**蛮力法**

**1610300c02 夏鹏程**

**总分: 74 / 77**

编程题总分: 74 / 77

7-1答案正确 得分: 8 / 8

求整数数组中相差最小的两个元素（称为最接近数）的差值。

**输入格式:**

首先输入数组的长度n，2<=n<=10000，然后输入n个整数作为数组元素。

**输出格式:**

输出数组中相差最小的两个元素的差的绝对值。

**输入样例:**

5

13 26 98 47 55

**输出样例:**

8

#include<stdio.h>

int main()

{ int a[6];

int i,j,y;

int x=100000;

for(i=0;i<6;i++)

scanf("%d",&a[i]);

for(i=0;i<6;i++)

{

for(j=i+1;j<6;j++)

{

if(a[i]<=a[j])

y=a[j]-a[i];

else

y=a[i]-a[j];

if(x>y)

x=y;

}

}

printf("%d",x);

return 0;

}

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 2 ms | 136KB |
| 1 | 答案正确 | 2 ms | 132KB |
| 2 | 答案正确 | 2 ms | 128KB |
| 3 | 答案正确 | 2 ms | 128KB |

7-2答案正确 得分: 8 / 8

将一个给定的真分数化简为最简分数形式。

**输入格式:**

输入两个正整数m与n，1<=m<n<=10000，m作为分子，n作为分母。

**输出格式:**

输出真分数的最简形式。

**输入样例:**

6 8

**输出样例:**

3/4

#include <iostream>

using namespace std;

int MinCommonFactor(int a,int b) //求最小公因数

{ int i;

for(i=2;i<=b;i++)

if(a%i==0&&b%i==0)

return i;

return 0;

}

int main()

{ int n,d;

cin>>n>>d; //分别输入一个真分数的分子和分母

int n1=n,d1=d;

int min=MinCommonFactor(n1,d1);

while(min>1)

{

n1/=min;

d1/=min;

min=MinCommonFactor(n1,d1);

}

cout<<n1<<"/"<<d1<<endl;

}

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 2 ms | 256KB |
| 1 | 答案正确 | 2 ms | 248KB |
| 2 | 答案正确 | 2 ms | 248KB |
| 3 | 答案正确 | 2 ms | 256KB |

7-3部分正确 得分: 4 / 6

求*a*​*n*​​除以m的余数。

**输入格式:**

分别输入整数a，n，m，其中a、n和m均为大于1的整数。

**输出格式:**

输出余数。

**输入样例:**

5 6 7

**输出样例:**

1

#include<stdio.h>

int main()

{ int a,n,m,i,y,x=1;

scanf("%d",&a);

scanf("%d",&n);

scanf("%d",&m);

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

x=a\*x;

y=x%m;

printf("%d",y);

return 0;

}

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 2 ms | 256KB |
| 1 | 答案正确 | 2 ms | 256KB |
| 2 | 答案错误 | 2 ms | 128KB |

7-4答案正确 得分: 15 / 15

1元5角钱人民币兑换5分、2分和1分的硬币（每一种都要有）共100枚，会有很多种兑换方案。请编写程序给出各种兑换方案。

**输入格式:**

输入为一个正整数n，表示要求输出前n种可能的方案。方案的顺序，是按照5分硬币从少到多排列的。

**输出格式:**

显示前n种方案中5分、2分、1分硬币各多少枚。每行显示一种方案，数字之间空一格，最后一个数字后没有空格。

注意：如果全部方案不到n种，就顺序输出全部可能的方案。

**输入样例:**

5

**输出样例:**

1 46 53

2 42 56

3 38 59

4 34 62

5 30 65

#include<stdio.h>

int main()

{ int x,y,z,m,i=0;

scanf("%d",&m);

for(x=1;x<30;x++)

{ for(y=1;y<75;y++)

{ for(z=1;z<100;z++)

{ if(5\*x+2\*y+z==150&&x+y+z==100)

{ printf("%d %d %d\n",x,y,z);

i++;

}

if(i>=m)

break;

}

if(i>=m)

break;

}

if(i>=m)

break;

}

return 0; }

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 1 ms | 128KB |
| 1 | 答案正确 | 1 ms | 128KB |
| 2 | 答案正确 | 1 ms | 128KB |
| 3 | 答案正确 | 2 ms | 128KB |

7-5部分正确 得分: 7 / 8

所谓完数就是该数恰好等于除自身外的因子之和。例如：6=1+2+3，其中1、2、3为6的因子。本题要求编写程序，找出任意两正整数*m*和*n*之间的所有完数。

**输入格式：**

输入在一行中给出2个正整数*m*和*n*（1<*m*≤*n*≤10000），中间以空格分隔。

**输出格式：**

逐行输出给定范围内每个完数的因子累加形式的分解式，每个完数占一行，格式为“完数 = 因子1 + 因子2 + ... + 因子k”，其中完数和因子均按递增顺序给出。若区间内没有完数，则输出“None”。

**输入样例：**

2 30

**输出样例：**

6 = 1 + 2 + 3

28 = 1 + 2 + 4 + 7 + 14

#include <stdio.h>

#include <math.h>

int main()

{ int m, n, i, j;

int sum, index;

int arr[100] = {1};

scanf("%d %d", &m, &n);

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

{ index = 1, sum = 1;

for (j=2; j<=i/2; j++) {

if (i % j == 0) {

sum += j;

arr[index++] = j;

}

}

if (i == sum) {

printf("%d = ", i);

for (j=0; j<index-1; j++) {

printf("%d + ", arr[j]);

}

printf("%d\n", arr[index-1]);

}

}

return 0;

}

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 2 ms | 136KB |
| 1 | 答案正确 | 157 ms | 212KB |
| 2 | 答案正确 | 2 ms | 256KB |
| 3 | 答案错误 | 103 ms | 128KB |
| 4 | 答案正确 | 1 ms | 128KB |

7-6答案正确 得分: 15 / 15

给定一个整数m（50<m<20000），找出小于m的最大的10个素数。

**输入格式:**

输入在一行中给出一个正整数m（50<m<20000）。

**输出格式:**

在一行中按递减顺序输出10个满足条件的素数，每个素数输出占6列。没有其它任何附加格式和字符。

**输入样例:**

229

**输出样例:**

227 223 211 199 197 193 191 181 179 173

**感谢中国青年政治学院的同学修正数据！**

#include<stdio.h>

#include <string.h>

#include <math.h>

int biao[2400];

int main()

{ biao[0] = 2;

int n = 1;

for (int i = 3; i < 20100; i++)

{ int flag = 1;

int x = (int)sqrt((float)i) + 1;

for (int j = 2; j < x; j++)

{ if (i % j == 0)

{

flag = 0;

}

}

if (flag)

{ biao[n] = i;

n++;

}

}

int m;

scanf("%d", &m);

int k;

for (int i = 0; i < m; i++)

{ if (biao[i] >= m)

{ k = i - 1;

break;

}

}

for (int i = k; i > k - 10; i--)

{ printf("%6d", biao[i]);

}

return 0; }

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 11 ms | 256KB |
| 1 | 答案正确 | 11 ms | 256KB |
| 2 | 答案正确 | 11 ms | 256KB |

7-7答案正确 得分: 9 / 9

（原题来自蓝桥杯训练题）话说这个世界上有各种各样的兔子和乌龟，但是研究发现，所有的兔子和乌龟都有一个共同的特点——喜欢赛跑。于是世界上各个角落都不断在发生着乌龟和兔子的比赛，小华对此很感兴趣，于是决定研究不同兔子和乌龟的赛跑。他发现，兔子虽然跑比乌龟快，但它们有众所周知的毛病——骄傲且懒惰，于是在与乌龟的比赛中，一旦任一秒结束后兔子发现自己领先t米或以上，它们就会停下来休息s秒。对于不同的兔子，t，s的数值是不同的，但是所有的乌龟却是一致——它们不到终点决不停止。

然而有些比赛相当漫长，全程观看会耗费大量时间，而小华发现只要在每场比赛开始后记录下兔子和乌龟的数据——兔子的速度v1（表示每秒兔子能跑v1米），乌龟的速度v2，以及兔子对应的t，s值，以及赛道的长度l——就能预测出比赛的结果。但是小华很懒，不想通过手工计算推测出比赛的结果，于是他找到了你——湖北工业大学计算机学院的高才生——请求帮助，请你写一个程序，对于输入的一场比赛的数据v1，v2，t，s，l，预测该场比赛的结果。

**输入格式:**

输入只有一行，包含用空格隔开的五个正整数v1，v2，t，s，l，其中(v1, v2<=100; t<=300; s<=10; l<=10000 且为v1, v2的公倍数)

**输出格式:**

输出包含两行，第一行输出比赛结果——一个大写字母“T”或“R”或“D”，分别表示乌龟获胜，兔子获胜，或者两者同时到达终点。第二行输出一个正整数，表示获胜者（或者双方同时）到达终点所耗费的时间（秒数）。

**输入样例1:**

10 5 5 2 20

**输出样例1:**

D

4

**输入样例2:**

10 5 5 1 20

**输出样例2:**

R

3

**输入样例3:**

10 5 5 3 20

**输出样例3:**

T

4

#include <cmath>

#include <cstdio>

#include <algorithm>

#include <iostream>

using namespace std;

int main()

{ int v1,v2,t,s,L;

int s1=0,t1=0,s2=0,t2=0;

scanf("%d%d%d%d%d",&v1,&v2,&t,&s,&L);

t1+=1;

t2+=1;

s1=s1+v1;

s2=s2+v2;

while(true)

{ if(s1==L&&s2==L)

{ cout<<"D"<<endl;

cout<<t1<<endl;

break;

}

if(s1==L)

{ cout<<"R"<<endl;

cout<<t1<<endl;

break;

}

if(s2==L)

{ cout<<"T"<<endl;

cout<<t2<<endl;

break;

}

if(s1-s2>=t)

{ t1+=s;

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

{ t2+=1;

s2+=v2;

if(s2==L)

{

cout<<"T"<<endl;

cout<<t2<<endl;

exit(0);

}

}

}

else

{ t1+=1;

t2+=1;

s1=s1+v1;

s2=s2+v2;

}

}

return 0; }

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 1 | 答案正确 | 2 ms | 252KB |
| 2 | 答案正确 | 2 ms | 256KB |
| 3 | 答案正确 | 2 ms | 256KB |
| 4 | 答案正确 | 2 ms | 256KB |
| 5 | 答案正确 | 2 ms | 384KB |
| 6 | 答案正确 | 2 ms | 252KB |
| 7 | 答案正确 | 2 ms | 252KB |
| 8 | 答案正确 | 2 ms | 252KB |
| 9 | 答案正确 | 2 ms | 252KB |

7-8答案正确 得分: 8 / 8

从输入中读取一个数n，求出n！中末尾0的个数。

**输入格式:**

输入有若干行。第一行上有一个整数m，指明接下来的数字的个数。然后是m行，每一行包含一个确定的正整数n，1<=n<=1000000000。

**输出格式:**

对输入行中的每一个数据n，输出一行，其内容是n！中末尾0的个数。

**输入样例:**

3

3

100

1024

**输出样例:**

0

24

253

#include<iostream>

using namespace std;

long long min(long long a,long long b)

{ return a<b?a:b;

}

int main()

{ long long m,n;

cin>>m;

while(m--)

{ cin>>n;

long long num2=0,num5=0,result;

result=1;

while(n/(result\*=2))

num2+=n/result;

result=1;

while(n/(result\*=5))

num5+=n/result;

cout<<min(num2,num5)<<endl;

}

return 0;

}

| **测试点** | **结果** | **耗时** | **内存** |
| --- | --- | --- | --- |
| 0 | 答案正确 | 3 ms | 376KB |
| 1 | 答案正确 | 2 ms | 384KB |
| 2 | 答案正确 | 2 ms | 248KB |