**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 ）