时间限制：1秒

空间限制：32768K

小Q得到一个神奇的数列: 1, 12, 123,...12345678910,1234567891011...。

并且小Q对于能否被3整除这个性质很感兴趣。

小Q现在希望你能帮他计算一下从数列的第l个到第r个(包含端点)有多少个数可以被3整除。

**输入描述:**

输入包括两个整数l和r(1 <= l <= r <= 1e9), 表示要求解的区间两端。

**输出描述:**

输出一个整数, 表示区间内能被3整除的数字个数。

**输入例子1:**

2 5

**输出例子1:**

3

**例子说明1:**

12, 123, 1234, 12345...  
其中12, 123, 12345能被3整除。

首先这个题需要多组样例读入但题里没说

其次，l、r很大，但是1e9，并没有超int，但是在算i\*(i+1)的时候可能会爆，所以要转ll

#include <stdio.h>

#include<iostream>

#include<cstring>

#include<cmath>

typedef long long ll;

using namespace std;

int main()

{

ll l,r;

while(cin>>l>>r)

{

ll ans(0);

for(int i=l;i<=r;i++)

{

ll t=(ll)i\*(i+1)/2;

if(t%3==0)ans++;

}

cout<<ans<<"\n";

}

return 0;

}

法二：

1. // 神奇的解法，序号%3取余，若是2，0就能整除，1不能整除
2. // 例如：序号1 数字1余数1，不能整除；序号2数字12余数2，能整除；序号3数字123余数0，能整除

#include <stdio.h>

#include<iostream>

#include<cstring>

#include<cmath>

typedef long long ll;

using namespace std;

int main()

{

ll l,r;

while(cin>>l>>r)

{

ll ans(0);

for(int i=l;i<=r;i++)

if(i%3!=1)ans++;

cout<<ans<<"\n";

}

return 0;

}