1. **程序纠错题（20分）**

打开程序文件proj1.cpp。程序中有类Clock（“时钟”）的定义和主函数main的定义。程序中位于每个“// ERROR\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*”下的语句行有错误，请加以更正。更正后程序的输出应该是：

Initial times are

0 d : 0 h : 0 m : 59 s

After one second times are

0 d : 0 h : 1 m : 0 s

注意：只能修改每个“// ERROR\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*”下的那一行，不要改动程序中的其他内容。

// proj1.cpp

#include <iostream>

using namespace std;

class Clock

{

public:

Clock(unsigned long i = 0);

void set(unsigned long i = 0);

void print() const;

void tick();

Clock operator++();

private:

unsigned long total\_sec,seconds,minutes,hours,days;

};

Clock::Clock(unsigned long i)

: total\_sec(i), seconds(i % 60),

minutes((i / 60) % 60),

hours((i / 3600) % 24),

days(i / 86400) {}

void Clock::set(unsigned long i)

{

total\_sec = i;

seconds = i % 60;

minutes = (i / 60) % 60;

hours = (i / 3600) % 60;

days = i / 86400;

}

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【6分】

void Clock::print()

{

cout << days << " d : " << hours << " h : "

<< minutes << " m : " << seconds << " s" << endl;

}

void Clock::tick()

{

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【7分】

set(total\_sec++);

}

Clock Clock::operator ++()

{

tick();

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【7分】

return this;

}

int main()

{

Clock ck(59);

cout << "Initial times are" << endl;

ck.print();

++ck;

cout << "After one second times are" << endl;

ck.print();

return 0;

}

1. **程序填空题（20分）**

打开程序文件proj2.cpp。程序中定义了坐标点类Point和多边形类Polygon。请在程序中的横线处填写适当的代码，然后删除横线，以实现上述类定义。

此程序的正确输出结果应为：

多边形周长：4

注意：只能在横线处填写适当的代码，不要改动程序中其它内容，也不能删除或移动“//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*”。

// proj2.cpp

#include <iostream>

#include <cmath>

using namespace std;

class Point{

public:

double x,y;

Point(double x=0.0, double y=0.0): x(x),y(y){}

double distanceTo(Point p)const{

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【5分】

return sqrt((x-p.x)\*(x-p.x)+\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

}

};

const int MAX\_SIDES=100;

class Polygon{

Point vertex[MAX\_SIDES-1];

int num\_of\_side;

public:

Polygon():num\_of\_side(0){}

void addVertex(Point p){

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【5分】

if(num\_of\_side<MAX\_SIDES){ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; num\_of\_side++; }

}

void addVertex(Point p,int pos){

if(num\_of\_side>=MAX\_SIDES) return;

if(pos>=num\_of\_side || pos<0) return;

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【5分】

for(int i=num\_of\_side-1;i>=pos;i--) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

vertex[pos]=p;

num\_of\_side++;

}

double circumference(){

if(num\_of\_side<3) return 0.0;

double cir=vertex[0].distanceTo(vertex[num\_of\_side-1]);

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* 【5分】

for(int i=0; i<num\_of\_side-1; i++) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

return cir;

}

};

int main(){

Polygon poly;

poly.addVertex(Point(0.0,0.0));

poly.addVertex(Point(1.0,1.0));

poly.addVertex(Point(1.0,0.0));

poly.addVertex(Point(0.0,1.0),1);

cout<<"多边形周长："<<poly.circumference()<<endl;

return 0;

}

1. **程序补全题（20分）**

打开程序文件proj3.cpp。程序中的IntQueue是以链表方式实现的整数队列，队列中的每个元素是整数，保存在一个结点中，并通过结点的链接形成一个链，head和tai1分别是指向该链的队首结点和队尾结点的指针。既然是队列，入队元素总是添加在队列尾部，出队元素总是队列的首元素。请补充完整成员函数show。此程序的正确输出结果应为：

刚建立时：

添加8，3后：8 3

添加5后：8 3 5

删除首元素8后：3 5

添加-7后：3 5 -7

删除首元素3后：5 -7

注意：只允许在//\*\*\*\*\*\*\*\*333\*\*\*\*\*\*\*\*和//\*\*\*\*\*\*\*\*666\*\*\*\*\*\*\*\*两行之间填入若干语句，不允许改动程序中的其它内容。

// proj3.cpp

#include <iostream>

using namespace std;

class Node{

public:

int data;

Node\* next;

Node(int intdata):data(intdata),next(NULL){}

};

class IntQueue{

Node \*head;

Node \*tail;

public:

IntQueue():head(NULL),tail(NULL){}

bool isEmpty()const{ return head==NULL; }

void enqueue(int intdata);

bool dequeue(int& intdata);

void show(ostream& os=cout)const;

};

void IntQueue::enqueue(int intdata){

if(isEmpty())

head=tail=new Node(intdata);

else

tail=tail->next=new Node(intdata);

}

bool IntQueue::dequeue(int& intdata){

if(isEmpty()) return false;

intdata=head->data;

Node \*p=head;

head=head->next;

delete p;

return true;

}

void IntQueue::show(ostream& os) const {

//\*\*\*\*\*\*\*\*333\*\*\*\*\*\*\*\* 【20分】

//\*\*\*\*\*\*\*\*666\*\*\*\*\*\*\*\*

return;

}

int main(){

IntQueue que;

cout<<"刚建立时："; que.show();

que.enqueue(8);

que.enqueue(3);

cout<<"添加8，3后："; que.show();

que.enqueue(5);

cout<<"添加5后："; que.show();

int x;

que.dequeue(x);

cout<<"删除首元素"<<x<<"后：";que.show();

que.enqueue(-7);

cout<<"添加-7后："; que.show();

que.dequeue(x);

cout<<"删除首元素"<<x<<"后：";que.show();

return 0;

}

1. **程序设计题（40分）**

定义一个Person类，拥有保护数据成员：name、age和sex分别表示人的姓名、年龄和性别。雇员类Employee是Person的派生类，新增私有数据成员：部门department和薪水salay，新增成员函数display()实现雇员的姓名、年龄、性别、部门和薪水的输出。其中，派生类对象输出时需要用到的所有参数均由派生类的构造函数给出。

（1）根据描述画出类图。【10分】

（2）请用C++代码编程实现，并设计主函数进行测试（程序命名为proj4.cpp）。【30分】