/\* \*/内为自己编写的内容。注释/取消注释   
1）注释：组合键“Ctrl+K+C”；   
2）取消注释：组合键“Ctrl+K+U”  
不过是单行注释 如果相要全部注释 可以先按Ctrl+A 再按 Ctrl+K+C

第四章 复合类型

4.1 显示信息

#include<iostream>

const int Asize = 20;

using namespace std;

struct student//定义结构描述

{

char firstname[Asize];

char lastname[Asize];

char grade;

int age;

};

void display(student);//函数原型放在结构描述后

int main()

{

cout << "what is your first name?" << endl;

student lcg;//创建结构变量（结构数据对象）

cin.getline(lcg.firstname, Asize);

cout << "what is your last name?" << endl;

cin.getline(lcg.lastname, Asize);

cout << "what letter grade do you deserve?" << endl;

cin >> lcg.grade;

cout << "what is your age?" << endl;

cin >> lcg.age;

display(lcg);

return 0;

}

void display(student name)

{

cout << "Name: " << name.firstname << "," << name.lastname << endl;

cout << "Grade:" << char(name.grade + 1) << endl;

cout << "Age:" << name.age << endl;

}

/\*

#include <iostream>

using namespace std;

const int Asize = 20;

struct student

{

char firstname[Asize];

char lastname[Asize];

char grade;

int age;

};

int main()

{

student lcg;

cout << "what is your first name ?" << endl;

cin.getline(lcg.firstname, Asize);

//cin >> lcg.firstname[Asize];

cout << "what is your last name ?" << endl;

cin.getline(lcg.lastname, Asize);

//cin >> lcg.lastname[Asize];

cout << "what letter grade do you deserve ?" << endl;

cin >> lcg.grade;

cout << "what is your age ?" << endl;

cin >> lcg.age;

cout << "Name: " << lcg.lastname << ", "<< lcg.firstname << endl ;

cout << "Grade: " << lcg.grade << endl;

cout << "Age: " << lcg.age << endl;

return 0;

}

\*/

4.2 string类

#include<iostream>

#include<string>

int main()

{

using namespace std;

string name, dessert;

cout << "Enter your name: \n";

getline(cin, name);

cout << "Enter your favorite dessert: \n";

getline(cin, dessert);

cout << "I have some delicious " << dessert;

cout << " for you, " << name << ".\n";

return 0;

}

/\*

#include <iostream>

#include <string>

using namespace std;

int main()

{

string str1;

string str2;

cout << "Enter your name: \n";

cin >> str1 ;

cout << "Enter your favorite dessert:\n";

cin >> str2 ;

cout << "I Have AAAA" << str2 ;

cout << " for you, " << str1 << endl;

return 0;

}

\*/

错误：结果将导致只能输入一个str1，不能输入str2.

4.3 数组显示姓名

#include<iostream>

#include<cstring>

const int Asize = 20;

int main()

{

using namespace std;

char fname[Asize];

char lname[Asize];

char fullname[2 \* Asize + 1];

cout << "Enter your first name:";//输入名字，存储在fname[]数组中

cin.getline(fname, Asize);

cout << "Enter your last name:";//输入姓，存储在lname[]数组中

cin.getline(lname, Asize);

strncpy\_s(fullname, lname, Asize);//把姓lname复制到fullname空数组中

strcat\_s(fullname, ", ");//把“， ”附加到上述fullname尾部

strncat\_s(fullname, fname, Asize);//把fname名字附加到上述fullname尾部

fullname[2 \* Asize] = '\0';//为防止字符型数组溢出，在数组结尾添加结束符

cout << "Here's the information in a single string:" << fullname << endl;//显示组合结果

return 0;

}

/\*

#include <iostream>

#include <string>

const int number = 20;

struct xmname

{

char fname[number];

};

int main()

{

using namespace std;

string lname;

xmname str1;

cout << "Enter your first name:" << endl;

cin.getline(str1.fname, number); //结构、数组使用cin.getline

cout << "Enter your last name:" << endl;

getline(cin, lastname); //字符串cin直接放getline里面

cout << "Here's .... string:" << lname << ", " << str1.fname <<endl;

return 0;

}

\*/

4.4 string显示姓名

#include<iostream>

#include<string>

int main()

{

using namespace std;

string fname,lname,attach,fullname;

cout<<"Enter your first name:";

getline(cin,fname); //note:将一行输入读取到string类对象中使用的是getline(cin,str)

//它没有使用句点表示法，所以不是类方法

cout<<"Enter your last name:";

getline(cin,lname);

attach=", ";

fullname=lname+attach+fname;

cout<<"Here's the information in a single string:"<<fullname<<endl;

return 0;

}

4.5 结构初始化并显示

#include <iostream>

using namespace std;

const int Asize = 20;

struct CandyBar

{

char brand[Asize];

double weight;

int calory;

};

int main()

{

CandyBar snack;

//snack.brand = "Mocha Munch"; 错误，结构体数组赋值有三种方法，for循环按字符赋值， 用strcpy赋值，用memcpy赋值。

strcpy\_s(snack.brand, "Mocha Munc");

snack.calory = 350;

snack.weight = 2.3;

//CandyBar snack={"Mocha Munch",2.3,350}; 统一赋值

cout << "Here's the information of snack:\n";

cout << "brand:" << snack.brand << endl;

cout << "weight:" << snack.weight << endl;

cout << "calory:" << snack.calory << endl;

return 0;

}

4.6 结构内数组初始化并显示

#include<iostream>

const int Asize = 20;

struct CandyBar

{

char brand[Asize];

double weight;

int calory;

};

int main()

{

using namespace std;

CandyBar snack[3] = {

{ "Mocha Munch",2.3,350 },

{ "XuFuJi",1.1,300 },

{ "Alps",0.4,100 }

}; //数组包含多个元素

for (int i = 0; i<3; i++) //利用for循环来显示snack变量的内容

{

cout << snack[i].brand << endl

<< snack[i].weight << endl

<< snack[i].calory << endl << endl;

}

return 0;

}

4.7 结构的输入与显示

#include<iostream>

const int size = 20;

struct Pizza

{

char company[size];

double diameter;

double weight;

};

int main()

{

using namespace std;

pizza pie;

cout << "What's the name of pizza company:";

cin.getline(pie.company, Size);

cout << "What's the diameter of pizza:";

cin >> pie.diameter;

cout << "What's the weight of pizza:";

cin >> pie.weight;

cout << "company:" << pie.company << endl;

cout << "diameter:" << pie.diameter << "inches" << endl;

cout << "weight:" << pie.weight << "ounches" << endl;

return 0;

}

4.8 new为结构动态分配内存

#include<iostream>

#include<string>

const int Size = 20;

struct pizza

{

char company[Size];

double diameter;

double weight;

};

int main()

{

using namespace std;

pizza \*pie=new pizza;

cout << "What's the diameter of pizza:";

cin >> pie->diameter;

cout << "What's the name of pizza company:";

cin.getline(pie->company, Size);

cout << "What's the weight of pizza:";

cin >> pie->weight;

cout << "diameter:" << pie->diameter << "inches" << endl;

cout << "company:" << pie->company << endl;

cout << "weight:" << pie->weight << "ounches" << endl;

delete pie; //记得释放内存

return 0;

}

4.9 new动态分配数组

#include<iostream>

#include<string>

using namespace std;

struct CandyBar

{

string brand;

double weight;

int calory;

};

int main()

{

CandyBar\* snack = new CandyBar[3];

snack[0].brand = "A"; //单个初始化由new动态分配的内存

snack[0].weight = 1.1; //此时snack是指针，只能初始化一个值

snack[0].calory = 200;

snack[1].brand = "B";

snack[1].weight = 2.2;

snack[1].calory = 400;

snack[2].brand = "C";

snack[2].weight = 4.4;

snack[2].calory = 500;

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

{

cout << " brand: " << snack[i].brand << endl;

cout << " weight: " << snack[i].weight << endl;

cout << " calorie: " << snack[i].calory << endl << endl;

}

delete[] snack;

return 0;

}

4.10 输入数据，计算次数和平均值：使用array对象

#include <iostream>

#include <array>

int main()

{

using namespace std;

array<double, 4>ad = { 0 };

cout << "Enter your success of the three times 40 meters running:\n";

cin >> ad[0] >> ad[1] >> ad[2];

cout << "success1:" << ad[0] << endl;

cout << "success2:" << ad[1] << endl;

cout << "success3:" << ad[2] << endl;

ad[3] = (ad[0] + ad[1] + ad[2]) / 3;

cout << "average:" << ad[3] << endl;

return 0;

}

/\*

#include<iostream>

#include<string>

using namespace std;

struct saring

{

double ftime;

double stime;

double ttime;

double avr;

};

int main()

{

saring name;

cout << "Enter your success of the three times 40 meters running:\n";

cin >> name.ftime;

cin >> name.stime;

cin >> name.ttime;

cout << "success1:" << name.ftime<< endl;

cout << "success2:" << name.stime << endl;

cout << "success3:" << name.ttime << endl;

name.avr = (name.ftime + name.stime + name.ttime) / 3;

cout << "average:" << name.avr << endl;

return 0;

}

\*/

第五章 循环和关系表达式

5.1 输入两个整数，计算其之间的整数和

#include <iostream>

int main()

{

using namespace std;

cout << "please enter two integers: ";

int num1, num2;

cin >> num1 >> num2;

int sum = 0;

for (int temp = num1; temp <= num2; ++temp)//or temp++

sum += temp;

cout << "the sum from " << num1 << " to " << num2 << " is " << sum << endl;

return 0;

}

/\*

#include <iostream>

using namespace std;

int main()

{

int i, j;

cout << "please set the lowest number" << endl;

cin >> i;

cout << "please set the bigest number" << endl;

cin >> j;

int sum=0;

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

{

sum += i ;

}

cout << sum << endl;

}

\*/

5.2 计算100! 的值

#include <iostream>

#include <array>

using namespace std;

int main()

{

array<long double, 101> ai = {0} ; //注意0~100共101个数

ai[1] = ai[0] = 1LL;

for (int i = 2; i <= 100; i++)

ai[i] = i \* ai[i - 1];

for (int i = 0; i <= 100; i++)

cout << i << "! = " << ai[i] << endl;

// std::cin.get();

return 0;

}

5.3 实时显示输入的累计和。输入0时，结束。

#include <iostream>

int main()

{

using namespace std;

cout<<"Please enter an integer: ";

int sum=0,num;

while((cin>>num)&&num!=0)

{

sum+=num;

cout<<"So far, the sum is "<<sum<<endl;

cout<<"Please enter an integer: ";

}

return 0;

}

5.4 比较两个数的增长

#include <iostream>

int main()

{

using namespace std;

double sum1, sum2;

sum1 = sum2 = 0.0;

int year = 0;

while (sum2 <= sum1)

{

++year;

sum1 += 10;

sum2 = (100 + sum2)\*0.05 + sum2;

}

cout << "经过" << year << "年后，cleo的投资价值才能超过daphne的投资价值。" << endl;

cout << "此时，cleo的投资价值为" << sum1 << "，而daphne的投资价值为" << sum2 << endl;

return 0;

}

/\*

#include <iostream>

int main()

{

using namespace std;

int years=1;

double dsum,csum;

dsum = 110;

csum = 105;

while (csum < dsum)

{

dsum = dsum + 10;

csum = csum + 0.05\*csum;

years++;

}

cout << "经过" << years << "年后，Cleo的投资价值才能超过Daphne的投资价值。" << endl;

cout << "此时，Cleo的投资价值为" << dsum << endl;

cout << "而Daphne的投资价值为"<<csum<< endl;

return 0;

}

\*/

5.5 循环输入每月的销售量，存储在char\*数组，并显示

#include <iostream>

const int MONTHS = 12;

const char\* months[MONTHS] = { "January","February","March","April","May","June","July","August","September","October","November","December" };

int main()

{

using namespace std;

int sales[MONTHS], sum = 0;

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

{

cout << "请输入在" << months[i] << "的C++ For Fools的销售量：";

cin >> sales[i];

sum += sales[i];

}

cout << "这一年中的C++ For Fools的总销售量为：" << sum << endl;

return 0;

}

5.6 循环输入每月的销售量，存储在二维数组，并显示

#include <iostream>

#include <string>

using namespace std;

const int MONTHS = 12;

string months[MONTHS] = { "January","February","March","April","May","June","July","August","September","October","November","December" };

const char\* years[3] = { "第一年","第二年","第三年" };

int main()

{

using namespace std;

int year\_sale[3], sum = 0, sales[3][MONTHS];

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

{

int temp = 0;

cout << years[i] << "的每个月销售量:" << endl;

for (int j = 0; j<MONTHS; j++)

{

cout << "请输入" << months[j] << "的销售量:";

cin >> sales[i][j];

temp += sales[i][j];

}

year\_sale[i] = temp;

sum += year\_sale[i];

}

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

cout << years[i] << "的销售量为：" << year\_sale[i] << endl;

cout << "这三年的总销售量为：" << sum << endl;

return 0;

}

5.7 结构和动态数组存储信息，并显示

#include <iostream>

#include <string>

using namespace std;

struct car {

string name;

int year;

};

int main()

{

cout << "How many cars do you wish to catalog? ";

int num;

//cin >> num; //导致第一次无法输入make name。

(cin >> num).get(); //因为后面make name没有输入cin.get();

car\* ps = new car[num]; //数组长度取决与赋予的输入值

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

{

cout << "Car #" << i + 1 << ":\n"; //将字符和变量分开输出

cout << "Please enter the make: ";

//cin.get(); //若前面没有get()返回，需要加上。

getline(cin, ps[i].name);

cout << "Please enter the year made: ";

//cin >> ps->year; //这样只能输入一次

(cin >> ps[i].year).get();

}

cout << "Here is your collection:\n";

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

cout << ps[i].year << " " << ps[i].name << endl;

delete[] ps;

return 0;

}

/\*

#include <iostream>

#include <string>

using namespace std;

struct car

{

string maker;

int year;

};

int main()

{

int number;

cout << "How many cars do you wish to catalog? ";

cin >> number; //(cin >> number).get();

car\* a = new car[number];

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

{

cout << "Car #" << i + 1 << ": " << endl;

cout << "Please enter the maker: ";

cin.get(); //不写

getline(cin, a[i].maker);

cout << "Please enter the year made: ";

cin >> a[i].year; //(cin>>a[i].year).get();

}

cout << "Here is your collection: " << endl;

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

cout << a[i].year << " " << a[i].maker << endl;

delete[] a;

return 0;

}\*/

5.8 计算输入的单词，直到Done停止，使用char数组

#include <iostream>

#include <cstring>

using namespace std;

const int lon = 6;

int main()

{

char \*doc[lon];

char\* ps = doc[lon];

cin.getline(\*doc, 20);

while( strcmp(\*doc,"done") )

{

cin.getline(\*doc, 20);

}

cout << doc ;

return 0;

}

5.9 计算输入的单词，直到Done停止，使用string对象

#include <iostream>

#include <cstring>

#include <string>

int main()

{

using namespace std;

string word;

int sum = 0;

cout << "Enter words (to stop,type the word done):\n";

cin >> word;

while (word!= "done")

// while (strcmp(word, "done")) //strcmp函数参数必须是两个地址

{

sum++;

cin >> word;

}

cout << "You entered a total of " << sum << " words.\n";

return 0;

}

5.10 嵌套循环

#include <iostream>

int main()

{

using namespace std;

cout << "Enter number of rows:";

int num;

cin >> num;

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

{

for (int j = num - i; j>1; j--)

cout << ".";

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

cout << "\*";

cout << endl;

}

return 0;

}

第六章 分支语句和逻辑运算符

6.1 大小写字母转换，直到@停止

#include <iostream>

#include <cctype>

int main()

{

using namespace std;

char ch;

cin.get(ch);

while (ch != '@')

{

if (isdigit(ch))

cin.get(ch);

else

{

if (islower(ch))

ch = toupper(ch);

else

ch = tolower(ch);

cout << ch;

cin.get(ch); //在循环内最后还要读取输入流，防止死循环

}

}

return 0;

}

6.2 输入数字到数组，遇到非数字停止，计算平均值和大于平均值的数

#include <iostream>

#include<cctype>

#include<array>

int main()

{

using namespace std;

double sum = 0, average = 0;

array<double, 10>ad = { 0 };

int i = 0, total = 0;

double temp;

while (cin >> temp && i<10 && !isdigit(temp))

{

ad[i] = temp;

sum += ad[i];

++i;

}

if (i != 0)

average = sum / i;

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

if (ad[j]>average)

++total;

cout << "这些数字的平均值为" << average << endl;

cout << "并且共有" << total << "个数字大于平均值。\n";

return 0;

}

/\*

#include <iostream>

using namespace std;

const int num = 10;

int main()

{

double donation[num];

int i = 0;

int count = 0;

double sum = 0.0;

cout << "please enter: \n";

while (i < num && cin >> donation[i])

{

sum += donation[i++];

}

if (i== 0)

cout << "no data--bye \n";

else

{

double average = sum / i;

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

{

if (donation[j] > average)

++count;

} cout << " average ="

<< average << endl

<< "the numbers bigger than the average: "

<< count << endl;

} return 0;

}

\*/

6.3 设计switch选择a.b.c.d

#include <iostream>

using namespace std;

int main()

{

cout << "Please enter one of the following choices:\n"

<< "c)carnivore p)pianist\n"

<< "t)tree g)game\n";

cout << "Please enter a c, p, t, or g: ";

char ch;

cin >> ch;

while (ch != 'c'&&ch != 'p'&&ch != 't'&&ch != 'g')

{

cout << "Please enter a c, p, t, or g: ";

cin >> ch;

}

switch (ch)

{

case 'c':

cout << "A maple is a carnivore.\n";

break;

case 'p':

cout << "A maple is a pianist.\n";

break;

case 't':

cout << "A maple is a tree.\n";

break;

case 'g':cout << "A maple is a game.\n";

}

return 0;

}

void showmenu()

{

cout << "please enter one of the following choices:\n"

<< "c) carnivore p)pianist \n"

<< "t)tree g)game \n";

6.4 结构，switch选择a,b,c

#include <iostream>

const int strsize = 20;

struct bop {

char fullname[strsize];

char title[strsize];

char bopname[strsize];

int preference;

};

int main()

{

using namespace std;

cout << "Benevolent Order of Programmers Report\n"

<< "a. display by name b. display by title\n"

<< "c. display by bopname d. diplay by preference\n"

<< "q. quit\n";

char ch;

bop member[5] = {

{ "Wimp Macho","English Teacher","DEMON",0 },

{ "Raki Rhodes","Junior Programmer","BOOM",1 },

{ "Celia Laiter","Super Star","MIPS",2 },

{ "Hoppy Hipman","Analyst Trainee","WATEE",1 },

{ "Pat Hand","Police","LOOPY",2 }

};

cout << "Enter your choice:";

while (cin >> ch && ch != 'q')

{

switch (ch)

{

case 'a':

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

cout << member[i].fullname << endl;

break;

case 'b':

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

cout << member[i].title << endl;

break;

case 'c':

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

cout << member[i].bopname << endl;

break;

case 'd':

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

{

if (member[i].preference == 0)

cout << member[i].fullname << endl;

else if (member[i].preference == 1)

cout << member[i].title << endl;

else if (member[i].preference == 2)

cout << member[i].bopname << endl;

}

break;

}

cout << "Next choice: ";

}

cout << "Bye!\n";

return 0;

}

6.5 计算个人所得税

#include <iostream>

using namespace std;

int main()

{

double income, tax;

cout << "Please enter your income: ";

while (cin >> income && income >= 0)

{

if (income <= 5000)

tax = 0;

else if (income <= 15000)

tax = 0.1\*(income - 5000);

else if (income <= 35000)

tax = 10000 \* 0.1 + 0.15\*(income - 15000);

else

tax = 10000 \* 0.1 + 0.15 \* 20000 + 0.2\*(income - 35000);

cout << "Your tax is: " << tax << endl;

cout << "Please enter your income: ";

} cout << "

Bye!\n";

return 0;

} /

6.6 动态分配的结构数组

#include <iostream>

#include <string>

using namespace std;

struct patron

{

string name;

double money;

};

int main()

{

int num,i,temp=0;

cout << "请输入捐款的人数：";

cin >> num;

patron \* ps = new patron[num]; //结构数组

cin.get();

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

{

cout << "please enter the donor name :";

getline(cin,ps[i].name);

cout << " please enter the donor amount :";

cin >> ps[i].money;

cin.get();

}

cout << "Grand Patrons:\n";

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

if (ps[i].money>10000)

{

cout << ps[i].name << "\n" << ps[i].money << endl;

++temp;

}

if (temp == 0)

cout << "none\n";

cout << "Patrons:\n";

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

if (ps[i].money <= 10000)

{

cout << ps[i].name << "\n" << ps[i].money << endl;

++temp;

}

if (temp == 0)

cout << "none\n";

delete ps;

return 0;

}

6.7 每次读取一个单词，计算元音数目

#include <iostream>

#include <cctype>

using namespace std;

int main()

{

char word[20];

int vow = 0, consonant = 0, other = 0;

cout << "Enter words (q to quit):\n";

while (cin >> word)

{

if (isalpha(word[0]))

{

if (word[0] == 'a' || word[0] == 'e' || word[0] == 'i' || word[0]

== 'o' || word[0] == 'u' ||

word[0] == 'A' || word[0] == 'E' || word[0] == 'I' || word[0] ==

'O' || word[0] == 'U')

vow++;

else if (word[0] == 'q' && strlen(word) == 1)

break;

else

consonant++;

}

else

other++;

} cout << vow << "words beginning with vowels\n"

<< consonant << " words beginning with consonants\n"

<< other << " others\n";

return 0;

}

6.8 打开文件读取字符

#include <iostream>

#include <fstream>

#include <cstdlib>

using namespace std;

const int Size = 20;

int main()

{

char filename[Size];

ifstream infile; //用于处理文件输入的对象

cout << "Enter name of data file: ";

cin.getline(filename, Size); //输入文件名

infile.open(filename); //寻找指定的文件名

if (!infile.is\_open()) //如果无法打开

{

cout << "Could not open the file " << filename << endl;

cout << "Program terminating.\n";

exit(EXIT\_FAILURE);

} char a;

int count = 0;

infile >> a;

while (infile.good()) //指出是否输入成功，有无EOF

{

++count;

infile >> a;

} if (

infile.eof())

cout << "End of file reached.\n";

else if (infile.fail())

cout << "Input terminated by data mismatch.\n";

else

cout << "Input terminated for unknown reason.\n";

if (count == 0)

cout << "No data processed.\n";

else

cout << "The text contains " << count << " character(s)" << endl;

infile.close();

return 0;

}

6.9 打开文件并重新排版

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <string>

using namespace std;

struct charity

{

string name;

double money;

};

int main()

{

string filename;

ifstream infile;

cout << "Enter name of data file: ";

getline(cin, filename);

infile.open(filename);

if (!infile.is\_open())

{

cout << "Could not open the file " << filename << endl;

cout << "Program terminating.\n";

exit(EXIT\_FAILURE);

}

int number, count = 0;

infile >> number;

charity \*pt = new charity[number];

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

{

infile.get();

getline(infile, pt[i].name);

infile >> pt[i].money;

if (pt[i].money > 10000)

count++;

} if (count == 0)

cout << "None(money > 10000)";

else

{

cout << "Grand Patron:\n";

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

{

if (pt[i].money > 10000)

cout << pt[i].name << " " << pt[i].money << endl;

}

} if (10 -count == 0)

cout << "None(money < 10000)";

else

{

cout << "Patron:\n";

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

{

if (pt[i].money < 10000)

cout << pt[i].name << " " << pt[i].money << endl;

}

} delete[] pt;

return 0;

}

7.1 求调和平均数

#include<iostream>

using namespace std;

int main()

{

int x, y;

double avr;

while( cin >> x >> y && x!=0 && y!=0)

{

avr = 2.0\*x\*y / (x + y);

cout << avr << endl;

}

cout << "Bye" << endl;

return 0;

}

7.2