**1-6章题库**

找出2到100之间的所有素数，并保存到out.txt中，每个素数占一行。

例如：out.txt

2

3

5

7

#include<fstream>

using namespace std;

int main()

{ int a,b,x;

ofstream outfile;

outfile.open("out.txt");

for(a=2;a<101;a++)

{ x=1;

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

if(a%b==0)

x=0;

if(x==1)

outfile<<a<<'\n';

}

outfile.close();

return 0;

}

从in.txt读取一个正整数n

求1!+2！+3！+...+n!的和。

结果输出到out.txt。

建议：结果的数据类型用long int。

如:

in.txt

4

则out.txt

33

#include<fstream>

using namespace std;

int main()

{ int n,a,b;

long int s=1,x=0;

ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

infile>>n;

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

{ for(b=a;b>0;b--)

s=s\*b;

x=x+s;

s=1;

}

outfile<<x;

return 0;

}

从输入文件in.txt中读取所有的整数(整数个数大于2)，

输出最小值整数

如:in.txt

34 5 78 9 2 89 -8 -1903 3234

则:out.txt

-1903

#include<fstream>

using namespace std;

int main()

{ int b,c,min=0;

ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int a[9];

for(b=0;b<9;b++)

infile>>a[b];

for(c=0;c<9;c++)

if(a[c]<a[min])

min=c;

outfile<<a[min];

return 0;

}

从输入文件in.txt中读取所有的整数(整数个数大于2)，

输出最大值整数

如:in.txt

34 5 78 9 2 89 -8 -1903 3234

则:out.txt

3234

#include<fstream>

using namespace std;

int main()

{ int b,c,max=0;

ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int a[9];

for(b=0;b<9;b++)

infile>>a[b];

for(c=0;c<9;c++)

if(a[c]>a[max])

max=c;

outfile<<a[max];

return 0;

}

法二：

#include <fstream>

using namespace std;

int main( ){

ifstream infile("in.txt",ios::in);

ofstream outfile("out.txt",ios::out);

int a,max;

infile>>max;

while(infile>>a)

max=max>a?max:a;

outfile<<max<<endl;

infile.close();

outfile.close();

return 0;

}

已知XYZ+YZZ=532，其中X,Y和Z为数字，编一程序求出X,Y和Z。必须用循环结构。

并依次输出x,y,z并换行到out.txt，如果有多组解请输出多行。

#include<fstream>

using namespace std;

int main()

{ int a,b,x,y,z;

ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

for(x=0;x<=9;x++)

for(y=0;y<=9;y++)

for(z=0;z<=9;z++)

{ a=100\*x+10\*y+z;

b=100\*y+10\*z+z;

if(a+b==532)

outfile<<x<<","<<y<<","<<z<<'\n';

}

return 0;

}

编写函数fun用选择对数组进行从小到大排序。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

#include <iostream>

#include <fstream>

using namespace std;

void fun(int a[],int n)

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i,j,min,t;

for(i=0;i<n-1;i++)

{ min=i;

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

if(a[min]>a[j])

min=j;

t=a[i];

a[i]=a[min];

a[min]=t;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int a[10]={3,2,13,-1,3,7,10,5,6,11},i;

cout<<"排序前原数组为："<<endl;

for(i=0;i<10;i++)

cout<<a[i]<<" ";

fun(a,10);

cout<<endl<<"从小到大排序之后数组为："<<endl;

for(i=0;i<10;i++)

cout<<a[i]<<" ";

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int iIN[10],i;

for(i=0;i<10;i++)

{

IN>>iIN[i];

}

fun(iIN,10);

for(i=0;i<10;i++)

{

OUT<<iIN[i]<<'\n';

}

IN.close();

OUT.close();

}

编写函数fun:实现字符串的复制，即将str1中的字符串复制到str2中。（不允许用字符串库函数strcpy）。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

#include<iostream>

#include<fstream>

using namespace std;

void fun(char str1[],char str2[])

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i;

for(i=0;str1[i]!='\0';i++)

str2[i]=str1[i];

str2[i]='\0';

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

char c1[40],c2[40];

cout<<"请输入要复制的字符串：\n";

cin.getline(c1,40);

fun(c1,c2);

cout<<"复制的结果："<<c2;

xxx();

return 0;

}

void xxx()

{

ifstream inf;

ofstream outf;

inf.open("in.txt",ios::in);

outf.open("out.txt",ios::out);

int i;

char s1[40],s2[40];

while(!inf.eof())

{

inf.getline(s1,40);

fun(s1,s2);

outf<<s2;

}

inf.close();

outf.close();

}

编写函数fun生成杨辉三角（要求8行）。杨辉三角组成如下：

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

.....

#include <iostream>

#include <fstream>

using namespace std;

void fun(int a[8][8])

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i,j;

for(i=0;i<8;i++)

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

if(j==0||i==j)

{a[i][j]=1;

}

else {a[i][j]=a[i-1][j-1]+a[i-1][j];

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int i,j;

int a[8][8]={0};

fun(a);

for(i=0;i<8;i++)//输出杨辉三角

{

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

{

cout<<a[i][j]<<' ';

}

cout<<endl;

}

xxx();

return 0;

}

void xxx()

{

/\* 本函数用于打开文件，输入测试数据，调用fun函数，输出数据，关闭文件。\*/

ofstream outf;

outf.open("out.txt",ios::out);

int b[8][8]={0},i,j;

fun(b);

for(i=0;i<8;i++)

{

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

{

outf<<b[i][j]<<' ';

}

outf<<endl;

}

outf.close();

}

从in.txt顺序读取所有的字符。把读取的顺序数字字符组成整数，输出到out.txt中。

每五个整数一行，整数与整数之间用一个空格隔开，第五个整数后没有空格直接换行。

最后一行未满五个整数的，最后一个整数后没有空格直接换行输出。

注意格式。

格式如下：

如:in.txt

)yR[6.[L`56\*LVRA<%S<xu>267eT]r>Y~S523]/H80\X"E12D1\)

则out.txt

6 56 267 523 80 12 1

#include<fstream>

using namespace std;

int main()

{ ifstream inf("in.txt",ios::in);

if(!inf)

return 1;

ofstream outf("out.txt",ios::out);

char c; //c变量为读取

int t=0, s=0; //s变量为读取的整数，t变量为整数的个数

bool flag=false; //false用来表示前一个字符是否为数字，是则为数字，否为不是数字

while(inf>>c)

{ if(c>='0'&&c<='9')

{ s=s\*10+c-'0';

flag=true;

}

else

{ if(flag) //flag为true，表示以前读取的数需要输出了

{ if(t!=0)

{ if(t%5==0)

outf<<endl; //五个一行

else

outf<<' '; //未满5个用空格隔开

}

outf<<s;

t++;

s=0;

}

flag=false;

}

}

//最后一个数据的处理

if(flag) //flag为true，表示以前读取的数需要输出

{ if(t!=0)

{ if(t%5==0)

outf<<endl; //五个一行

else outf<<' '; //未满五个用空格隔开

}

outf<<s;

}

outf<<endl; //如果结束的话最后一个数也要换行

inf.close();

outf.close();

return 0;

}

/\* 求满足条件n!<m的最大数n。

m为正整数，由in.txt读入,

将结果n输出到out.txt文件中。

如:in.txt

30

则:out.txt

4

#include<fstream>

#include<iostream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int m,n,s=1,i,j;

infile>>m;

for(n=10;n>1;n--)

{

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

{ s=s\*i;

}

if(s<m)

{ outfile<<n;

break;

}

s=1;

}

infile.close();

outfile.close();

}

从输入文件in.txt中读取所有的整数(整数个数大于2)，

输出这些整数的平均值 .

如:in.txt

34 5 78 9 2 89 -8 -1903 3234

则:out.txt

171.111

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int a[9],i,sum=0,n;

for(i=0;i<9;i++)

{ infile>>a[i];

}

for(i=0;i<9;i++)

{ sum=sum+a[i];

}

n=sum/9;

outfile<<n;

return 0;

}

/\* 某女士手里拎了一篮鸡蛋，从她身边奔跑而过一匹惊马，

吓了她一跳，结果把篮里的鸡蛋打翻了，

她说两个一数，三个一数，四个一数，五个一数时，

余数分别为1，2，3和4。

问篮里原有多少个鸡蛋？(求出最小数)

通过程序编程算出。

将结果输出到out.txt文件中。

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ofstream outfile;

outfile.open("out.txt");

int n;

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

{ if(n%2==1&&n%3==2&&n%4==3&&n%5==4)

{outfile<<n;

break;

}

}

return 0;

}

/\*

编写函数fun实现矩阵（3行3列）的转置（即行列互换）

例如：输入下面的矩阵：

100 200 300

400 500 600

700 800 900

转置结果：

100 400 700

200 500 800

300 600 900

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

void fun(int array[3][3])

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i,j,t;

for(i=0;i<3;i++)

{ for(j=i;j<3;j++) //从对角线上方开始换，若从j=1开始换会换两次

{ t=array[i][j];

array[i][j]=array[j][i];

array[j][i]=t;

}

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int i,j;

int array[3][3]={{100,200,300},{400,500,600},{700,800,900}};

for (i=0; i < 3; i++)

{

for (j=0; j < 3; j++)

cout<<array[i][j]<<' ';

cout<<'\n';

}

fun(array);

cout<<"Converted array:"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

cout<<array[i][j]<<' ';

cout<<'\n';

}

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int i,j, array[3][3];

for (i=0; i < 3; i++)

for (j=0; j < 3; j++)

IN>>array[i][j];

fun(array);

for (i=0; i < 3; i++)

{

for (j=0; j < 3; j++)

OUT<<array[i][j]<<' ';

OUT<<'\n';

}

IN.close();

OUT.close();

}

/\* 求两个正整数m,n的最大公约数。

两个正整数m,n从in.txt循环读入,直到全部读完，

并将结果输出到out.txt文件中。

如:in.txt

24 12

34 67

8 6

5 89

23 46

则:out.txt

12

1

2

1

23

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<fstream>

using namespace std;

int fun(int m,int n)

{

int i,t;

if(m<n)

{

t=m;

}

else

{

t=n;

}

for(i=t;i>=1;i--)

{

if(m%i==0&&n%i==0)//i是m和n公因子

{

return i;

}

}

}

int main()

{

ifstream infile;

ofstream outfile;

infile.open("in.txt");

outfile.open("out.txt");

int m,n,t,i;

while(!infile.eof())//判读文件是否到尾部

{

infile>>m>>n;

t=fun(m,n);

outfile<<t<<'\n';

}

infile.close();

outfile.close();

return 0;

}

编写函数fun：将字符串中的大写字母转换为对应的小写字母，其它字符不变。

例如，输入：

HELLO world!

结果：

hello world!

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

void fun(char str[])

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i;

for(i=0;str[i]!='\0';i++)

if(str[i]>='A' && str[i]<='Z')

str[i]=str[i]+32;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

char str[40];

cout<<"请输入字符串\n";

cin.getline(str,40);

fun(str);

cout<<"大写字母转换成小写字母的结果：\n"<<str;

xxx();

return 0;

}

void xxx()

{

/\* 本函数用于打开文件，输入测试数据，调用fun函数，输出数据，关闭文件。\*/

ifstream inf;

ofstream outf;

inf.open("in.txt",ios::in);

outf.open("out.txt",ios::out);

char s1[40];

inf.getline(s1,40);

fun(s1);

outf<<s1;

inf.close();

outf.close();

}

/\* 从in.txt读取整数n,根据n来输出一个图案到out.txt。(n大于2小于20的正整数)

如:in.txt

3

则out.txt

\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*

如:in.txt

5

则:out.txt

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int i,j,k,n;

infile>>n;

for(i=1;i<=n;i++) //i表示行数（前n行）

{ for(j=1;j<=n-i;j++) //j表示\*前的空格数

{ outfile<<" ";}

for(k=1;k<=2\*i-1;k++) //k表示开始输出\*

{ outfile<<"\* ";}

outfile<<'\n';

}

for(i=n-1;i>0;i--) //i表示行数（前n行）

{ for(j=1;j<=n-i;j++) //j表示\*前的空格数

{ outfile<<" ";}

for(k=1;k<=2\*i-1;k++) //k表示开始输出\*

{ outfile<<"\* ";}

outfile<<'\n';

}

infile.close();

outfile.close();

return 0;

}

/\*

菲波那契数列:1, 1, 2, 3, 5, 8...,从第三项开始，每个项等于前两项之和。求菲波那契数列的前n项之和。（n>3）,

n从in.txt读入，计算结果保存到out.txt中 。

如:

in.txt

8

则out.txt

54

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int n,sum=0,i,j;

infile>>n;

int a[n];

a[0]=1;

a[1]=1;

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

{ a[i+2]=a[i]+a[i+1];}

for(j=0;j<=n-1;j++)

{ sum=sum+a[j];

}

outfile<<sum;

infile.close();

outfile.close();

return 0;

}

/\*

编程计算1+2+3+...+n-1+n的值。

n(小于100)从in.txt读取，结果输出到out.txt中

如:

in.txt

6

则out.txt

21

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include <fstream>

using namespace std;

int main()

{

ifstream inf("in.txt",ios::in);

ofstream outf("out.txt",ios::out);

int n,i,sum=0;

inf>>n;

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

{ sum=sum+i;

}

outf<<sum;

inf.close();

outf.close();

return 0;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

/\* 学校有象棋、跳棋共26副，

恰好可供120个学生同时进行活动.

象棋2人下一副棋，跳棋6人下一副.

象棋和跳棋各有几副？

通过程序编程算出。

将结果输出到out.txt文件中。

按照象棋数、跳棋数（两数之间用一个空格隔开）的顺序输出后换行。

有多组值时，就换行输出多组。

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int a,b; //a表示象棋数，b表示跳棋数

for(a=0;a<=26;a++)

{ for(b=0;b<=26;b++)

{ if(2\*a+6\*b==120&&a+b==26)

outfile<<a<<'\t'<<b<<'\n';

}

}

infile.close();

outfile.close();

return 0;

}

编写函数fun删除字符串中的数字字符。

例如：字符串内容为：12adDF\*FD768,结果：adDF\*FD

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

#include<cstring>

using namespace std;

void fun(char str[])

{

/\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i=0;

while(str[i]) //遍历数组

{ if(str[i]>='0'&&str[i]<='9') //当前字符为数字字符

{ strcpy(str+i,str+i+1);

i--; //当前字符保持不变，下一循环继续判断覆盖过的当前字符

}

i++;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

char str[40];

cout<<"请输入字符串"<<endl;

cin.getline(str,40);

fun(str);

cout<<endl<<"删除数字字符之后是"<<str;

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

char k[100];

IN>>k;

fun(k);

OUT<<k<<'\n';

IN.close();

OUT.close();

}

/\*

编写函数fun用选择对数组进行从小到大排序。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

void fun(int a[],int n)

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i,j,min,t;

for(i=0;i<n-1;i++)

{ min=i;

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

{ if(a[min]>a[j])

min=j;

}

t=a[i];

a[i]=a[min];

a[min]=t;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int a[10]={3,2,13,-1,3,7,10,5,6,11},i;

cout<<"排序前原数组为："<<endl;

for(i=0;i<10;i++)

cout<<a[i]<<" ";

fun(a,10);

cout<<endl<<"从小到大排序之后数组为："<<endl;

for(i=0;i<10;i++)

cout<<a[i]<<" ";

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int iIN[10],i;

for(i=0;i<10;i++)

{

IN>>iIN[i];

}

fun(iIN,10);

for(i=0;i<10;i++)

{

OUT<<iIN[i]<<'\n';

}

IN.close();

OUT.close();

}

/\*

编写函数fun将数组逆序存放。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

void fun(int a[],int m)

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int k,i;

for(i=0;i<m/2;i++)

{ k=a[i];

a[i]=a[m-1-i];

a[m-1-i]=k;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int i;

int a[10]={12,34,54,13,15,16,56,45,3,5};

cout<<"原数组是";

for(i=0;i<10;i++)

cout<<a[i]<<" ";

fun(a,10);

cout<<endl<<"逆序数组";

for(i=0;i<10;i++)

cout<<a[i]<<" ";

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int iIN[10],i;

for(i=0;i<10;i++)

{

IN>>iIN[i];

}

fun(iIN,10);

for(i=0;i<10;i++)

{

OUT<<iIN[i]<<'\n';

}

IN.close();

OUT.close();

}

/\*

编写函数fun,求5行5列矩阵的主、副对角线上元素之和。注意，两条对角线相交的元素只加一次。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

int fun(int a[5][5])

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int s=0;

int i,j;

for(i=0;i<5;i++)

s=s+a[i][i]+a[i][4-i];

s=s-a[2][2];

return s;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int a[5][5]={{1,3,5,7,9},{2,4,6,8,10},{2,3,4,5,6},{4,5,6,7,8},{1,3,4,5,6}};

int y,i,j;

cout<<"原矩阵是："<<endl;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

cout<<a[i][j]<<" ";

}

cout<<endl;

}

y=fun(a);

cout<<"对角线和是："<<y;

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int iIN[5][5],iOUT;

int i,j;

for(i=0;i<5;i++)

for(j=0;j<5;j++)

IN>>iIN[i][j];

iOUT=fun(iIN);

OUT<<iOUT;

IN.close();

OUT.close();

}

百钱百鸡

/\*

我国古代难题百鸡问题：鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。

百钱买百鸡，问鸡翁、母、雏各几何？

结果保存到out.txt中，格式举例如下：(注意分隔符逗号，为英文逗号）

鸡翁1,鸡母1,鸡雏1

鸡翁1,鸡母1,鸡雏1

....

注意：完善现有代码，不要删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<math.h>

#include<fstream>

using namespace std;

int main()

{ ofstream outfile("out.txt");

int x,y,z;

for(x=0;x<20;x++) //Rooster less than 100/5

{

for(y=0;y<(100-5\*x)/3;y++)

{

z=100-x-y;

if(z%3==0 && 5\*x+3\*y+(100-x-y)/3==100)

outfile<<"鸡翁"<<x<<","<<"鸡母"<<y<<","<<"鸡雏"<<z<<endl;

}

}

return 0;

}

输出正整数的位数和它的各位数字之和

从in.txt读取一个正整数，输出它的位数和它的各位数字之和。

结果输出到out.txt。

格式如下：

位数与各位数字之和用两行打印。

如:

in.txt

8349

则out.txt

4

24

注意：

1、请勿删除注释语句。

2、把代码写入下面的begin与end之间。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile("out.txt");

int n,sum=0,t=0;

infile>>n;

do

{ sum=sum+n%10;

n=n/10;

t=t+1;

}

while(n>0);

outfile<<t<<endl<<sum<<endl;

return 0;

}

/\*

给3个正整数a、b、c，判断是否是三角形的三条边。是的话，输出为Y,否的话，输出为N

三角形三条边满足条件：任意两边之和大于第三边，并且任意两边之差小于第三边。

a,b,c从in.txt读取，结果输出到out.txt中

如:

in.txt

3 4 5

则out.txt

Y

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

int a,b,c,x,y,z;

infile>>a>>b>>c;

x=a+b;

y=a+c;

z=b+c;

if(x>c&&y>b&&z>a) outfile<<"Y";

else outfile<<"N";

return 0;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

/\* 在爱尔兰守神节那天，举行每年一度的庆祝游戏，

指挥者若将乐队排成10人、9人、8人、7人、6人、5人、4人、3人和2人一排时，

最后的一排总是缺少一个人，

那些人想这个位置大概是给数月前死去的乐队成员凯西还留着位置。

指挥者见到总缺一人恼火了，叫大家排成一列纵队前进。

那么乐队究竟有多少人？(求出最小数)

通过程序编程算出。

将结果输出到out.txt文件中。

注意：请勿删除注释语句。

---------------------------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ ofstream outfile;

outfile.open("out.txt");

int x,i,a=0;

for(x=10;x<10000;x++)

{ for(i=2;i<11;i++)

{ if(x%i==1)

a=a+1;

}

if(a==9)

{outfile<<x;

break;

}

else a=0;

}

return 0;

}

找出数组的最大值

编写函数fun找出数组的最大值。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include <iostream>

#include <fstream>

using namespace std;

int fun(int a[],int n)

{

/\*\*\*\*\*\*\*\*\*\*Begin\*\*\*\*\*\*\*\*\*\*/

int i,max;

max=a[0];

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

{ if(max<a[i])

max=a[i];

}

return max;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

int a[10]={3,2,13,-1,3,7,10,5,6,11},i,max;

cout<<"原数组为："<<endl;

for(i=0;i<10;i++)

cout<<a[i]<<" ";

max=fun(a,10);

cout<<endl<<"数组最大值为："<<max;

xxx();

return 0;

}

void xxx()

{

ifstream IN("in.txt",ios::in);

ofstream OUT;

OUT.open("out.txt",ios::out);

int iIN[10],i,max;

for(i=0;i<10;i++)

{

IN>>iIN[i];

}

max=fun(iIN,10);

OUT<<max;

IN.close();

OUT.close();

}

**特殊图案**

/\* 从in.txt顺序读取一个字符a和一个整数n,

根据a和n来输出一个图案到out.txt。

(n大于2小于20的奇数)

如:in.txt

A 7

则out.txt

A

AAA

AAAAA

AAAAAAA

AAAAA

AAA

A

如:in.txt

B 11

则:out.txt

B

BBB

BBBBB

BBBBBBB

BBBBBBBBB

BBBBBBBBBBB

BBBBBBBBB

BBBBBBB

BBBBB

BBB

B

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<fstream>

using namespace std;

int main()

{ int n,sum=0,t=0;

ifstream infile("in.txt",ios::in);

ofstream outfile;

outfile.open("out.txt");

char c;

infile>>c>>n;

int i ,j;

for(i=1;i<=1+n/2;i++)

{

for(j=1;j<=1+n/2-i;j++)

outfile<<" ";

for(j=1;j<=2\*i-1;j++)

outfile<<c;

outfile<<endl;

}

for(i=i-2;i;i--)

{ for(j=1;j<=1+n/2-i;j++)

outfile<<" ";

for(j=1;j<=2\*i-1;j++)

outfile<<c;

outfile<<endl;

}

outfile.close();

infile.close();

return 0;

}

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

**两个字符串的连接**

/\*

编写函数fun：实现两个字符串的连接（不使用库函数strcat）。

例如，分别输入下面两个字符串：

FirstString--

SecondString

程序输出：

FirstString--SecondString

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句

---------------------------------------------------------------------------------------------------\*/

#include<iostream>

#include<fstream>

using namespace std;

void fun(char p1[],char p2[])

{

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

int i,j;

for(i=0;p1[i]!='\0';i++)//找到第一个字符串尾部

{

}

for(j=0;p2[j]!='\0';j++)//将第二个字符串插入到第一个字符串尾部

{

p1[i]=p2[j];

i++;

}

p1[i]='\0';

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

void xxx();

char s1[100],s2[40];

cout<<"请输入两个字符串\n";

cin.getline(s1,40);

cin.getline(s2,40);

fun(s1,s2);

cout<<"字符串连接结果："<<s1;

xxx();

return 0;

}

void xxx ( )

{/\* 本函数用于打开文件，输入测试数据，调用fun函数，输出数据，关闭文件。\*/

ifstream inf;

ofstream outf;

inf.open("in.txt",ios::in);

outf.open("out.txt",ios::out);

char s1[100], s2[40] ;

inf.getline(s1,40);

inf.getline(s2,40);

fun(s1,s2);

outf<<s1;

inf.close();

outf.close();

}

/\* 从in.txt顺序读取所有的字符。把读取的顺序数字字符组成整数，输出到out.txt中。

每个数字占一行

注意格式。

格式如下：

如:in.txt

)yR[6.[L`56\*LVRA<%S<xu>267eT]r>Y~S523]/H80\X"E12D1\)

则out.txt

6

56

267

523

80

12

1

#include<iostream>

#include<fstream>

#include<cstring>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

char ch[200];

infile>>ch;

int n=strlen(ch);

int i=0;

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

{ if(ch[i]>='0'&&ch[i]<='9')

{ do

{ outfile<<ch[i];

i++;

} while(ch[i]>='0'&&ch[i]<='9');

outfile<<endl;

}

}

return 0;

}

/\* 从in.txt顺序读取所有的字符,统计字符（空格也是字符）。

统计出所有的大写字母个数，小写字母个数，阿拉伯数字个数及其他字符个数，

要求顺序输出大写字母个数，小写字母个数，阿拉伯数字个数，其他字符个数及总字符个数到out.txt.

每输出一个数换行。

注意格式。

格式如下：

如:in.txt

/t X8;qYT | -K

则out.txt

4

2

1

7

14

注意：请勿删除注释语句。

#include<iostream>

#include<cstring>

#include<fstream>

using namespace std;

int main()

{ ifstream infile("in.txt");

ofstream outfile;

outfile.open("out.txt");

char ch[200];

infile>>ch;

int a=0,b=0,c=0,d;

int n=strlen(ch);

int i=0;

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

{ if(ch[i]>='A'&&ch[i]<='Z')

a++;

if(ch[i]>='a'&&ch[i]<='z')

b++;

if(ch[i]>='0'&&ch[i]<='9')

c++;

}

d=n-a-b-c;

outfile<<a<<'\n'<<b<<'\n'<<c<<'\n'<<d;

return 0;

}