1.

strcmp(A,B)==0时字符串A与B相等，当A小于B时返回负数，当A大于B时返回正数。

2.

未知数组大小建立数组

#include <iostream>

using namespace std;

int main()

{

int size; //数组大小

cin>>size;

int \*Array=new int[size]; //未知数组大小，所以用指针指向

Array[0]=0;

for(int i=0;i<size;i++) //对数组赋值

{

Array[i]=i\*10;

cout<<Array[i]<<"\t";

if((i+1)%5==0) //每6个数换一行，因为有0，所以是i+1

cout<<endl;

}

cout<<endl<<endl;

delete[] Array; //释放内存

Array=NULL;

return 0;

}

3.

#include <iostream>

#include <string> //一定要包含string头文件（声明string变量）

using namespace std;

#define code "12345"

int main()

{

// char codde;

string codde; //注意为string而不是char

cout<<"Please input:";

cin>>codde;

if(codde==code) //string可以直接用等号对比

cout<<"Success!"<<endl;

else

cout<<"Failed!"<<endl;

return 0;

}

String可以看作是一个字符数组，即可以用Array[i]调用字符串内某个字符，字符串长度可以用Array.length()表示。

4.

用while或do时可以直接用break作为退出选项，其他选项循环可以自行循环不退出。

5.

用switch case作为函数返回值。

例：

int ex(int num,int sta)

{

switch(num)

{

case 0:return sta\*0; //有return直接退出函数，不用break了。

case 1:return sta\*1; //同时让函数的返回值有选择性。

case 2:return sta\*2;

}

}

^6.

应用题分析

//两球队比赛，甲队尾A,B,C,乙队为X,Y,Z。

//A说他不和X比，C说他不和X,Y比。（唯一有用条件）

//求比赛名单。

//A->i;B->j;C->k.

#include "stdio.h"

void main()

{

char i,j,k;

for(i='X';i<='Z';i++)

for(j='X';j<='Z';j++)

if(i!=j) //i,j为两人，不可能相同。（现实限制条件）

for(k='X';k<='Z';k++)

if(i!=k&&j!=k) //不可能为同一人。（现实限制条件）

if(i!='X'&& k!='X'&& k!='Z') //A说他不和X比，C说他不和X,Y比。（题目条件）

printf("A--%c B--%c C--%c\n",i,j,k);

}

7.

#include <stdio.h>

#include <stdlib.h> //srand和rand函数的头文件

#include <time.h> //时间time（）函数头文件

#include <iostream>

using namespace std;

int main()

{

int i;

int result;

srand((int)time(0)); //初始化随机数，利用系统时间产生随机数种子（不是产生随机数）

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

{

result=1+(int)(6.0\*rand()/(RAND\_MAX+1.0));

//生成1-6的随机数 rand()返回值为0到RAND\_MAX

//因为rand的内部实现是用线性同余法做的，他不是真的随机数，只不过是因为其周期特别长，

//所以有一定的范围里可看成是随机的，rand()会返回一随机数值，范围在0至RAND\_MAX 间。

//在调用此函数产生随机数前，必须先利用srand()设好随机数种子，如果未设随机数种子，

//rand()在调用时会自动设随机数种子为1。rand()产生的是假随机数字，每次执行时是相同的。

//若要不同,以不同的值来初始化它.初始化的函数就是srand()。

cout<<result<<endl; //RAND\_MAX的范围最少是在32767之间（int），即双字节（16位数）。

//若用unsigned int 双字节是65535，四字节是4294967295的整数范围。

}

return 0;

}

1. 文件按行读取和按指定宽度读取（方便数组在文件中的定位和操作）

#include <iostream>

#include <fstream> //文件操作函数

#include <string> //getline()所在头文件

using namespace std;

#define sizeA 25 //总数据大小

#define width 5 //每隔width行读取一次数据

#define sizeB sizeA/width //用于输出的数组B

int main()

{

int Pos = 0, j = 0; //Pos为第一个输出数据的行，j为按行输出控制变量

int ArrA[sizeA]; //储存原始数据

string ArrB[sizeB]; //储存整理后数据

////////////////////////////////////////////////////////////////////////////////////////////////////////////

for (int i = 0; i < sizeA; i++) //为数组A（原始数组）赋值

ArrA[i] = i+1;

////////////////////////////////////////////////////////////////////////////////////////////////////////////

ofstream FileA;

FileA.open("FirstFile.txt"); //以文本形式打开文件

if (FileA.is\_open())

{

for (int i = 0; i < sizeA; i++) //将数组A内数据存入文件中

FileA << ArrA[i] << "\n"; //每个数据占一行

cout << "Output Succeed!" << endl;

}

FileA.close();

////////////////////////////////////////////////////////////////////////////////////////////////////////////

cout << "Please input your position:";

cin >> Pos; //输入开始位置

////////////////////////////////////////////////////////////////////////////////////////////////////////////

ifstream FileB;

FileB.open("FirstFile.txt", ios::in); //in以只读形式打开文件

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

{

while (getline(FileB, ArrB[i]) && j < Pos + (i \* width))

j++; //由开始位置Pos开始，每隔width行从文件读取该行数据读入数组B中

}

/\* while (getline(FileB, ArrB[0]) && i < Pos - 1) //按行读取源代码

{

i++;

}

\*/

FileB.close();

////////////////////////////////////////////////////////////////////////////////////////////////////////////

/\* ifstream FileB; //读取全部文件源代码

FileB.open("FirstFile.txt", ios::in);

for (int i = 0; i < 25; i++)

FileB.getline(ArrB[i],sizeof(int));

FileB.close();

\*/

////////////////////////////////////////////////////////////////////////////////////////////////////////////

for (int i = 0; i < sizeB; i++) //输出数组B

{

cout << ArrB[i] << endl;

}

return 0;

}

9.

更多综合应用，详见StuInfo工程。

10.

实现用户名密码登陆功能，并可以将所有信息储存到文件内，即已有用户的数据不会丢失，下次开始还会有原有的数据，随时注册，及时登陆。

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

class StrLibrary

{

public:

string Name;

string Password;

void MENU();

void REGIST(int i); //注册

void LOGIN();

void CHANGE(){}

void CHOISE(int i);

void GETFILE(); //从文件读取信息

};

#define DELAY 600000000

#define L 25 //数据库总大小

static StrLibrary StrLib[L];

static int I = 0; //数据库现有数据个数

void StrLibrary::MENU()

{

StrLibrary o\_Menu;

int CH;

cout << "\n\n1.登陆\t2.注册\n\n3.修改\t0.退出";

cout << "\n\n请输入您要选择的操作:";

cin >> CH;

o\_Menu.GETFILE(); //及时从文件刷新数据（到内存里）

if (I <= 0) //数据库为空，文件为空

{

if (CH != 2 && CH != 0)

{

cout << "\n\nThe Library is empty,Please regist.";

o\_Menu.MENU();

}

}

system("cls");

o\_Menu.CHOISE(CH);

}

void StrLibrary::REGIST(int i)

{

StrLibrary o\_Reg;

string NameP, PasswordP; //暂时使用的string

cout << "\n\nPlease input your infomation.";

cout << "\n\nName:";

cin >> NameP;

StrLib[i].Name = NameP; //将输入的数据储存到内存中，然后储存到文件中

cout << "\nPassword:";

cin >> PasswordP;

StrLib[i].Password = PasswordP;

I++;

ofstream FileA("InfoLibrary.txt");

for (int i = 0; i < I; i++) //将刚刚输入的储存到文件中

{

FileA << StrLib[i].Name << "\n";

FileA << StrLib[i].Password << "\n";

}

FileA.close();

o\_Reg.MENU();

}

void StrLibrary::LOGIN()

{

StrLibrary o\_Log;

string NameQ, PasswordQ; //暂时使用的数据

int Flag = 0;

cout << "\n\nPlease input your infomation.";

cout << "\n\nName:";

cin >> NameQ;

cout << "\nPassword:";

cin >> PasswordQ;

for (int i = 0; i < I; i++) //主要判别代码！！！！！！！！！！！

{

if (StrLib[i].Name == NameQ)

{

Flag = 0;

if (StrLib[i].Password == PasswordQ)

{

cout << "\n\nWelcom!\n\n\n";

Flag = 0;

exit(0);

}

else

{

Flag = 2;

}

}

else

{

Flag = 1;

}

}

if (Flag == 1)

{

system("cls");

cout << "\n\nName Wrong!";

for (int i = 0; i < DELAY; i++){}

system("cls");

o\_Log.MENU();

}

else if (Flag == 2)

{

system("cls");

cout << "\n\nPassword Wrong!";

for (int i = 0; i < DELAY; i++){}

system("cls");

o\_Log.MENU();

}

}

void StrLibrary::CHOISE(int i)

{

StrLibrary o\_Ch;

switch (i)

{

case 1:

{

system("cls");

o\_Ch.GETFILE(); //随时从文件更新，实现刚注册就可以登陆的功能

o\_Ch.LOGIN();

break;

}

case 2:

{

system("cls");

o\_Ch.GETFILE();

o\_Ch.REGIST(I);

break;

}

case 0:

exit(0);

}

}

void StrLibrary::GETFILE()

{

StrLibrary o\_GF;

ifstream FileB;

int Count = 0; string C;

int j = 0, i = 0;

FileB.open("InfoLibrary.txt", ios\_base::in);

while (getline(FileB, C)) //统计行数

Count++;

I = Count / 2;

FileB.close();

FileB.open("InfoLibrary.txt", ios\_base::in);

while (getline(FileB, StrLib[i].Name) && j < 1 + (2 \* i))

{ //从文件导入用户名

j++;

if (j == 1 + (2 \* i))

i++;

if (i == (Count / 2))

break;

}

j = 0;

i = 0;

FileB.close();

FileB.open("InfoLibrary.txt", ios\_base::in);

while (getline(FileB, StrLib[i].Password) && j < 2 + (2 \* i))

{ //导入密码

j++;

if (j == 2 + (2 \* i))

i++;

if (i == (Count / 2))

break;

}

j = 0;

i = 0;

FileB.close();

/\* for (int i = 0; i < Count / 2; i++) //显示库内现有数据

cout <<i<<"Count is" << Count / 2 << "\t" << StrLib[i].Name << "\t" << StrLib[i].Password << endl;\*/

}

int main()

{

StrLibrary Lib;

Lib.MENU();

return 0;

}

11.