BEGIN-4**入门训练 Fibonacci数列**

时间限制：1.0s   内存限制：256.0MB

关键字 入门 数列 取模

问题描述

Fibonacci数列的递推公式为：Fn=Fn-1+Fn-2，其中F1=F2=1。

当n比较大时，Fn也非常大，现在我们想知道，Fn除以10007的余数是多少。

输入格式

输入包含一个整数n。

输出格式

输出一行，包含一个整数，表示Fn除以10007的余数。

说明：在本题中，答案是要求Fn除以10007的余数，因此我们只要能算出这个余数即可，而不需要先计算出Fn的准确值，再将计算的结果除以10007取余数，直接计算余数往往比先算出原数再取余简单。

样例输入

10

样例输出

55

样例输入

22

样例输出

7704

数据规模与约定

1 <= n <= 1,000,000。

**参考代码**

**C++**

#include <stdlib.h>

#include <stdio.h>

#define MOD 10007

#define MAXN 1000001

int n, i, F[MAXN];

int main()

{

scanf("%d", &n);

F[1] = 1;

F[2] = 1;

for (i = 3; i <= n; ++i)

F[i] = (F[i-1] + F[i-2]) % MOD;

printf("%d\n", F[n]);

return 0;

}

**C**

#include <stdlib.h>

#include <stdio.h>

#define MOD 10007

#define MAXN 1000001

int n, i, F[MAXN];

int main()

{

scanf("%d", &n);

F[1] = 1;

F[2] = 1;

for (i = 3; i <= n; ++i)

F[i] = (F[i-1] + F[i-2]) % MOD;

printf("%d\n", F[n]);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException{

BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));

String s=reader.readLine();

int n=Integer.valueOf(s);

int f1=1,f2=1,f3=0;

if(n<3){

System.out.print("1");

return;}

for(int i=3;i<=n;i++)

{if(f1>10007)f1=f1%10007;

if(f2>10007)f2=f2%10007;

f3=f1+f2;

f1=f2;

f2=f3;

}

System.out.print(f3%10007);

}

}

BEGIN-3**入门训练 圆的面积**

时间限制：1.0s   内存限制：256.0MB

关键字 入门 实数输出

**参考代码:**

**C++**

#include <stdio.h>

#include <math.h>

int main()

{

int r;

double s, PI;

scanf("%d", &r);

PI = atan(1.0) \* 4;

s = PI \* r \* r;

printf("%.7lf", s);

return 0;

}

**C**

#include <stdio.h>

#include <math.h>

int main()

{

int r;

double s, PI;

scanf("%d", &r);

PI = atan(1.0) \* 4;

s = PI \* r \* r;

printf("%.7lf", s);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.math.BigDecimal;

public class Main {

/\*\*

\* @param args

\* @throws IOException

\*/

public static void main(String[] args) throws IOException {

// TODO Auto-generated method stub

BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

BigDecimal bg = new BigDecimal(bf.readLine());

BigDecimal value = bg.multiply(bg.multiply(new BigDecimal(Math.PI))).setScale(7,BigDecimal.ROUND\_HALF\_UP);

System.out.println(value);

}

}

BEGIN-2**入门训练 序列求和**

时间限制：1.0s   内存限制：256.0MB

关键字 入门 求和公式

问题描述

求1+2+3+...+n的值。

输入格式

输入包括一个整数n。

输出格式

输出一行，包括一个整数，表示1+2+3+...+n的值。

样例输入

4

样例输出

10

样例输入

100

说明：有一些试题会给出多组样例输入输出以帮助你更好的做题。

一般在提交之前所有这些样例都需要测试通过才行，但这不代表这几组样例数据都正确了你的程序就是完全正确的，潜在的错误可能仍然导致你的得分较低。

样例输出

5050

数据规模与约定

1 <= n <= 1,000,000,000。

说明：请注意这里的数据规模。

本题直接的想法是直接使用一个循环来累加，然而，当数据规模很大时，这种“暴力”的方法往往会导致超时。此时你需要想想其他方法。你可以试一试，如果使用1000000000作为你的程序的输入，你的程序是不是能在规定的上面规定的时限内运行出来。

本题另一个要值得注意的地方是答案的大小不在你的语言默认的整型(int)范围内，如果使用整型来保存结果，会导致结果错误。

如果你使用C++或C语言而且准备使用printf输出结果，则你的格式字符串应该写成%I64d以输出long long类型的整数。

**参考代码：**

**C++**

#include <iostream>

using namespace std;

int main()

{

long long n;

cin >> n;

cout << (1+n) \* n / 2;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n;

long long ret = 0, tmp;

scanf("%d", &n);

tmp = n;

printf("%I64d", (1+tmp) \* tmp / 2);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String args[]) throws NumberFormatException, IOException{

BufferedReader strin = new BufferedReader(new InputStreamReader(System.in));

long i = Integer.parseInt(strin.readLine());

long sum = (1+i)\*i/2;

System.out.println(sum);

}

}

BEGIN-1**入门训练 A+B问题**

时间限制：1.0s   内存限制：256.0MB

关键字 入门

问题描述

输入A、B，输出A+B。

说明：在“问题描述”这部分，会给出试题的意思，以及所要求的目标。

输入格式

输入的第一行包括两个整数，由空格分隔，分别表示A、B。

说明：“输入格式”是描述在测试你的程序时，所给的输入一定满足的格式。

做题时你应该假设所给的输入是一定满足输入格式的要求的，所以你不需要对输入的格式进行检查。多余的格式检查可能会适得其反，使用你的程序错误。

在测试的时候，系统会自动将输入数据输入到你的程序中，你不能给任何提示。比如，你在输入的时候提示“请输入A、B”之类的话是不需要的，这些多余的输出会使得你的程序被判定为错误。

输出格式

输出一行，包括一个整数，表示A+B的值。

说明：“输出格式”是要求你的程序在输出结果的时候必须满足的格式。

在输出时，你的程序必须满足这个格式的要求，不能少任何内容，也不能多任何内容。如果你的内容和输出格式要求的不一样，你的程序会被判断为错误，包括你输出了提示信息、中间调试信息、计时或者统计的信息等。

样例输入

12 45

说明：“样例输入”给出了一组满足“输入格式”要求的输入的例子。

这里给出的输入只是可能用来测试你的程序的一个输入，在测试的时候，还会有更多的输入用来测试你的程序。

样例输出

57

说明：“样例输出”给出了一组满足“输出格式”要求的输出的例子。

样例输出中的结果是和样例输入中的是对应的，因此，你可以使用样例的输入输出简单的检查你的程序。

要特别指出的是，能够通过样例输入输出的程序并不一定是正确的程序，在测试的时候，会用很多组数据进行测试，而不局限于样例数据。有可能一个程序通过了样例数据，但测试的时候仍只能得0分，可能因为这个程序只在一些类似样例的特例中正确，而不具有通用性，再测试更多数据时会出现错误。

比如，对于本题，如果你写一个程序不管输入是什么都输入57，则样例数据是对的，但是测试其他数据，哪怕输入是1和2，这个程序也输出57，则对于其他数据这个程序都不正确。

数据规模与约定

-10000 <= A, B <= 10000。

说明：“数据规模与约定”中给出了试题中主要参数的范围。

这个范围对于解题非常重要，不同的数据范围会导致试题需要使用不同的解法来解决。比如本题中给的A、B范围不大，可以使用整型(int)来保存，如果范围更大，超过int的范围，则要考虑其他方法来保存大数。

有一些范围在方便的时候是在“问题描述”中直接给的，所以在做题时不仅要看这个范围，还要注意问题描述。

提示

本题的C++源代码如下：

1. #include <iostream>
3. **using namespace** std**;**
5. **int** main**()**
6. **{**
7. **int** a**,** b**;**
8. cin **>>** a **>>** b**;**
9. cout **<<** a **+** b**;**
10. **return** 0**;**
11. **}**

本题的C源代码如下：

1. #include <stdio.h>
3. **int** main**()**
4. **{**
5. **int** a**,** b**;**
6. scanf**(**"%d%d", **&**a**,** **&**b**);**
7. printf**(**"%d"**,** a**+**b**);**
8. **return** 0**;**
9. **}**

本题的Java源代码如下：

1. **import** java.util.\*;
3. **public** **class** Main
4. {
5. **public** **static** **void** main(String args[])
6. {
7. Scanner sc = **new** Scanner(System.in);
8. Integer a = sc.nextInt();
9. Integer b = sc.nextInt();
10. System.out.println(a + b);
11. }
12. }

说明：要答题，请点击页面上方的“提交此题”按钮，页面将跳转到提交代码的页面，选择好你的编译语言，将你的编写好的代码粘贴到代码框中，再点击“提交答案”即可。

你的答案提交给系统后系统会自动对你的代码进行判分，并跳转到结果的列表里面，你可以直接从列表中看到你提交的代码的状态，一般几秒钟后就可以看到判分的结果。

本题作为第一题，在提示中已经分别给了C++和Java的代码，你可以直接把这个代码拷贝过去作为自己的代码提交。

请特别注意，Java的主类名必须是Main。

**参考代码：**

**C++**

#include <iostream>

using namespace std;

int main()

{

int a, b;

cin >> a >> b;

cout << a + b;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int a, b;

scanf("%d%d", &a, &b);

printf("%d", a+b);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class Main

{

public static void main(String[] args) throws IOException

{

args = new BufferedReader(new InputStreamReader(System.in)).readLine().split(" ");

System.out.println(Integer.parseInt(args[0])+Integer.parseInt(args[1]));

}

}