题目链接

题目描述

一个整数n的阶乘可以写成n!，它表示从1到n这n个整数的乘积。阶乘的增长速度非常快，例如，13!就已经比较大了，已经无法存放在一个整型变量 中；而35!就更大了，它已经无法存放在一个浮点型变量中。因此，当n比较大时，去计算n!是非常困难的。幸运的是，在本题中，我们的任务不是去计算 n!，而是去计算n!最右边的那个非0的数字是多少。例如，5! = 12345 = 120，因此5!最右边的那个非0的数字是2。再如：7! = 5040，因此7!最右边的那个非0的数字是4。请编写一个程序，输入一个整数n(n<=100)，然后输出n! 最右边的那个非0的数字是多少。

输入

输入只有一个整数n。

输出

输出只有一个整数，即n! 最右边的那个非0的数字。

样例输入

6

样例输出

2

首先求一个数的阶乘，有关阶乘的求法就不在解释了，但是我们应该注意到的一点就是，如果当前的这个数比较大的话，呢么他的阶乘肯定就是int 甚至long long int都存不下的。

但是这道题我们并不需要具体求出来这个数的阶乘是多少，我们需要做的就只是要求出来这个阶乘的从后往前数第一个不为0的数字。以前写过求一个数的阶乘的后三位的，我们只需要把每次阶乘乘上后对1000取余即可。这是因为，如果要求后面三位的话，我们只需要知道每次的后三位是什么就可以了，前面的数字不管如何，都对最后三位没有影响。

但是这道题并不能单纯的求出后几位然后找，因为可能后面好几位的全部都是0，我们不能够确定要求的是后几位。所以我们可以把每次计算出来的数先把最后的0全部去掉，再取余，我们虽然是要求的最后一位部位0的，也不能仅仅对于10 取余，因为如果下次乘上求得的数的最后一位为0，且需要进位的话，进位后的数字就不正确了。

#include<iostream>

#include<math.h>

#include<stdio.h>

#include<cstring>

#define ll long long

using namespace std;

int main()

{

// freopen("input.txt","r",stdin);

int n;

int ans;

while(cin>>n)

{

ans=1;

for(int i=1;i<=n;i++)

{

ans\*=i;

while(ans%10==0)

ans/=10;

ans%=1000;

}

while(ans%10==0)

ans/=10;

cout<<ans%10<<"\n";

}

return 0;

}