**算法训练 Torry的困惑(基本型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　Torry从小喜爱数学。一天，老师告诉他，像2、3、5、7……这样的数叫做质数。Torry突然想到一个问题，前10、100、1000、10000……个质数的乘积是多少呢？他把这个问题告诉老师。老师愣住了，一时回答不出来。于是Torry求助于会编程的你，请你算出前n个质数的乘积。不过，考虑到你才接触编程不久，Torry只要你算出这个数模上50000的值。

输入格式

　　仅包含一个正整数n，其中n<=100000。

输出格式

　　输出一行，即前n个质数的乘积模50000的值。

样例输入

1

样例输出

2

本题的C++参考代码如下：

#include <iostream>

using namespace std;

int a[100005];

int main()

{

unsigned int i, j, n, cnt = 1, cj = 2;

cin >> n;

if (n == 1)

{

cout << 2 << endl;

return 0;

}

a[0] = 2;

for (i = 3; i < 2000000; i++)

{

for (j = 0; j < cnt; j++)

{

if (a[j] \* a[j] > i)

{

break;

}

else

{

if (!(i % a[j]))

{

break;

}

}

}

if (a[j] \* a[j] > i)

{

a[cnt++] = i;

cj = (cj % 50000) \* (i % 50000) ;

if (cnt == n)

{

break;

}

}

}

cout << cj % 50000 << endl;

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

int pr[100010];

int top;

int isPrime(int n)

{

int i;

for(i = 0; i < top; i++)

{

if(n % pr[i] == 0)

{

return 0;

}

}

return 1;

}

int findNextPrime(void)

{

int n = pr[top - 1] + 1;

while(!isPrime(n))

{

n++;

}

pr[top++] = n;

return n;

}

int main(void)

{

int i, n;

int result = 2;

scanf("%d", &n);

pr[0] = 2;

top = 1;

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

{

int x = findNextPrime();

result \*= x;

result %= 50000;

}

printf("%d", result);

return 0;

}

本题的Java参考代码如下：

import java.io.\*;

public class Main {

public static void main(String[] args) {

try{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in),1);

String line = br.readLine();

int n = Integer.parseInt(line);

getans(n);

}catch(Exception e){

System.out.println("Message:"+e.getMessage());

}

}

public static void getans(int n){

if(n<1);

else if(n==1)System.out.println(2);

else if(n<7){

long ans = 6,nowp = 3;

for(int i=0;i<n-2;i++){

nowp=nextprime(nowp);

ans\*=nowp;

}

System.out.println(ans);

}else if(n==7)System.out.println(10510);

else if(n<100){

long ans = 10510,nowp = 17;

for(int i=0;i<n-7;i++){

nowp=nextprime(nowp);

ans=ans\*nowp%50000;

}

System.out.println(ans);

}else if(n==100)System.out.println(31090);

else if(n<1000){

long ans = 31090,nowp = 541;

for(int i=0;i<n-100;i++){

nowp=nextprime(nowp);

ans=ans\*nowp%50000;

}

System.out.println(ans);

}else if(n==1000)System.out.println(1570);

else if(n<=10000){

long ans = 1570,nowp = 7919;

for(int i=0;i<n-100;i++){

nowp=nextprime(nowp);

ans=ans\*nowp%50000;

}

System.out.println(ans);

}else if(n==10000)System.out.println(48410);

else if(n<100000){

long ans = 48410,nowp = 115321;

for(int i=0;i<n-100;i++){

nowp=nextprime(nowp);

ans=nowp%50000\*ans%50000;

}

System.out.println(ans);

}else if(n==100000)System.out.println(20710);

else{

long ans = 20710,nowp = 1452433;

for(int i=0;i<n-100;i++){

nowp=nextprime(nowp);

ans=nowp%50000\*ans%50000;

}

System.out.println(ans);

}

}

public static long nextprime(long now){

now+=2;

int check = 0;

while(true){

check = checkprime(now);

if(check==0)

now+=2;

else

break;

}

return now;

}

public static int checkprime(long n){

for(int i=3;i<=Math.sqrt(n);i+=2)

if(n%i==0)return 0;

return 1;

}

}