#include <iostream> //reading & writing from keyboard

#include <cmath> //the square root function & absolute value

#include <string>//when use string, include this

#include <vector>//when use vector, include this

#include “.h” //the class head file

#include <fstream> //read and write file

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

{

/\* code \*/

}

int main(int argc, char const \*argv[])

{

/\* code \*/

return 0;

}

if (/\* condition \*/)

{

/\* code \*/

}

else

/\*code\*/

cout, cin, endl, vector, sort前面要有std::

定义的function前面要有返回的值的类型。bool return T/F, void 没有return的值（可以只打一个return）

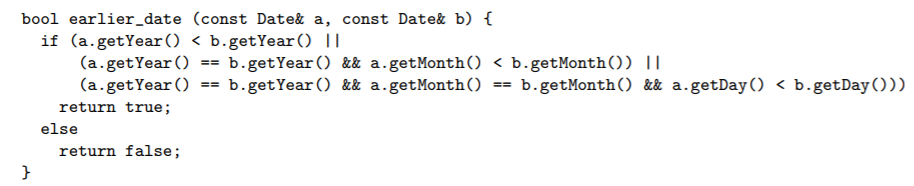
char型--------只有一个字符

string 型------字符串

cout<<”h”<<endl;

cin>>a;

std::vector<int> v //建立vector，<>中为vector所存元素类型

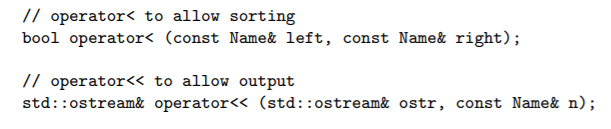
v.push\_back(s); //想用vector先#include

vector或string有.size( )

//File: date.h

#ifndef \_\_name\_h\_

#define \_\_name\_h\_

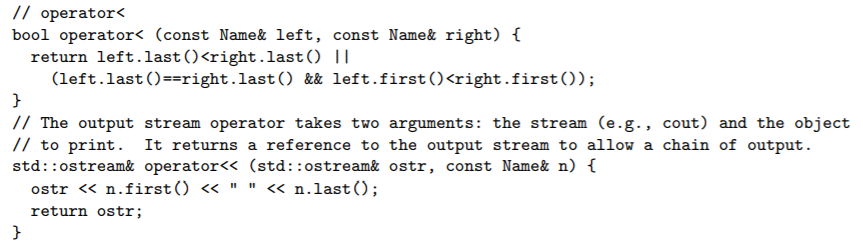
class Date{

public:

Date();

Date(int aMonth, int aDay, int aYear);

//Accessors

 int getDay() const;

//Modifiers

void setDay(int aDay);

//other member functions that operate on date class

bool isEqual (const Date& date2) const;

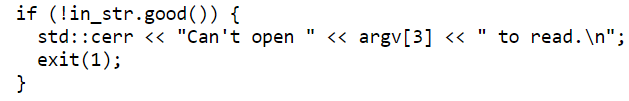
void print() const;

private: //representation

int day;

int month;

int year;

};

#endif

//File: date.cpp

#include <iostream>

#include ”date.h”

Date::Date(){//default constructor

day=1;

month=1;

year=1900;

}

Date::Date(int aMonth, int aDay, int aYear) {

// construct from month, day, & year

month = aMonth;

day = aDay;

year = aYear;

}

int Date::getDay() const { return day; }

void Date::setDay(int d) { day = d; }

bool Date::isEqual(const Date& date2) const

{

return day == date2.day && month == date2.month && year == date2.year;

}

void Date::print() const {

std::cout << month << "/" << day << "/" << year;

}

Std::sort(v.begin(),v.end(),earlier\_date);

Non-member function

.h file

.cpp file

读取文件

std::ifstream in\_str(argv[3]);（读取）

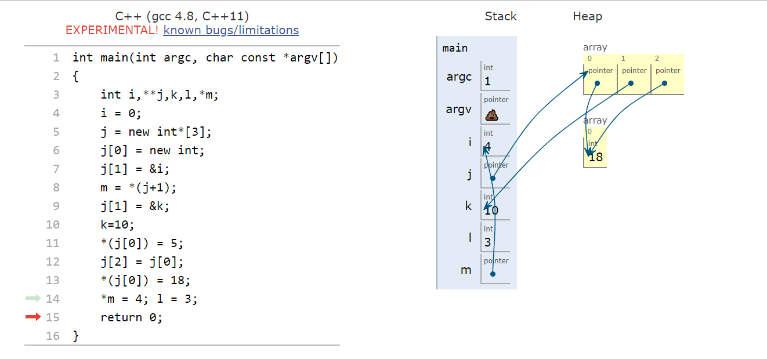
while (in\_str >> my\_variable) {

// do something with my\_variable

}

&：取地址运算符

\*：指针运算符（间接访问运算符）



空指针：NULL

Passing by reference is more efficient than passing than value.

float a[5]={0,1,2,3,4}

float\* p;

a[2] ↔\*(a+2) a[0] ↔ \*a

p=a ↔ p=&a[0] p=a+2 ↔ p=&a[2]

and && or ||

#include <string> 字符串切片

Sting.substr(a,b) 从a开始切b个

字符串转数字

#include<sstream>

Int a;

std::istringstream ia(string);

ia>>a

数字转字符串

#include <string>

Std::to\_string(number)

输出靠左：

out\_str.width(length\_code);out\_str<<std::left<<dept1[a].getcode()<<" ";

输出靠右：

std::setw(2) << i\*5

如果要在function里面更改导入的参数的值，要用引用传递。

Clean up memory:

delete a[0];

delete [] a[1];

delete [] a;

