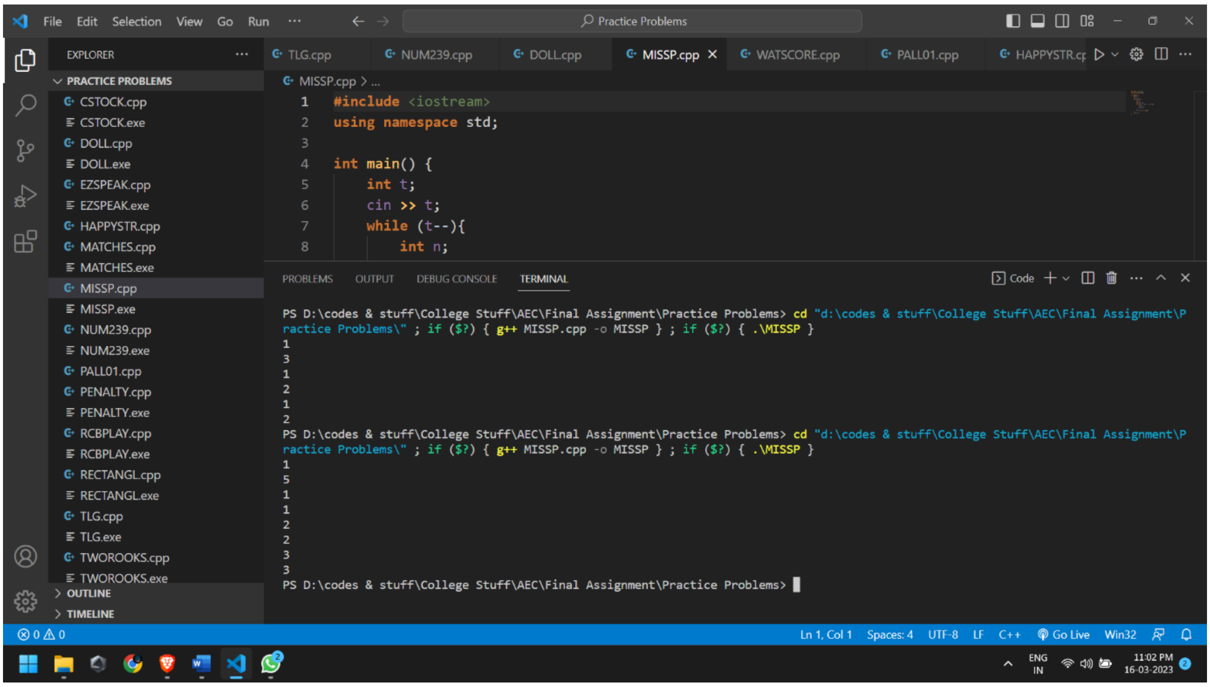
Sample

Input	**Output**
1 3 1 2 1	2

Solution

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin >> t;
    while (t--){
        int n;
        cin >> n;
        int ans = 0;
        for (int i = 0; i < n; i++){
            int t;
            cin >> t;
            ans ^= t;
        }
        cout << ans << endl;
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin >> t;
    while (t--){
        int n;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ MISSP.cpp -o MISSP } ; if ($?) { .\MISSP }
1
3
1
2
1
2
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ MISSP.cpp -o MISSP } ; if ($?) { .\MISSP }
1
5
1
1
2
2
3
3
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> █
```

Problem 22

You are participating in a contest which has 11 problems (numbered 1 through 11). The first eight problems (i.e. problems 1, 2,…,8) are *scorable*, while the last three problems (9, 10 and 11) are *non-scorable* — this means that any submissions you make on any of these problems do not affect your total score.

Your total score is the sum of your best scores for all scorable problems. That is, for each scorable problem, you look at the scores of all submissions you made on that problem and take the maximum of these scores (or 0 if you didn't make any submissions on that problem); the total score is the sum of the maximum scores you took.

You know the results of all submissions you made. Calculate your total score.

Input

- The first line of the input contains a single integer *T* denoting the number of test cases. The description of *T* test cases follows.

- The first line of each test case contains a single integer N denoting the number of submissions you made.
- N lines follow. For each i $(1 \le i \le N)$, the ith of these lines contains two space-separated integers pi and si, denoting that your ith submission was on problem pi and it received a score si.

Output

For each test case, print a single line containing one integer — your total score.

Constraints

- $1 \le \le T \le 10$
- $1 \le N \le 1{,}000$
- $1 \le pi \le 11$ for each valid i
- $0 \le si \le 100$ for each valid i

Sample

Input	Output

2
5
2 45
9 100
8 0
2 15
8 90
1
11 1
	135
	0

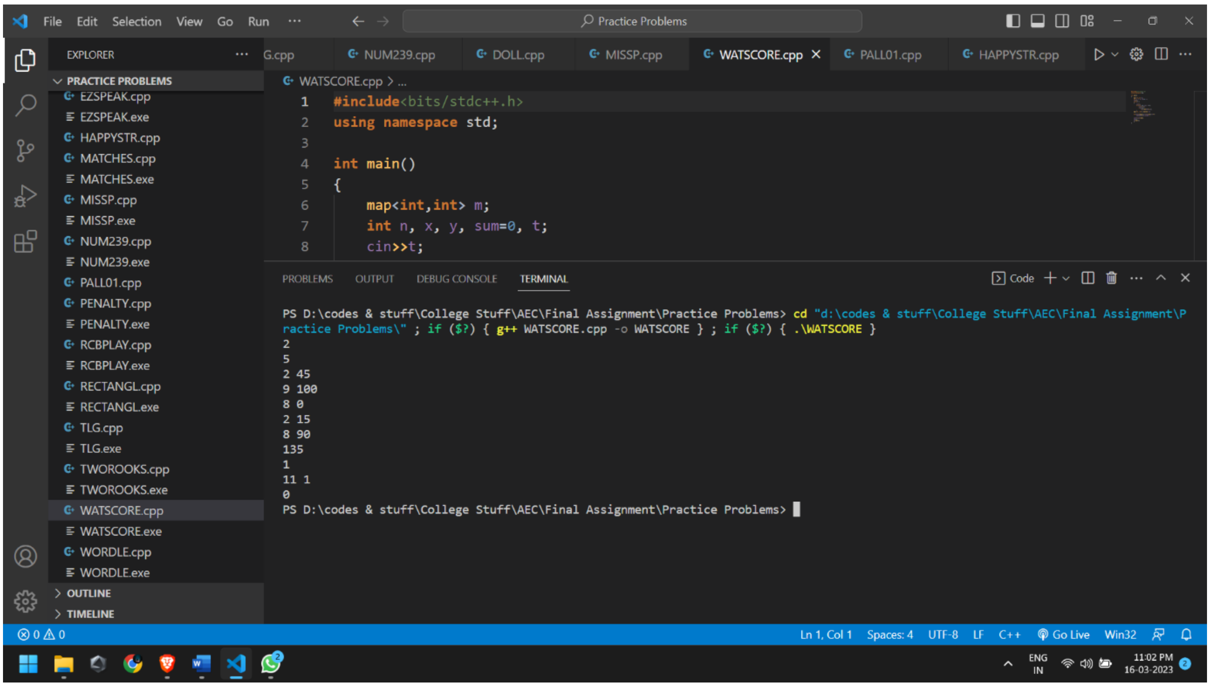
Solution

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

int main()
{
    map<int,int> m;
    int n, x, y, sum=0, t;
    cin>>t;
    while(t--)
    {
        cin>>n;
        for(int i=0; i<n; i++){
            cin>>x>>y;
            if(x<=8)
                m[x]=max(m[x],y);
        }
    map<int, int>::iterator it;

    for(it=m.begin();it!=m.end();it++)
        sum=sum+(*it).second ;

    cout<<sum<<endl;
    sum=0;
    m.clear();
    }
}
```



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

int main()
{
    map<int,int> m;
    int n, x, y, sum=0, t;
    cin>>t;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ WATSCORE.cpp -o WATSCORE } ; if ($?) { .\WATSCORE }
2
5
2 45
9 100
8 0
2 15
8 90
135
1
11 1
0
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 23

The citizens of Byte land regularly play a game. They have blocks each denoting some integer from 0 to 9. These are arranged together in a random manner without seeing to form different numbers keeping in mind that the first block is never a 0. Once they form a number they read in the reverse order to check if the number and its reverse is the same. If both are same then the player wins. We call such numbers *palindrome*.

Ash happens to see this game and wants to simulate the same in the computer. As the first step he wants to take an input from the user and check if the number is a palindrome and declare if the user wins or not.

Input

The first line of the input contains T, the number of test cases. This is followed by T lines containing an integer N.

Output

For each input output "wins" if the number is a palindrome and "loses" if not, in a new line.

Constraints

- 1<=T<=20
- 1<=N<=20000

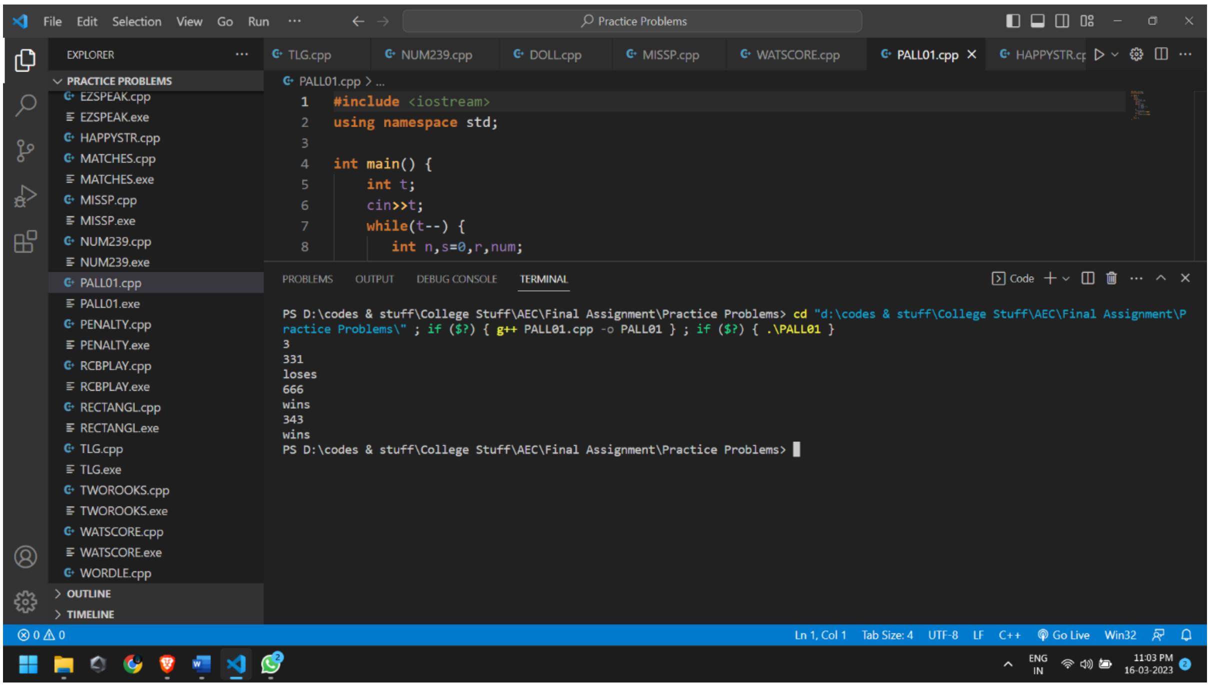
Sample

Input	**Output**
3 331 666 343	loses wins wins

Solution

```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--) {
       int n,s=0,r,num;
       cin>>n;
       num = n;
       while(n > 0) {
           r = n%10;
           s = (s*10) + r;
           n = n/10;
       }
       if(num == s) {
           cout<<"wins"<<endl;
       }else {
           cout<<"loses"<<endl;
       }
    }
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--) {
        int n,s=0,r,num;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ PALL01.cpp -o PALL01 } ; if ($?) { .\PALL01 }
3
331
loses
666
wins
343
wins
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> 
```

Problem 24

Chef has a string S with him. Chef is happy if the string contains a **contiguous substring** of length **strictly greater** than 22 in which all its characters are vowels.

Determine whether Chef is happy or not.

Note that, in English alphabet, vowels are a, e, i, o, and u.

Input

- First line will contain T, number of test cases. Then the test cases follow.
- Each test case contains of a single line of input, a string S.

Output

For each test case, if Chef is happy, print HAPPY else print SAD.

You may print each character of the string in uppercase or lowercase (for example, the strings hAppY, Happy, haPpY, and HAPPY will all be treated as identical).

Constraints

- $1 \leq T \leq 1000$
- $3 \leq |S| \leq 1000$, where $|S|$ is the length of S.
- S will only contain lowercase English letters.

Sample

Input	**Output**
4 aeiou abxy aebcdefghij	 Happy Sad Sad

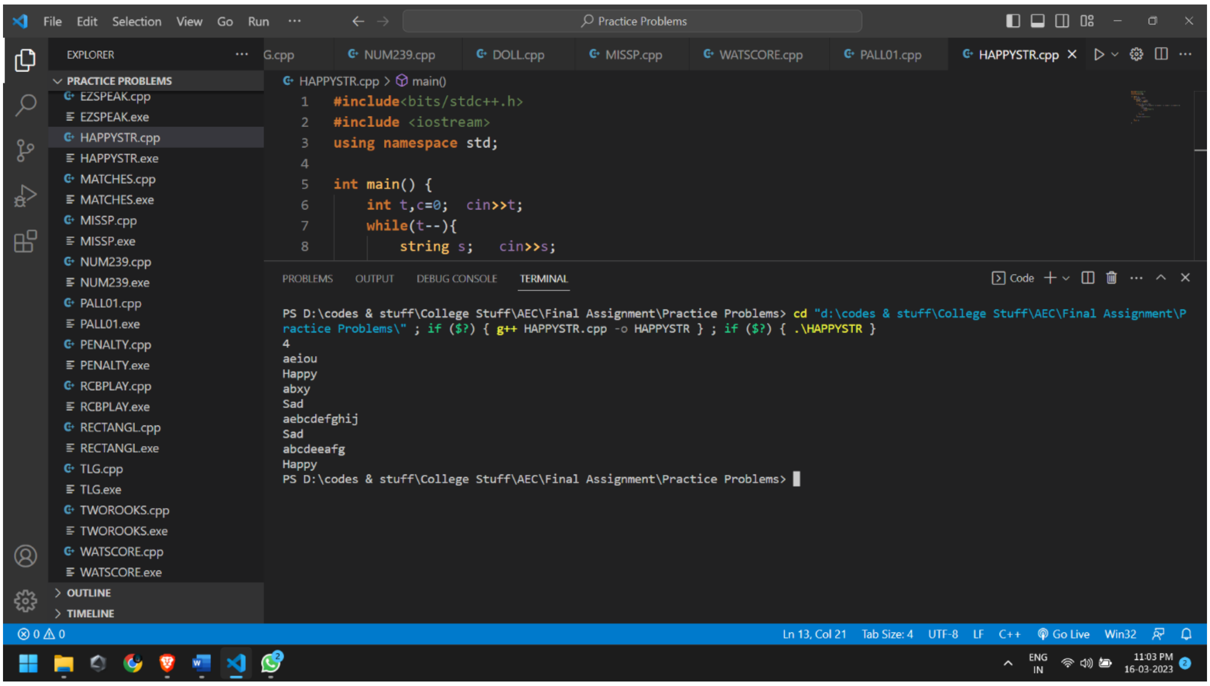
abcdeeafg	Happy

Solution

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

bool isVowel(char c){
    if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
        return true;
    return false;
}

int main(){
    int t;
    cin>>t;
    while(t--){
        int count=0;
        string s;
        cin>>s;
        for(int i=0; i<s.length(); i++){
            count = 0;
            if(isVowel(s[i])){
                for(int j=i; j<i+3; j++){
                    if(isVowel(s[j]))
                        count++;
                }
                if(count > 2){
                    cout<<"Happy"<<endl;
                    break;
                }
            }
        }
        if(count == 0)
            cout<<"Sad"<<endl;
    }
}
```



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

int main() {
    int t,c=0;  cin>>t;
    while(t--){
        string s;   cin>>s;
```

```
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems> cd "d:\codes & stuff\College Stuff\AEC\Final Assignment\P
ractice Problems\" ; if ($?) { g++ HAPPYSTR.cpp -o HAPPYSTR } ; if ($?) { .\HAPPYSTR }
4
aeiou
Happy
abxy
Sad
aebcdefghij
Sad
abcdeeafg
Happy
PS D:\codes & stuff\College Stuff\AEC\Final Assignment\Practice Problems>
```