Icover Test File

Problem-1

A palindromic number or numeral palindrome is a 'symmetrical' number like 19891 that remains the same when its digits are reversed. In this problem you will be given an integer, you have to say whether the number is a palindromic number or not.

Input

Input starts with an integer T (≤ 20000), denoting the number of test cases.

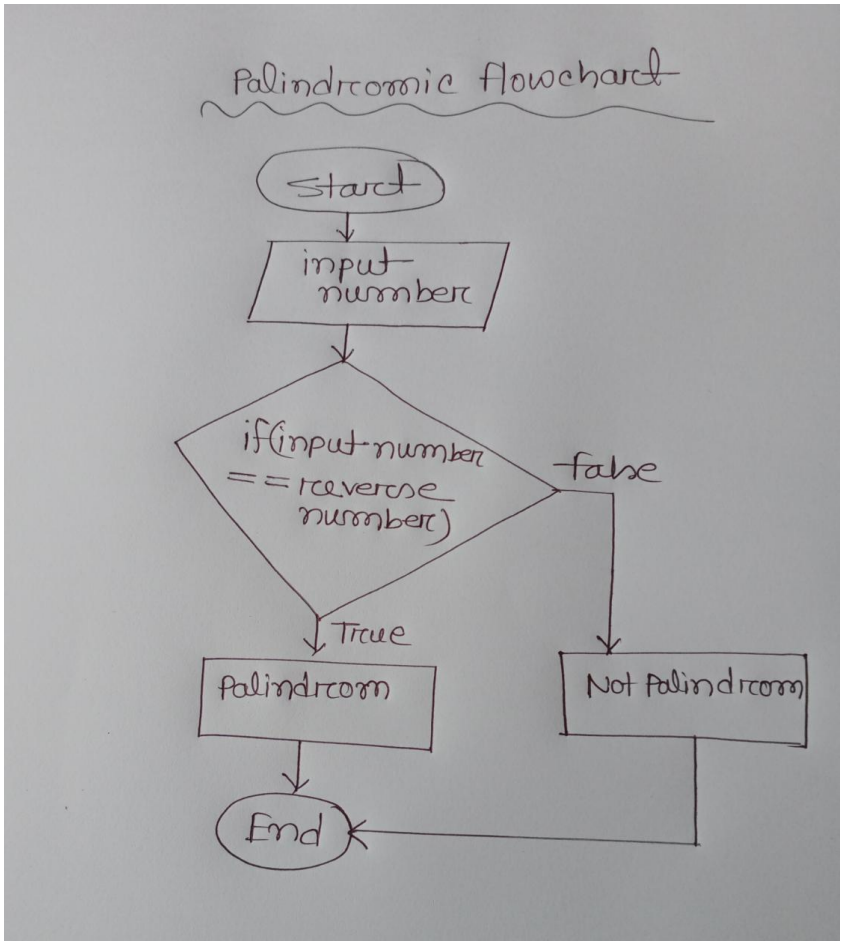
Each case starts with a line containing an integer n ($0 \leq n < 10^9$).

Output

For each case, print the case number and `Yes` if n is palindromic, otherwise print `No`.

Solution

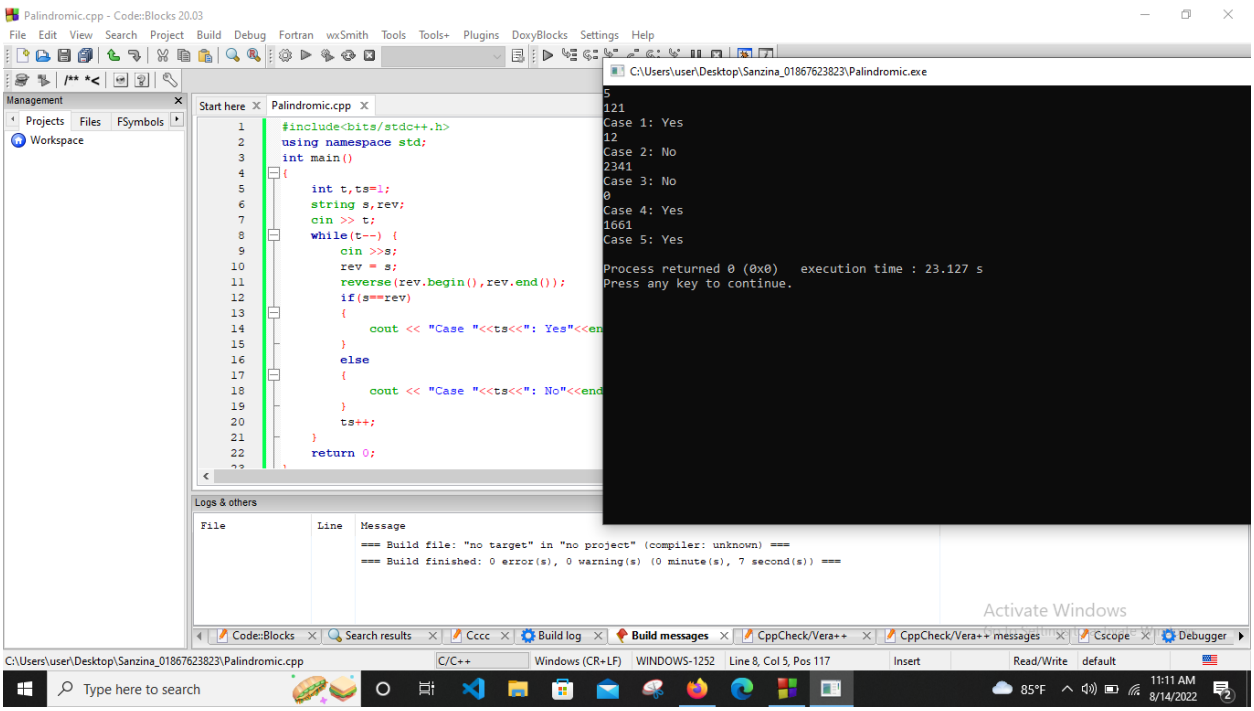
Flowchart:



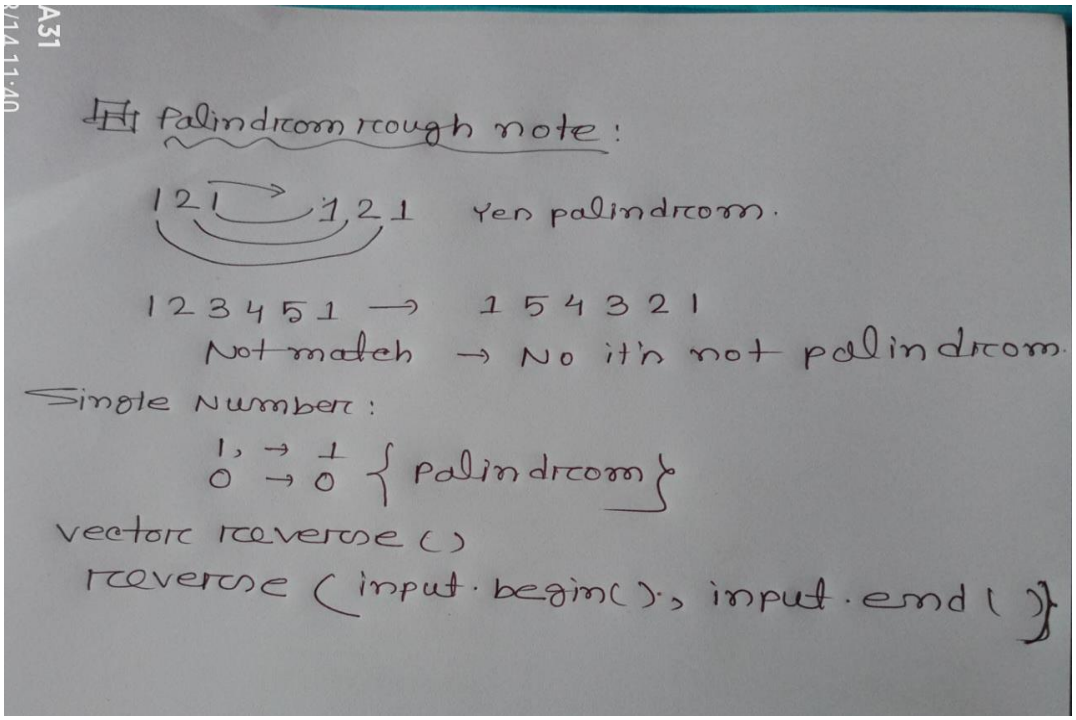
Code:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int t,ts=1;
   string s,rev;
```

```
    cin >> t;
    while(t--) {
        cin >>s;
        rev = s;
        reverse(rev.begin(),rev.end());
        if(s==rev)
        {
            cout << "Case "<<ts<<": Yes"<<endl;
        }
        else
        {
            cout << "Case "<<ts<<": No"<<endl;
        }
        ts++;
    }
    return 0;
}
```


Palindromic.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSymbols
Workspace
Start here
Palindromic.cpp
#include<bits/stdc++.h>
using namespace std;
int main()
{
int t,ts=1;
string s,rev;
cin >> t;
while(t--) {
cin >>s;
rev = s;
reverse(rev.begin(),rev.end());
if(s==rev)
{
cout << "Case "<<ts<<": Yes"<<en
}
else
{
cout << "Case "<<ts<<": No"<<end
}
ts++;
}
return 0;
}
C:\Users\user\Desktop\Sanzina_01867623823\Palindromic.exe
5
121
Case 1: Yes
12
Case 2: No
2341
Case 3: No
0
Case 4: Yes
1661
Case 5: Yes
Process returned 0 (0x0) execution time : 23.127 s
Press any key to continue.
Logs & others
File Line Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 7 second(s)) ===
Activate Windows
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger
C:\Users\user\Desktop\Sanzina_01867623823\Palindromic.cpp
C/C++ Windows (CR+LF) WINDOWS-1252 Line 8, Col 5, Pos 117 Insert Read/Write default
Type here to search
85°F
11:11 AM
8/14/2022

Rough Note:


Palindrom rough note:
121 → 121 Yes palindrom.
12345 1 → 1 5 4 3 2 1
Not match → No it's not palindrom.
Single Number:
1, → 1
0 → 0 { palindrom }
vector reverse()
reverse (input.begin(), input.end())

Problem-2

There is sequence 1, 12, 123, 1234, ..., 12345678910, ... Now you are given two integers A and B, you have to find the number of integers from Ath number to Bth (inclusive) number, which are divisible by 3.

For example, let A = 3. B = 5. So, the numbers in the sequence are, 123, 1234, 12345. And 123, 12345 are divisible by 3. So, the result is 2.

Input

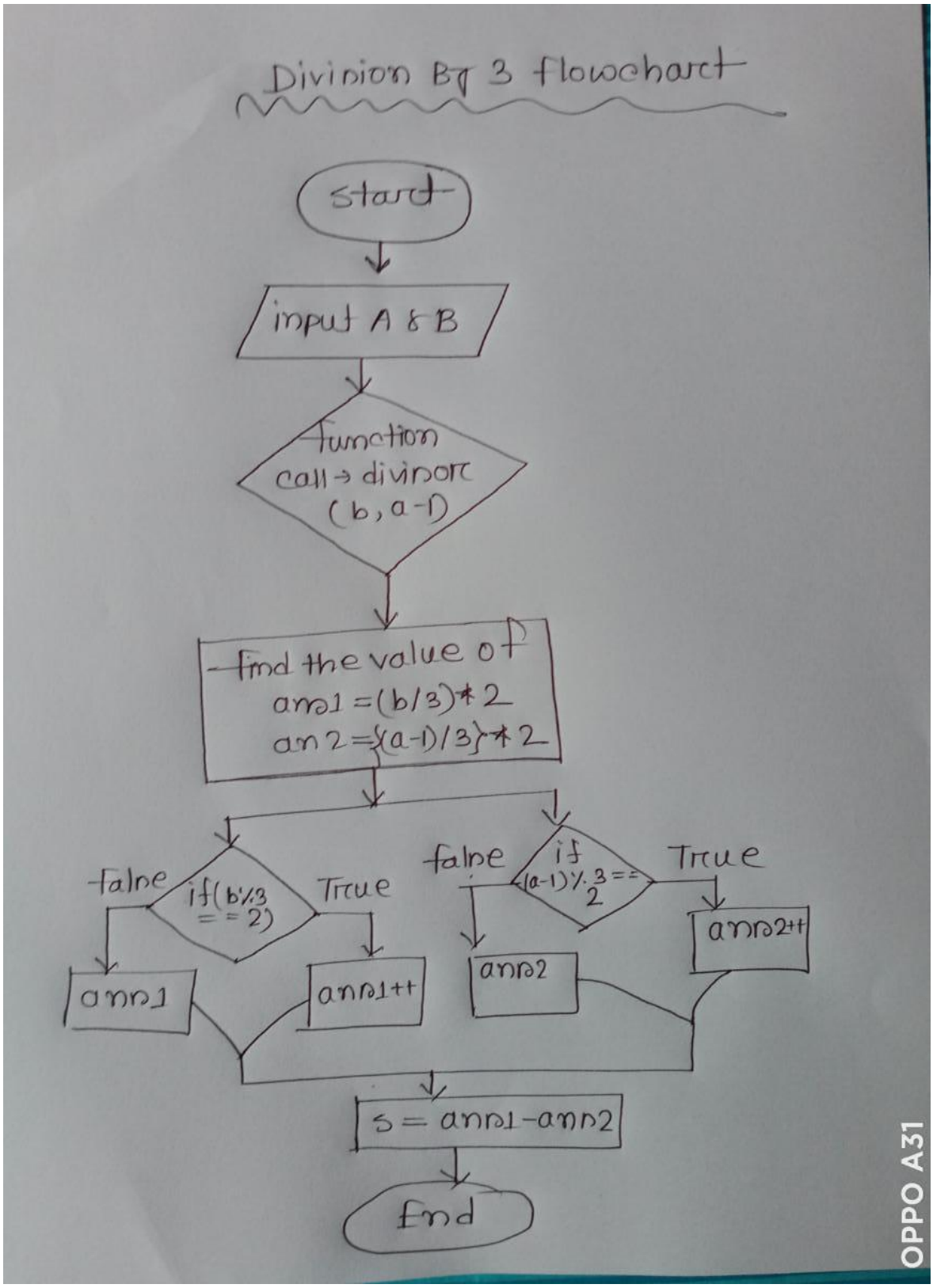
Input starts with an integer T (≤ 10000), denoting the number of test cases. Each case contains two integers A and B ($1 \leq A \leq B < 231$) in a line.

Output

For each case, print the case number and the total numbers in the sequence between Ath and Bth which are divisible by 3.

Solution

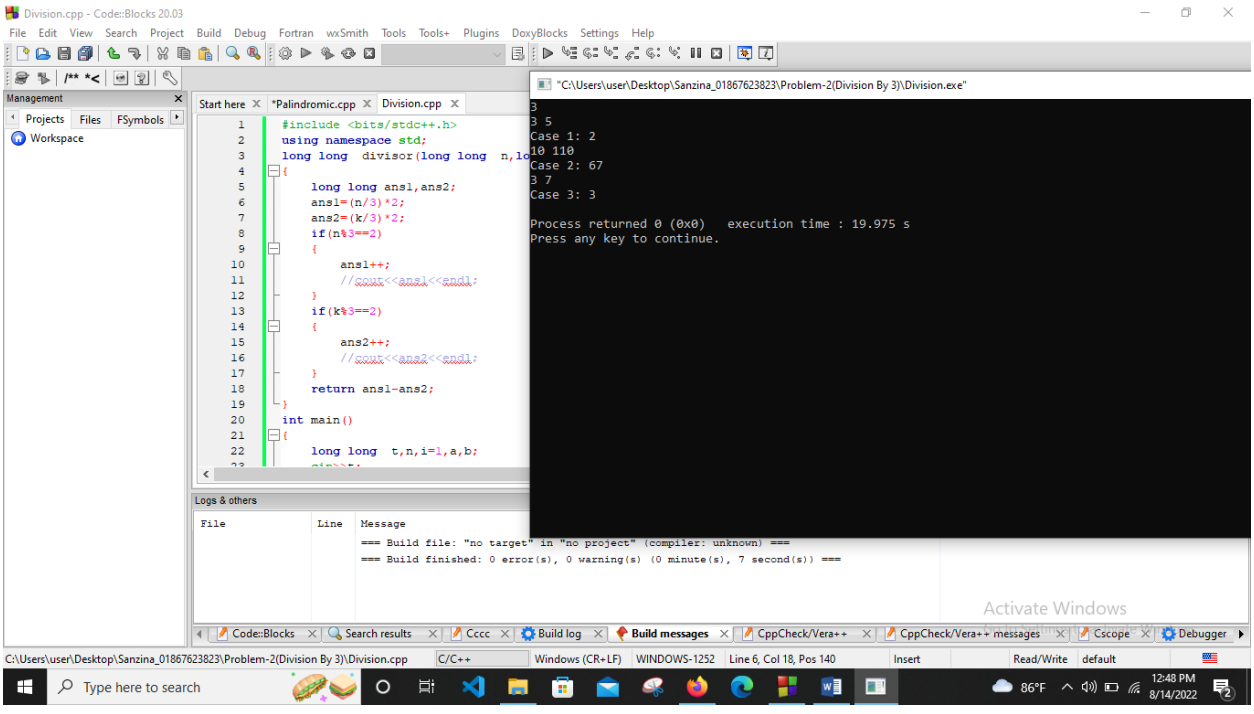
Flowchart:


Division By 3 flowchart
start
input A & B
function call → divisor (b, a-1)
find the value of
ans1 = (b/3)*2
ans2 = {(a-1)/3}*2
false
if (b%3 == 2)
True
ans1
ans1++
false
if (a-1)%3 == 2
True
ans2++
ans2
s = ans1-ans2
end
OPPO A31

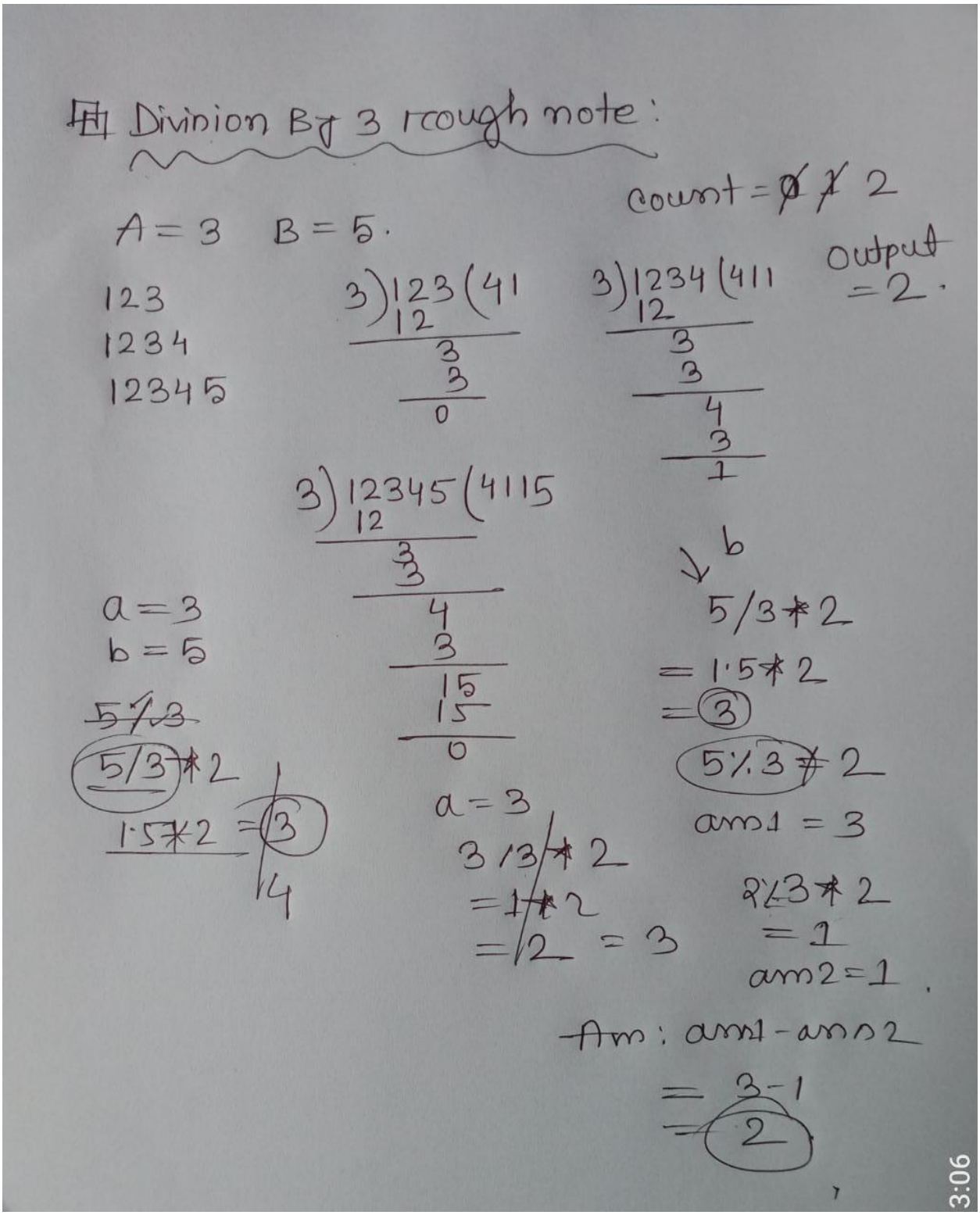
Code:

```
#include <bits/stdc++.h>
using namespace std;
long long  divisor(long long  n,long long k)
{
   long long ans1,ans2;
   ans1=(n/3)*2;
   ans2=(k/3)*2;
   if(n%3==2)
   {
      ans1++;
      //cout<<ans1<<endl;
   }
   if(k%3==2)
   {
      ans2++;
      //cout<<ans2<<endl;
   }
   return ans1-ans2;
}
int main()
{
```

```
    long long  t,n,i=1,a,b;
    cin>>t;
    while(t--)
    {
       cin>>a>>b;
       int s=divisor(b,a-1);
       //cout<<s<<endl;
       cout << "Case "<<i++<<": "<<s<<endl;
    }
    return 0;
}
```



Rough Note:


Division By 3 rough note:
count = 0 * 2
A = 3 B = 5.
123
1234
12345
3)123(41
12
3
3
0
3)1234(411
12
3
3
4
3
1
output = 2.
3)12345(4115
12
3
3
4
3
15
15
0
a = 3
b = 5
5/3
5/3*2
1.5*2 = 3
4
a = 3
3/3*2
= 1*2
= 2 = 3
b
5/3*2
= 1.5*2
= 3
5%3*2
am1 = 3
2%3*2
= 1
am2 = 1
Am: am1 - am2
= 3 - 1
= 2
3:06