

## Talent Battle 100 Days Coding Series

You have a grid with  $N$  rows and  $M$  columns. You have two types of tiles — one of dimensions  $2 \times 2$  and the other of dimensions  $1 \times 1$ . You want to cover the grid using these two types of tiles in such a way that:

- Each cell of the grid is covered by exactly one tile; and
- The number of  $1 \times 1$  tiles used is minimized.

Find the **minimum** number of  $1 \times 1$  tiles you have to use to fill the grid.

### 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 containing two space-separated integers  $N, M$ .

### Output Format

For each test case, print on a new line the minimum number of  $1 \times 1$  tiles needed to fill the grid.

### Sample Input

4  
1 1  
4 5  
6 8  
3 2

### Sample Output

1  
4  
0  
2

### C Solution

```
#include <stdio.h>

int main(void)
{
```

## Talent Battle 100 Days Coding Series

```
int t,a,b;
scanf("%d",&t);
while(t--)
{
    scanf("%d%d",&a,&b);
    if(a==1&&b==1)
    {
        printf("1\n");
    }
    else if((a%2==0)&&(b%2==0))
    {
        printf("0\n");
    }
    else if((a%2==0)&&(b%2!=0))
    {
        printf("%d\n",a);
    }
    else if((a%2!=0)&&(b%2==0))
    {
        printf("%d\n",b);
    }
    else
    {
        printf("%d\n",a+b-1);
    }
}
return 0;
}
```

TalentBattle

### C++ Solution

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

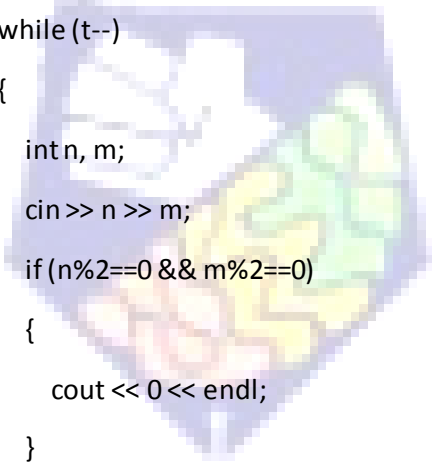
using namespace std;

#define ll long long

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);

    ll t;
    cin >> t;
    while (t--)
    {
        int n, m;
        cin >> n >> m;
        if (n%2==0 && m%2==0)
        {
            cout << 0 << endl;
        }
        else if (n%2==0 && m%2!=0)
        {
            cout << n << endl;
        }
        else if (n%2!=0 && m%2==0)
        {
            cout << m << endl;
        }
        else
        {

```



TalentBattle

## Talent Battle 100 Days Coding Series

```
        cout << n+m-1 << endl;
    }
}
return 0;
}
```

### Java

```
import java.util.*;
import java.lang.*;
import java.io.*;
class Main
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner sc=new Scanner (System.in);
        int t=sc.nextInt();
        while (t-->0){
            int n=sc.nextInt();
            int m=sc.nextInt();

            if(n%2==0&& m%2==0){
                System.out.println(0);
            }
            else if(n%2==0&& m%2!=0){
                System.out.println(n);
            }
            else if(n%2!=0&& m%2==0){
```

## Talent Battle 100 Days Coding Series

```
        System.out.println(m);
    }
    else{
        System.out.println(m+n-1);
    }
}
}
```

### Python

```
t=int(input())
while t!=0:
    m,n=map(int,input().split())
    if m%2==0 and n%2==0:
        print('0')
    elif m%2==0 and n%2==1:
        print(m)
    elif m%2==1 and n%2==0:
        print(n)
    else:
        print(m+n-1)
    t-=1
```

TalentBattle