**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

法一：

#include <fstream>  
#include <iostream>  
using namespace std;  
int main()  
{  
ifstream infile("in.txt");  
ofstream outfile("out.txt");  
int a[20];  
int i=0;  
while(!infile.eof())  
{  
    infile>>a[i++];  
}  
  
int sum=0;  
for(int j=0;j<i;j++)  
{  
sum=sum+a[j];  
}  
outfile<<sum/i<<&apos;\n&apos;;  
return 0;  
 }

法二：

#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();

}

法二： 思想：用新数组储存数据

#include<iostream>

#include<cstring>

using namespace std;

int main()

{ char b[]="12aaahwudhwudw357kjcw";

int i=0,j=0;

char c[100];

for(i=0;b[i];i++)

{ if(b[i]<'0'||b[i]>'9')

{ c[j]=b[i];

j++;

}

}

c[j]='\0';

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

{ cout<<c[i];

}

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。

法一：

#include <fstream>

using namespace std;

int main( ){

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

if(!inf)

return 1;

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

char s;

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

while(inf.get(s)){

if(s>='A'&&s<='Z'){//大写字母个数计数

a++;

}

else if(s>='a'&&s<='z'){//小写字母个数计数

b++;

}

else if(s>='0'&&s<='9'){//数字字符个数计数

c++;

}

else

d++;//其他字符个数计数

}

outf<<a<<endl<<b<<endl<<c<<endl<<d<<endl<<a+b+c+d<<endl;

inf.close();

outf.close();

return 0;

}

/\*

编写函数fun统计字符串中数字字符出现的次数,没有返回0。

例如:字符串为：abcdef35adgh3kjsdf7，统计结果为4。

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 <string.h>

using namespace std;

int fun(char s[])

{

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

int i,n;

n=0;

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

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

n++;

return n;

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

}

int main()

{ void xxx();

char s[80];

int t;

cout<<"请输入字符串：";

cin.getline(s,80);

t=fun(s);

cout<<endl<<"该字符串中数字字符出现的次数是："<<t;

xxx();

return 0;

}

void xxx()

{

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

ofstream OUT;

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

char k[100];

int i,iOUT;

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

{

IN>>k;

iOUT=fun(k);

OUT<<iOUT<<'\n';

}

IN.close();

OUT.close();

}

}

/\*

从输入文件in.txt中读取1个整数n，

输出蛤蟆调歌词到out.txt。

格式如下：

in.txt

5

out.txt

1load,1mouth,2eyes,4legs,putong,jumps into the pool.

2loads,2mouths,4eyes,8legs,putong,putong,jump into the pool.

3loads,3mouths,6eyes,12legs,putong,putong,putong,jump into the pool.

4loads,4mouths,8eyes,16legs,putong,putong,putong,putong,jump into the pool.

5loads,5mouths,10eyes,20legs,putong,putong,putong,putong,putong,jump into the pool.

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,i,t;

infile>>n;

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

{ t=i;

if(i==1) {outfile<<i<<"load"<<","<<i<<"mouth"<<","<<2\*i<<"eyes"<<","<<4\*i<<"legs"<<","<<"putong"<<","<<"jumps into the pool"<<endl;

continue;

}

outfile<<i<<"loads"<<","<<i<<"mouths"<<","<<2\*i<<"eyes"<<","<<4\*i<<"legs"<<",";

while(t>0)

{ outfile<<"putong,";

t--;

}

outfile<<"jump into the pool"<<endl;

}

return 0;

}

/\* 求两个正整数m,n的最小公倍数。

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

45 9

20 40

则:out.txt

24

2278

24

45

40

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=i+t)

{

if(i%m==0&&i%n==0)//i是m和n的公倍数

{

return i;

}

}

}

int main()

{

ifstream infile;

ofstream outfile;

infile.open("in.txt");

outfile.open("out.txt");

int m,n,t,i;

while(!infile.eof())

{

infile>>m>>n;

t=fun(m,n);

outfile<<t<<'\n';

}

infile.close();

outfile.close();

return 0; /\*

/\*

编写函数fun生成一个两个对角线元素为1，其他元素都为0的3\*3的二维数组。

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 arr[][3])

{

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

int i,j;

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

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

if ((i==j)||(i+j)==2 )

arr[i][j]=1;

else

arr[i][j]=0;

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

}

int main()

{

void xxx();

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

fun(a);

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

{

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

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

cout<<"\n";

}

xxx();

return 0;

}

void xxx()

{

ofstream OUT;

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

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

fun(iIN);

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

{

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

{

OUT<<iIN[i][j];

}

OUT<<'\n';

}

OUT.close();

}

/\* 从in.txt顺序读取一个正整数n,

根据n来输出一个图案到out.txt。

(n大于2小于10的正整数)

如:in.txt

7

则out.txt

A

BBB

CCCCC

DDDDDDD

CCCCC

BBB

A

如:in.txt

11

则:out.txt

A

BBB

CCCCC

DDDDDDD

EEEEEEEEE

FFFFFFFFFFF

EEEEEEEEE

DDDDDDD

CCCCC

BBB

A

注意：

1、out.txt中间行顶格打印图案，即前面没有空格

2、请勿删除注释语句。

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

【参考代码】

#include <fstream>

using namespace std;

int main() {

ifstream inf("in.txt",ios::in);//声明inf文件输入流对象，并绑定in.txt文件，以便从in.txt读取数据

ofstream outf("out.txt",ios::out);//声明outf文件输出流对象，并绑定out.txt文件，以便输出数据到out.txt

int n;

inf>>n;

int i,j;

char c;

for(i=1,c='A';i<=1+n/2;i++,c++){//上三角形

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

outf<<' ';

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

outf<<c;

outf<<endl;

}

for(i=i-2,c=c-2;i;i--,c--){ //下三角形

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

outf<<' ';

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

outf<<c;

outf<<endl;

}

outf.close();

inf.close();

return 0;

}

**指针 题库**

/\*插入字符串

编写一个函数实现：把一个字符串插入到另一个字符串中的指定位置。

void copys2s(char \* src,char \* dest,int loc)

该函数实现将字符串dest插入字符串src中loc的位置。该函数定义可以使用字符串处理函数。

如：in.txt

abcdef

123

3

abcdef

123

1

abcdef

123

7

则：out.txt

ab123cdef

123abcdef

abcdef123

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include<iostream>

#include <cstring>

using namespace std;

void copys2s(char \* src,char \* dest,int loc)

{

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

法一：void copys2s(char \* src,char \* dest,int loca)

{

char temp[100];

strcpy(temp,src+loca-1);//把src中loca位置开始字符复制到temp

strcpy(src+loca-1,dest);//把dest的所有字符复制到src的loca位置后面

loca=strlen(src);//统计src字符个数

strcpy(src+loca,temp);//把temp中所有字符复制到src已有字符后面

}

法二：char temp[100];

strcpy(temp,src+loca-1);

strcpy(src+loca-1,dest);

Strcat(src,temp);

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

}

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

char src[100],dest[100];

int loc;

while(cin>>src>>dest>>loc)

{

copys2s(src,dest,loc);

cout<<src<<endl;

}

file\_in.close();

file\_out.close();

cin.rdbuf(std\_in);

cout.rdbuf(std\_out);

return 0;

}

/\*字符串的选择排序

使用选择排序法实现：将若干（5个）字符串（长度不超过30）按字母顺序（由小到大）输出。编写函数实现排序过程：

void selectSort(char (\*alpha)[30], int n)

其中alpha存放5个字符串的首地址（行指针），n表示字符串的个数5

输入5个字符串（长度不超过30），每个字符串占一行，以回车结束。输出从小到大排序后的字符串，每个字符串占一行，以回车结束。

如：in.txt

Follow me

Basic

Great Wall

FORTRAN

Computer design

则:out.txt

Basic

Computer design

FORTRAN

Follow me

Great Wall

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include <iostream>

#include <cstring>

using namespace std;

const int N=5;

const int M=30;

void selectSort(char (\*alpha)[M], int n)

{

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

void selectSort(char (\*alpha)[M], int n)

{

int i, j, k;

char temp[M];

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

{ k=i;

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

if(strcmp(alpha[k], alpha[j])>0)

k=j;

if (k!=i)

{

strcpy(temp,alpha[i]);

strcpy(alpha[i],alpha[k]);

strcpy(alpha[k],temp);

}

}

}

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

}

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

/\*删除升序数组中的重复元素（指针版）

编写程序，删除一维数组中所有相同的数，只保留一个，数组中的数已按由小到大的顺序排列，

int unique(int \*pArr,int num)

pArr和num分别为数组首地址和数组元素的个数，函数返回删除重复元素之后数组中元素个数。

如：in.txt

2 2 2 3 4 4 5 6 6 6 6 7 7 8 9 9 10 10 10

则：out.txt

2 3 4 5 6 7 8 9 10

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include <iostream>

using namespace std;

int unique(int \*pArr,int num)

{

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

int unique(int \*pArr,int num)

{

int \*p,count;

p=pArr;

for(int i=0;i<num-1;i++)

{

if (pArr[i+1]!=pArr[i])

{

p++;

\*p=pArr[i+1];

}

}

count=p-pArr+1;

return count;

}

/\*查找字符串

编写一个函数实现：在一个字符串中查找另一个给定的字符串首次出现的位置。

int find(char \* strSrc, char \* strFind)

该函数在字符串strSrc中查找字符串srcFind首次出现的位置，并返回一个整数表示该位置，返回-1表示查找失败或未找到。

如：in.txt

Welcome to the C++ world!

C++

WelcomeC++ to the C++ world!

C++

C++Welcome to the world!

C++

Welcome to the worldC++

C++

Welcome to the worldC+

C++

则:out.txt

16

8

1

21

-1

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include<iostream>

#include <cstring>

using namespace std;

int find(char \* strSrc, char \* strFind)

{

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

int find(char \* strSrc, char \* strFind)

{

int n1=strlen(strSrc);//

int n2=strlen(strFind);//

if(n1<n2) return -1;

int i,j;

for(i=0;i<=n1-n2;i++)//查找字符串strFind在strSrc开始位置

{

for(j=0;j<n2;j++)

{

if(\*(strSrc+i+j) != \*(strFind+j))

break;

}

if(j==n2) return i+1;

}

return -1;

}

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

}

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

char strSrc[100],strFind[100];

while(cin.getline(strSrc,100)&&cin.getline(strFind,100))

cout<<find(strSrc,strFind)<<"\n";

file\_in.close();

file\_out.close();

cin.rdbuf(std\_in);

cout.rdbuf(std\_out);

return 0;

}

/\*编写函数fun实现将字符串按字符降序排列。

如：in.txt

CEAedcab

则:out.txt

edcbaECA

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include<fstream>

#include<cstring>

using namespace std;

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

void fun(char \*s)

{

int n = strlen(s) ;

char \*p = s,temp;

int i,j;

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

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

{

if(p[j]<p[j+1])

{

temp = p[j];

p[j] = p[j+1];

p[j+1] = temp;

}

}

}

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

int main()

{

ifstream infile("in.txt",ios::in); //定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//定义输出文件流类对象outfile

//infile、outfile与cin、cout用法类似

char s[100];

while(infile>>s)

{

fun(s);

outfile<<s<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*编写函数fun实现两个整数的交换。

如:in.txt

60 65

则:out.txt

a=65 b=60

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include<fstream>

using namespace std;

void fun(int \*a,int \*b)

{

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

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

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

}

int main()

{

ifstream infile("in.txt",ios::in); //创建输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//创建输出文件流类对象outfile

//infile和outfile用法与cin和cout一样

int a,b;

while(infile>>a>>b)

{

fun(&a,&b);

outfile<<"a="<<a<<' '<<"b="<<b<<'\n';

}

infile.close(); //关闭文件in.txt

outfile.close(); //关闭文件out.txt

return 0;

/\*请编写一个函数fun,将一个数字字符串转换为一个整数。

如：in.txt

-1234

则:out.txt

-1234

注意:请勿改动现有任何内容，仅在Begin和End之间补充代码。

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

#include <fstream>

using namespace std;

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

long fun ( char \*p)

{

char \*q = p;

long s = 0;

long flag = 1;

if(\*q == '-')

{

flag = -1;

q++;

}

while(\*q!=0)

{

s = s\*10+(\*q-‘0’);

q++;

}

return s\*flag;

}

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

int main()

{

ifstream infile("in.txt",ios::in); //定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out); //定义输出文件流类对象outfile

//infile、outfile与cin、cout用法相同

char s[10];

while(infile>>s)

outfile<<fun(s)<<endl;

infile.close();

outfile.close();

return 0;

}

/\*字符串的截取:编写一个函数char \* substr(char \*s, int begin, int end，char \*s1)，

用于取得字符串s中从begin位置开始至end位置结束（包括begin和end位置上的字符，位置从0开始计算）的子字符串，将该子字符串存放在s1中。

如：in.txt

This is a wonderful world!

10 18

则:out.txt

wonderful

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include <iostream>

using namespace std;

void substr(char \*s, int begin, int end,char \*s1)

{

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

{

int i;

s+=begin;

for(i=0; i<=end-begin; i++)

s1[i]=s[i];

s1[i]='\0';

}

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

}

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

char s[100];

char s1[100];

int m,n;

cin.getline(s,100);

cin>>m>>n;

substr(s, m, n,s1);

cout<<s1<<endl;

file\_in.close();

file\_out.close();

cin.rdbuf(std\_in);

cout.rdbuf(std\_out);

return 0;

}

/\*编写函数fun，将M行N列的二维数组a中的数据按列的顺序依次放到一维数组b中。

其中a为指向二维数组的指针， b指向一维数组的指针，m和n分别是二维数组a

的行数和列数,函数返回值为一维数组b存放数据的个数。

如:in.txt

33 33 33 33

44 44 44 44

55 55 55 55

则:out.txt

33 44 55 33 44 55 33 44 55 33 44 55

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

using namespace std;

int fun(int (\*a)[10], int \*b, int m,int n)

{

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

int count = 0;

int i,j;

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

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

{

\*(b+count) = \*(\*(a+i)+j);

count++;

}

return count;

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

}

int main()

{

ifstream infile("in.txt",ios::in); //定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//定义输出文件流类对象outfile

//infile与cin用法相同，outfile与cout用法相同

int i, j, k ;

int w[10][10], a[100], n = 0, mm, nn ;

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

{

infile>>mm>>nn;

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

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

infile>>w[i][j] ;

n=fun(w,a,mm,nn) ;

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

outfile<<a[i]<<' ';

outfile<<'\n';

}

infile.close();

outfile.close();

return 0;

}

/\* 请编写函数fun将字符串中的前导\*号全部移到字符串的尾部。

假定输入的字符串中只包含字母和\*号。

如：in.txt

\*\*\*\*\*\*\*A\*BC\*DEF\*G\*\*\*\*

则:out.txt

A\*BC\*DEF\*G\*\*\*\*\*\*\*\*\*\*\*

注意：请勿改动现有内容，仅在Begin和End之间补充代码。

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

#include <fstream>

using namespace std;

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

void fun(char \*s)

{

char \*p=s,\*q;

while(\*p=='\*')

{

q = p;

while(\*q!=0)

{

\*q = \*(q+1);

q++;

}

\*(q-1) = '\*';

}

}

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

int main()

{

ifstream infile("in.txt",ios::in); //定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//定义输出文件流类对象outfile

//infile、outfile用法和cin、cout一样

char s[81];

while(infile>>s)

{

fun(s);

outfile<<s<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*编写函数fun找出一维整型数组元素中最大的值和它所在的下标，最大的值和它所在的下标通过形参传回。

如：in.txt

1 2 13 14 23 15 21 60 35 23

则:out.txt

max=60，index=7

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

using namespace std;

void fun(int a[], int n , int \*max, int \*d )//n为数组a存放的数组个数

{

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

int i=0;

\*max = a[i];

\*d = i;

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

if(a[i]>\*max)

{

\*max = a[i];

\*d = i;

}

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

}

int main()

{

ifstream infile("in.txt",ios::in);//定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//定义输出文件流类对象outfile

//infile与cin用法相同，outfile与cout用法相同。

int a[20],max,index,n=0;

while(infile>>a[n])

n++;

fun(a,n,&max,&index);

outfile<<"Max="<<max<<",Index="<<index<<'\n';

infile.close();

outfile.close();

return 0;

}

/\*编写函数fun：将一个字符串中的大写字母转换成小写，小写转换成大写，其它字符保持不变。

如:in.txt

HappyNew2019Year

则:out.txt

hAPPYnEW2019yEAR

请在Begin和End之间完善程序。注意：不要删掉注释语句。

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

#include <fstream>

using namespace std;

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

void fun(char \*s)

{

char \*p = s;

while(\*p!=0)

{

if(\*p>='A'&&\*p<='Z')

\*p = \*p+32;

else

if(\*p>='a'&&\*p<='z')

\*p = \*p-32;

p++;

}

}

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

int main()

{

ifstream infile("in.txt",ios::in);//定义输入流文件对象infile

ofstream outfile("out.txt",ios::out);//定义输出流文件对象outfile

//infile、outfile的用法同cin、cout

char str[100];

while(infile>>str) //从文件in.txt中读入字符串到数组str中

{

fun(str);

outfile<<str<<endl; //输出字符串到文件out.txt中

}

infile.close();

outfile.close();

return 0;

}

/\* 请编写一个函数fun，用来删除字符串中的所有空格。

如:in.txt

asd af aa z67

则out.txt

asdafaaz67

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

using namespace std;

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

void fun(char \*str)

{

char \*p = str;

char \*q;

while(\*p!=0)

{

if(\*p==' ')

{

q = p;

while(\*q!=0)

{

\*q = \*(q+1);

q++;

}

}

else

p++;

}

}

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

int main()

{

ifstream infile("in.txt",ios::in);//定义输入文件流类对象infile

ofstream outfile("out.txt",ios::out);//定义输出文件流类对象outfile

//infile、outfile用法与cin、cout一样

char str[81];

while(!infile.eof())

{

infile.getline(str,81);//读入一行字符串(可以有空格)

fun(str);

outfile<<str<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*字符串的选择排序

使用选择排序法实现：将若干（5个）字符串（长度不超过30）按字母顺序（由小到大）输出。编写函数实现排序过程：

void selectSort(char (\*alpha)[30], int n)

其中alpha存放5个字符串的首地址（行指针），n表示字符串的个数5

输入5个字符串（长度不超过30），每个字符串占一行，以回车结束。输出从小到大排序后的字符串，每个字符串占一行，以回车结束。

如：in.txt

Follow me

Basic

Great Wall

FORTRAN

Computer design

则:out.txt

Basic

Computer design

FORTRAN

Follow me

Great Wall

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include <iostream>

#include <cstring>

using namespace std;

const int N=5;

const int M=30;

void selectSort(char (\*alpha)[M], int n)

{

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

void selectSort(char (\*alpha)[M], int n)

{

int i, j, k;

char temp[M];

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

{ k=i;

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

if(strcmp(alpha[k], alpha[j])>0)

k=j;

if (k!=i)

{

strcpy(temp,alpha[i]);

strcpy(alpha[i],alpha[k]);

strcpy(alpha[k],temp);

}

}

}

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

}

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

char alpha[N][M];

int i;

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

cin.getline(alpha[i],M);

selectSort (alpha, N);

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

cout<<alpha[i]<<endl;

file\_in.close();

file\_out.close();

cin.rdbuf(std\_in);

cout.rdbuf(std\_out);

return 0;

}

/\* 回文字符串（指针版）:

写程序，输入多个字符串，判断输入的字符串是否"回文"，若是回文，输出YES，若不是回文输出NO.

每个输入的字符串的前端或后端可能包含多个空格，不算作字符串的一部分，不参与判断过程，应该先裁剪掉。

字符串的内部包含的空格，算作字符串的一部分，要参与判断的过程。

每个字符串的输入长度（包括所有的空格）不超过100，使用指针pi和pj完成计算过程，定义如下：

char s[100], \*pi, \*pj;

输入包括多个用例，每个用例包含一个字符串（前端或后端或内部可能包含多个空格，字符串的长度不超过100），以换行作为字符串的结束。

判断输入的字符串是否"回文"，若是回文，输出YES，若不是回文输出NO，以换行结束。

如：in.txt

abc cba

abc 12321 cba

abc 1221 cba

abc 123321 cba

abc 123321 cba

则:out.txt

YES

YES

YES

YES

NO

注意：请勿改动主函数main和其它函数中的任何内容，仅在Begin和End之间填入你编写的若干语句

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

#include <fstream>

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

ifstream file\_in("in.txt");

ofstream file\_out("out.txt");

if(!file\_in||!file\_out) return -1;

streambuf \*std\_in;

streambuf \*std\_out;

std\_in= cin.rdbuf(file\_in.rdbuf());

std\_out= cout.rdbuf(file\_out.rdbuf());

char s[100], \*pi, \*pj;

int n;

while(cin.getline(s,100))

{

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

法一：int i=0,l=0,q=0;

while(\*s==' '){int p=0; while(\*(s+p)) \*(s+p)=\*(s+p+1),p++;}

n=strlen(s);

pi=s;

int j=n;

pj=&s[n-1];

while(\*(pi+i)&&\*(pi+i)==\*pj)

{i++;pj--;

if(i>=n/2){file\_out<<"YES\n";l=1;break;}

}

if(l!=1)file\_out<<"NO\n";

法二： n=strlen(s);

pi=s; pj=s+n-1;

while(\*pi==' ')

{ pi++;}

while(\*pj==' ')

{pj--;}

while( (pi<pj) &&(\*pi==\*pj) )

{ pi++; pj--; }

if(pi<pj)

cout<<"NO"<<endl;

else

cout<<"YES"<<endl;

法三：

int a=0,x=1,n,b;

pi=&s[0];

tjt:

if(\*pi==' ')

{ a++;

pi++;

goto tjt;

}

n=strlen(s);

n=n-a;

for(b=0;b<n/2;b++)

{

if(\*pi!=\*(pi+n-1-2\*b))

x=0;

pi++;

}

if(x==1)

file\_out<<"YES"<<endl;

else file\_out<<"NO"<<endl;

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

}

file\_in.close();

file\_out.close();

cin.rdbuf(std\_in);

cout.rdbuf(std\_out);

return 0;

}

**选择题（指针）**

1. 设p指向的字符串为"string"，则cout<<p+3的输出结果为（ ing ）
2. 假定有如下定义语句char \*p="test"; 则\*(p++)的值为（ t ）
3. 若已定义：int a[]={0,1,2,3,4,5,6,7,8,9}, \*P = a, i;其中0 ≤ i ≤9，则对a数组元素不正确的引用是 ( a[10] )
4. 若定义int a=8, b, \*p = &b; 则下面对赋值语句p=&a; \*p = a; 的正确解释为（ D ）
5. 设有"int a[]={10,11,12}, \*p=&a[0];"，则执行完"\*p++;\*p+=1;"后a[0], a[1], a[2]的值依次是（ 10，12，12 ）
6. 假定一条定义语句为"int a[10], x, \*pa=a;"，若要把数组a中下标为3的元素值赋给x，

则不正确的语句为（ x=\*pa+3; ）

1. 下列定义中，（ int \*p[ ]; ）是定义指向数组的指针p
2. 如果有语句int p[5]={2,4,6,8};cout<<&p[3]<<endl; 则输出为（ 某个地址 ）
3. 若有int k=1,a[5]={6,22,36,47};int \*p1=a,\*p2=&a[2];则下列不能正确执行的语句是（ p1=p1+p2; ）
4. 有如下定义：int a[5] = {1,3,5,7,9}, \*p = a;下列表达式中不能得到数值5的是（ a[3] ）

**选择题较难（指针）**

1. 若有如下定义语句int a[2][3]={{3,7},{9,6,8}}; int (\*p)[3]=a; 则\*(p[1]+2)的值为（ 8 ）
2. 如果有语句int a=12,b=45,\*pa=&a,\*pb=&b;pa=pb; cout<<\*pa<<endl; 则输出为（ 45 ）
3. 若有如下定义语句char \*p="程序设计"; 则语句p++使指针p的偏移（ 1 ）个字节。
4. 如果有语句int a=3, \*p=&a; double b=4; p=&b; cout<<\*p ; 则输出为（ 4 ）
5. 若有int \*p[3];，则下列表述中正确的是（ p是一个有3个元素的指针数组，每个元素是一个int型的指针 ）
6. 如果有语句int a=21, b=41, \*pa=&a, \*pb=&b, \*t; \*t=\*pa; \*pa=\*pb; \*pb=\*t; 则\*pa的值为（ 41 ）

**8-10章练习**

1./\*

定义函数fun：根据p指针，计算该学生的4门课的平均分。

学生记录由学号，姓名，四门课成绩，平均分组成。

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;

struct student

{

int id;

char name[20];

int score[4];

int avg;

};

void fun(student\* p)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int sum=0;

int i;

for(i=0;i<4;i++)

{

sum=sum+p->score[i];

}

p->avg=sum/4;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

student stu;

int i;

ifstream ifile("in.txt");

ofstream ofile("out.txt");

while(!ifile.eof())

{

ifile>>stu.id; //从文件读入学号

ifile>>stu.name; //从文件读入姓名

for(i=0;i<4;i++)

ifile>>stu.score[i]; //从文件读入四门课的成绩

fun(&stu);

cout<<stu.id<<" "<<stu.name<<" ";

for(i=0;i<4;i++)

cout<<stu.score[i]<<" ";

cout<<"平均成绩:"<<stu.avg<<endl;

//输出到out.txt文件

ofile<<stu.id<<" "<<stu.name<<" ";

for(i=0;i<4;i++)

ofile<<stu.score[i]<<" ";

ofile<<"平均成绩:"<<stu.avg<<endl;

}

ifile.close(); //关闭输入文件in.txt

ofile.close(); //关闭输出文件out.txt

return 0;

}

2./\*

定义函数fun：在n个学生记录中，根据学生的学号，查找该学生对应的成绩，若没有该学号，则返回-1。

学生的记录由学号和成绩组成，N名学生的数据已在主函数中放入结构体数组s中，要查找的学生学号从文件读入，依次将学生的学号和查找结果存放在输出文件中。

（用于字符串比较的函数是strcmp）。

例如in.txt中显示：

GA09

GA010

GA011

则out.txt中输出：

GA09 -1

GA010 -1

GA011 91

注意：请勿改动主函数main和其它函数中的任何内容，只能在begin与end之间写代码。

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

#include<iostream>

#include<cstring>

#include<fstream>

using namespace std;

#define N 16

struct student

{ char num[10];

int score;

} ;

int fun (student stu[],int n, char num[])

{

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

int i=0;

bool flag=false;//找到标志

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

{

if(strcmp(stu[i].num,num)==0) //stu[i].num代表数组stu中每一项的字符 ，num为用户想要保存的学号（字符名）

{

flag=true;

break;

}

}

if(flag==true)

return stu[i].score;

else

return -1;

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

}

int main()

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

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

student s[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87},{"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},{"GA011",91},{"GA017",64},{"GA018",64},{"GA016",72}};

int score;

char num[10];//保存要查找学号

while(infile>>num)

{

score=fun(s,N,num);

outfile<<num<<score<<endl;

}

infile.close();

outfile.close();

return 0;

}

3.

/\*

声明一个三角形类Triangle，包括三个私有对象成员point1,point2,point3，这三个成员是点类Point的对象；

类Point包括公有成员x(float),y(float)表示点的坐标，包括构造函数给坐标初始化；

类Triangle包括构造函数，求面积的函数float Area()，包括依次输出三个点point1,point2,point3的坐标的函数Disp()；

程序中的输入文件in.txt包含以下数据：

1 1 2 2 3 3

0 0 0 4 4 4

1 1 4 4 1 4

输出文件out.txt包含以下数据（点的x,y坐标之间以一个空格分隔）：

1 1

2 2

3 3

0

0 0

0 4

4 4

8

1 1

4 4

1 4

4.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include<fstream>

#include<cmath>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Point

{

public: float x,y;

Point(float x1,float y1) {x=x1;y=y1;}

};

class Triangle

{

private: Point point1,point2,point3;

public:

Triangle(Point &p1,Point &p2,Point &p3):point1(p1),point2(p2),point3(p3){}

float Area()

{

float s1 = sqrt((point1.x-point2.x)\*(point1.x-point2.x)+(point1.y-point2.y)\*(point1.y-point2.y));

float s2 = sqrt((point3.x-point2.x)\*(point3.x-point2.x)+(point3.y-point2.y)\*(point3.y-point2.y));

float s3 = sqrt((point3.x-point1.x)\*(point3.x-point1.x)+(point3.y-point1.y)\*(point3.y-point1.y));

float s = 0.5\*(s1+s2+s3);

return sqrt(s\*(s-s1)\*(s-s2)\*(s-s3));

}

double Disp(ostream & output)

{

output<<point1.x<<' '<<point1.y<<endl;

output<<point2.x<<' '<<point2.y<<endl;

output<<point3.x<<' '<<point3.y<<endl;

}

};

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

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

float x1,y1,x2,y2,x3,y3;

while(!infile.eof())

{

infile>>x1>>y1>>x2>>y2>>x3>>y3;

Point p1(x1,y1),p2(x2,y2),p3(x3,y3);

Triangle t(p1,p2,p3);

t.Disp(outfile);

outfile<<t.Area()<<endl;

}

infile.close();

outfile.close();

return 0;

}

正常做法：

/\*

声明一个三角形类Triangle，包括三个私有对象成员point1,point2,point3，这三个成员是点类Point的对象；

类Point包括公有成员x(float),y(float)表示点的坐标，包括构造函数给坐标初始化；

类Triangle包括构造函数，求面积的函数float Area()，包括依次输出三个点point1,point2,point3的坐标的函数Disp()；

程序中的输入文件in.txt包含以下数据：

1 1 2 2 3 3

0 0 0 4 4 4

1 1 4 4 1 4

输出文件out.txt包含以下数据（点的x,y坐标之间以一个空格分隔）：

1 1

2 2

3 3

0

0 0

0 4

4 4

8

1 1

4 4

1 4

4.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include<fstream>

#include<cmath>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Point

{

public: float x,y;

Point(float x1,float y1) {x=x1;y=y1;}

};

class Triangle

{

private: Point point1,point2,point3;

public:

Triangle(Point &p1,Point &p2,Point &p3):point1(p1),point2(p2),point3(p3){}

float Area()

{

float s1 = sqrt((point1.x-point2.x)\*(point1.x-point2.x)+(point1.y-point2.y)\*(point1.y-point2.y));

float s2 = sqrt((point3.x-point2.x)\*(point3.x-point2.x)+(point3.y-point2.y)\*(point3.y-point2.y));

float s3 = sqrt((point3.x-point1.x)\*(point3.x-point1.x)+(point3.y-point1.y)\*(point3.y-point1.y));

float s = 0.5\*(s1+s2+s3);

return sqrt(s\*(s-s1)\*(s-s2)\*(s-s3));

}

void Disp()

{

outfile<<point1.x<<' '<<point1.y<<endl;

outfile<<point2.x<<' '<<point2.y<<endl;

outfile<<point3.x<<' '<<point3.y<<endl;

}

};

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

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

float x1,y1,x2,y2,x3,y3;

while(!infile.eof())

{

infile>>x1>>y1>>x2>>y2>>x3>>y3;

Point p1(x1,y1),p2(x2,y2),p3(x3,y3);

Triangle t(p1,p2,p3);

t.Disp();

outfile<<t.Area()<<endl;

}

infile.close();

outfile.close();

return 0;

}

4./\*

定义时间类Time，包括私有成员变量hour(int)、minute(int)、second(int)；

成员函数set\_time(int h,int m,int s)将读入的3个整数规范化后给3个成员变量赋值，使得

second满60向minute进1，并规范化至[0,59]；minute满60向hour进1，并规范化至[0,59]；hour满24则规范化至[0,23]；

成员函数show\_time(ofstream & output)读取3个成员变量，格式化为00:00:00形式后，输出到文件中；

程序中的输入文件in.txt包含以下数据：

0 0 0

1 59 62

1 63 63

2 63 59

3 123 63

3 126 125

23 59 67

26 68 61

输出文件out.txt包含以下数据（英文冒号）：

00:00:00

02:00:02

02:04:03

03:03:59

05:04:03

05:08:05

00:00:07

03:09:01

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include <fstream>

#include <iomanip>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Time

{public:

void set\_time(int h,int m,int s);

void show\_time(ofstream & output);

private:

int hour;

int minute;

int second;

};

void Time::set\_time(int h,int m,int s)

{

second = s % 60;

minute = m + s/60;

hour = h + minute /60;

hour = hour %24;

minute = minute % 60;

}

void Time::show\_time(ofstream & output)

{

output<<setfill('0');

output<<setw(2)<<hour<<":"<<setw(2)<<minute<<":"<<setw(2)<<second<<endl;

}

法二：

class Time

{ private:

int hour;

int minute;

int second;

public:

int set\_time(int h,int m,int s)

{ while(s>0&&s/60>=1)

{ m=m+1;

s=s-60;

}

while(m>0&&m/60>=1)

{ h=h+1;

m=m-60;

}

while(h>=24)

{ h=h-24; }

hour=h;

minute=m;

second=s;

}

void show\_time(ofstream & output)

{

output<<setfill('0');

output<<setw(2)<<hour<<":"<<setw(2)<<minute<<":"<<setw(2)<<second<<endl;

}

};

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

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

Time t;

int h,m,s;

while(!infile.eof())

{

infile>>h>>m>>s;

t.set\_time(h,m,s);

t.show\_time(outfile);

}

infile.close();

outfile.close();

return 0;

}

5.

/\*

设计抽象基类Shape,包括求面积的成员函数double area();

由Shape派生出3个派生类：Circle(圆形)、Rectangle(矩形)、Triangle(三角形)

Circle包括半径radius，Rectangle包括长width和宽height，Triangle包括底width和高height；

分别设计Circle(圆形)、Rectangle(矩形)、Triangle(三角形)的构造函数和求面积函数area()，

使得printArea() 函数能够正确输出面积值；

程序中的输入文件in.txt包含以下数据：

2.0

3.5 2.0

3.5 2.0

输出文件out.txt包含以下数据：

12.5664

7

3.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

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

class Shape

{public:

virtual double area() const =0;

};

class Circle:public Shape

{public:

Circle(double r):radius(r){}

virtual double area() const {return 3.14159\*radius\*radius;};

protected:

double radius;

};

class Rectangle:public Shape

{public:

Rectangle(double w,double h):width(w),height(h){}

virtual double area() const {return width\*height;}

protected:

double width,height;

};

class Triangle:public Shape

{public:

Triangle(double w,double h):width(w),height(h){}

virtual double area() const {return 0.5\*width\*height;}

protected:

double width,height;

};

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

void printArea(const Shape &s)

{outfile<<s.area()<<endl;}

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

double r,w,h;

infile>>r;

Circle circle(r);

printArea(circle);

infile>>w>>h;

Rectangle rectangle(w,h);

printArea(rectangle);

infile>>w>>h;

Triangle triangle(w,h);

printArea(triangle);

infile.close();

outfile.close();

return 0;

}

7.

/\*

定义电影类Movie，包括私有对象成员电影名name(string)，开始时间startTime和结束时间endTime；

startTime和endTime是Time类的对象；Time类包括公有成员hour(int)，minute(int)和构造函数；

Movie类包括构造函数，求电影时长(多少分钟)的函数 int Duration()，显示电影播放信息的函数Display();

程序中的输入文件in.txt包含以下数据:

The Fast and the Furious 8

12 30 14 50

Smurfs: The Lost Village

13 9 15 45

Battle of Memories

15 20 16 54

输出文件out.txt包含以下数据（英文冒号）：

The Fast and the Furious 8

start time:12:30

140minutes

Smurfs: The Lost Village

start time:13:9

156minutes

Battle of Memories

start time:15:20

94minutes

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Time

{

public:

int hour,minute;

Time (int h,int m)

{hour = h; minute= m;}

};

class Movie

{

private:

string name;

Time startTime,endTime;

public:

Movie(string str,Time &stime,Time &etime):startTime(stime),endTime(etime){name=str;}

int Duration()

{

int d;

d = (endTime.hour - startTime.hour)\*60+(endTime.minute-startTime.minute);

return d;

}

void Display(ofstream &output)

{

output<<name<<'\n'<<"start time:"<<startTime.hour<<':'<<startTime.minute<<'\n';

}

};

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

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

string str;

int hour1,minute1,hour2,minute2;

while(!infile.eof())

{

getline(infile,str);

infile>>hour1>>minute1>>hour2>>minute2;

infile.ignore();

Time startT(hour1,minute1),endT(hour2,minute2);

Movie mov(str,startT,endT);

mov.Display(outfile);

outfile<<mov.Duration()<<"minutes"<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*

定义箱子类Box，包括私有成员变量高度height(int)、宽度width(int)、长度length(int)，

包括构造函数和成员函数volume()；

构造函数要能够实现main()函数中对象box1,box2,box3,box4的正确构造；

成员函数volume(),求箱子的体积:height\*width\*length

程序中的输入文件in.txt包含以下数据:

(共2组数据，第1组中第1个箱子的height\*width\*length为5\*3\*2，第3个箱子的height\*width为2\*3)

5 3 2

2 3

6 3 5

5 3

输出文件out.txt包含以下数据:

30

30

60

60

90

90

150

150

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Box

{

private:

int height;

int width;

int length;

public:

Box(int h=10,int w=10,int len=10)

{height=h;

width=w;

length=len;

}

int volume( )

{return(height\*width\*length); }

};

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

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

int n1,n2,n3;

while(!infile.eof())

{

infile>>n1>>n2>>n3;

Box box1(n1,n2,n3),box2;

box2=box1;

infile>>n1>>n2;

Box box3(n1,n2);

Box box4(box3);

outfile<<box1.volume()<<'\n'<<box2.volume()<<'\n'

<<box3.volume()<<'\n'<<box4.volume()<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*

定义学生类Student，包括私有成员变量学号(int)、性别(char)、年龄(int)，包括构造函数，用于对私有成员初始化；

成员函数Display(ostream &output)用于输出：学号(int)、性别(char)、年龄(int)，将这些数据输出到输出流；

程序中的输入文件in.txt包含以下数据（3列数据分别表示学号、性别、年龄） ：

1001 f 18

1002 m 19

1003 m 17

1004 f 18

1005 f 19

输出文件out.txt包含以下数据(数据之间用一个空格隔开）：

1001 f 18

1002 m 19

1003 m 17

1004 f 18

1005 f 19

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Student

{

private:

int num;

char sex;

int age;

public:

Student(int n,char s,int a)

{

num=n; sex=s; age=a;

}

void Display(ostream &output)

{

output<<num<<' '<<sex<<' '<<age<<endl;

}

};

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

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

int n;char s;int a;

while(!infile.eof())

{

infile>>n>>s>>a;

Student s1(n,s,a);

s1.Display(outfile);

}

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据

6S

6Plus

7S

7Plus

定义类Apple，包括私有成员变量name(string)，包括构造函数和析构函数,使得输出文件out.txt包含以下数据

Apple6S

~Apple6S

Apple6Plus

~Apple6Plus

Apple7S

~Apple7S

Apple7Plus

~Apple7Plus

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include<string>

#include<fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Apple

{public:

Apple(string str)

{name=str;

outfile<<"Apple"<<name<<endl;}

~Apple( )

{outfile<<"~Apple"<<name<<endl;}

private:

string name;

};

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

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

string name;

while(infile>>name)

{

Apple \*pA = new Apple(name);

delete pA;

}

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据：

1001

1002

1003

定义类Student，包括私有成员变量学号(int)，包括构造函数和析构函数,使得输出文件out.txt包含以下数据：

Constructor1001

Constructor1002

Constructor1003

Destructor1003

Destructor1002

Destructor1001

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Student

{public:

Student(int n)

{num=n;

outfile<<"Constructor"<<num<<endl;}

~Student()

{outfile<<"Destructor"<<num<<endl;}

private:

int num;

};

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

void CreateObjects()

{

int n[3];

infile>>n[1]>>n[2]>>n[3];

Student s1[3]={n[1],n[2],n[3]};

}

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

CreateObjects();

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据:

Huawei

Sumsung

iPhone7

定义类Phone，包括私有成员变量name(string)，包括构造函数，成员函数setName()修改name,

常成员函数getName()获得name，使得输出文件out.txt包含以下数据;

Huawei

Sumsung

iPhone7

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

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

#include<string>

#include<fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

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

class Phone

{public:

Phone(){}

Phone(string str) {name=str;}

void setName(string str) {name = str;}

string getName() const {return name;}

private:

string name;

};

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

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

Phone ph1,ph2;

string name;

infile>>name;

ph1.setName(name);

infile>>name;

ph2.setName(name);

Phone const \*pPhone = &ph1;

outfile<<pPhone->getName()<<endl;

pPhone = &ph2;

outfile<<pPhone->getName()<<endl;

infile>>name;

const Phone ph3(name);

outfile<<ph3.getName()<<endl;

infile.close();

outfile.close();

return 0;

}

设计函数fun:从保存n个学生记录的stu1数组中，找出成绩在60和80分之间（包括60和80）的学生数据，并保存在sut2数组中，函数返回值表示找到的学生人数。

其中，学生记录由学号和成绩组成。

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

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

#include <iostream>

using namespace std;

#include<fstream>

#include<iomanip>

#define N 16

struct student

{

char num[10];

int score;

};

int fun(student stu1[],int n,student stu2[])

{

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

int i,j=0;

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

if (stu1[i].score>=60&&stu1[i].score<=80)

{

stu2[j]=stu1[i];

j++;

}

return j;

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

}

int main()

{

student stu1[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87},{"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},{"GA011",91},{"GA017",64},{"GA018",64},{"GA016",72}};

student stu2[N];

int i,n;

n=fun(stu1,N,stu2);

cout<<"所有的学生数据："<<endl;

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

cout<<stu1[i].num<<setw(4)<<stu1[i].score<<endl;

cout<<"60-80分之间的学生数据："<<endl;

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

cout<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

//结果写入out.txt

ofstream ofile;

ofile.open("out.txt");

ofile<<n<<endl;

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

ofile<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

ofile.close();

}

设计函数fun:从保存n个学生记录的stu1数组中，找出低于平均分的学生数据，并保存在sut2数组中，函数返回值表示找到的学生人数。

学生的记录由学号和成绩组成，N名学生的数据已在主函数中放入结构体数组中。

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

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

#include <iostream>

using namespace std;

#include<fstream>

#include<iomanip>

#define N 8

struct student

{

char num[10];

int score;

};

int fun(student stu1[],int n,student stu2[])

{

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

int i,j=0,sum=0;

float avg=0;

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

{

sum=sum+stu1[i].score;

}

avg=sum/n;

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

if (stu1[i].score<avg)

{

stu2[j]=stu1[i];

j++;

}

return j;

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

}

int main()

{

student stu1[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87}};

student stu2[N];

int i,n;

n=fun(stu1,N,stu2);

cout<<"所有学生数据："<<endl;

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

cout<<stu1[i].num<<setw(4)<<stu1[i].score<<endl;

cout<<n<<endl;

cout<<"低于平均分学生数据："<<endl;

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

cout<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

//结果写入out.txt

ofstream ofile;

ofile.open("out.txt");

ofile<<n<<endl;

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

ofile<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

ofile.close();

}

一．选择题

1.设有以下程序段，则表达式的值不为100的是\_\_\_\_D. (++p)->b\_\_\_。

struct st

{

int a,\*b;

};

int main()

{

int m1[] ={10,100}, m2[]={100,200};

st \*p, x[]={99,m1,100,m2};

p=x;

...

}

2.若有以下定义和语句：

struct student

{

int num,age;

};

student stu[3]={{1001,20},{1002,19},{1003,21}};

student \*p=stu;

则以下结果不是1002的是\_\_D. (++stu)->num\_\_\_\_\_。

3.设有以下的结构体说明和变量定义，如下图所示，指针p指向变量one，指针q指向变量two，则不能将结点连接到结点one之后的语句是\_\_\_\_A. p.next=&two;\_\_\_。

struct node

{

int n;

struct node \*next;

}one, two, \*p=&one, \*q=&two;



4.有以下说明和定义语句:

struct student

{ int age;

char num[8];

} ;

struct student stu[3] = {{20,"20041"}, {21,"20042"},{19,"20043"}};

struct student \*p = stu;

以下选项中引用结构体变量成员的表达式错误的是( C.stu[3].age )

.

5.根据以下定义，能输出字母M的语句是 B.cout<<class[2].name[0]<<endl; 。

struct person

{

char name[9];

Int age;

};

person class[4]={{"John",17},{"Paul",19},{"Mary",18},{"Adam",16}};

6.以下程序的运行结果是（“1”）。

#include<iostream>

using namespace std;

class Test

{

public:

Test(){}

Test(Test&t)

{

cout<<1;

}

};

Test fun(Test&u)

{

Test t=u;

return t;

}

int main()

{

Test x,y;

x=fun(y);

return 0;

7.构造函数是在( 创建对象)时被执行的.

8.下列关于类和对象的描述中不正确的是( 一个类只能有一个对象 ).

9.通常，复制构造函数的参数是( 某个对象的引用名 ).

1. C++语言结构体类型变量在程序执行期间（ C.所有成员一直驻留在内存中 ）。

以下程序运行后的输出结果是 C.2041 2044 。

struct STU

{ char name[10]; int num; };

void f1(struct STU c)

{ struct STU b = {"LiSiGuo", 2042};

c = b;

}

void f2(struct STU \*c)

{ struct STU b = {"SunDan", 2044};

\*c = b;

}

int main()

{ struct STU a = {"YangSan", 2041}, b = {"WangYin", 2043};

f1(a); f2(&b);

cout<<a.num<<' '<<b.num<<endl;

return 0;

}

12.static struct {int a1;float a2;char a3;}a[10]={1,3.5,'A'}; 说明数组a是地址常量，它有10个结构体型的下标变量，采用静态存

储方式，其中被初始化的下标变量是（ B.a[0] ）。

13.以下程序的输出结果是 6 。

int main()

{ struct cmplx

{ int x;；

int y；

}cnum[2] = {1, 3, 2, 7};

cout<<cnum[0].y / cnum[0].x \*cnum[1].x<<endl;

}

13设有以下程序段，则表达式的值不为100的是\_\_(++p)->b\_\_\_\_\_。

.struct st

{

int a,\*b;

};

int main()

{

int m1[] ={10,100}, m2[]={100,200};

st \*p, x[]={99,m1,100,m2};

p=x;

...

}

14.设有以下语句

typedef struct S

{

int g;

char h;

} T;

则下面叙述中正确的是 可以用T定义结构体变量 。



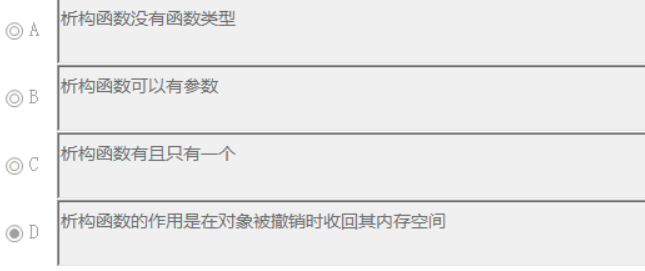
15.当定义一个结构体变量时，系统分配给它的内存是（各成员所需内存量的总和 ）。

16.下列有关类和对象的说法中，正确的是（ 类与对象的关系和数据类型与变量的关系相似 ）。

17.下列关于类和对象的描述中不正确的是( 一个类只能有一个对象 ).

18.假定C为一个类，则执行"C a1(20), a2[5];"语句时，自动调用该类构造函数的次数为 ( 6 ).

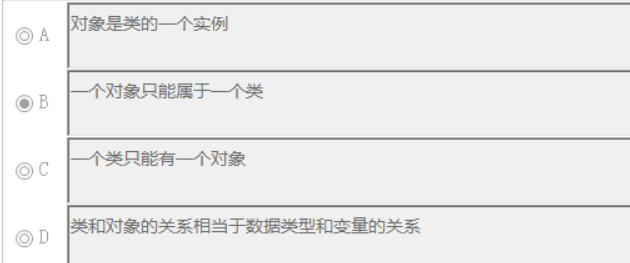
19.下列关于析构函数的描述中不正确的是( B.析构函数可以有参数 ).



20.类成员的访问权限不包括（**A**）



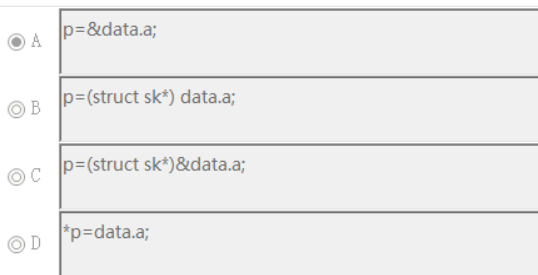
21：下列错误的是（C）



1. 相同结构体类型的变量之间，可以（ 赋值 ）。

23.设有如下定义： struct sk {int a; float b;} data,\*p;

若要使p指向data中的a域，正确的赋值语句是（ C.p=(struct sk\*)&data.a; ）。



24.设有结构体变量声明struct date {int year,month,day;}today;

则sizeof(struct date)和sizeof(today)的值分别为 ( D.12，12 )。

