

软件学院本科生(特色班)21—22 学年第 2 学期高级语言程序设计（2）期末考试试卷（A 卷）

专业： 软件工程      年级： 2021 级      学号：      姓名：      成绩：      草稿区

得 分

一、单项选择题（本题共 48 分，每小题 4 分）

注意：请将所有小题的答案填写在下表中。

题号	1	2	3	4	5	6	7	8	9	10	11	12
答案												

1.下面程序的输出结果是什么？

```
#include <iostream>
using namespace std;
class X {
private:
    int num;
public:
    X(int x) { num=x;}
    X(X & y) { num = ++y.num; }
    void display() { cout << num << " "; }
};
void main()
{
    X a(10);
    X b(a);
    a.display();
}
```

```
        b.display();  
    }
```

- |          |          |
|----------|----------|
| A. 10 10 | B. 10 11 |
| C. 11 11 | D. 11 10 |

2. 在 C++ 中下列声明之间有何不同？

```
int num;  
const int * myptr1 = & num;  
int * const myptr2 = & num;
```

- A. 根本没有不同。
- B. 不能用 myptr1 改变整数的内容，然而可以用 myptr2 改变它指向的整数的内容
- C. 不能用 myptr2 改变整数的内容，然而可以用 myptr1 改变它指向的整数的内容

3. 类 Test 的定义如下：

```
class Test {  
    int value;  
public:  
    Test() { value = 0;}  
    SetVariable(int i) { value = i; }  
};
```

则对语句

```
Test *p, my;    p = & my;
```

的正确描述是

- A. 语句 `p = & my` 是把对象 `my` 赋值给指针变量 `p`
- B. 语句 `Test *p, my` 会调用两次类 `Test` 的构造函数
- C. 语句 `*p.SetVariable(5)` 的调用是正确的
- D. 语句 `p->SetVariable(5)` 与语句 `my.SetVariable(5)` 等价

4. 下面程序段的输出是什么？

```
#include <iostream>
using namespace std;
class example{
private:
    static int num;
public:
    example() { num++; cout << num << " "; }
    ~example() { num--; cout << num << " "; }
};
int example::num = 1;
void main()
{
    example object1;
    example object2;
}
```

- A. 1 1 1 1
- B. 2 1 2 1
- C. 2 2 3 2
- D. 2 3 2 1

5. 下面程序的运行结果是？

```
#include <iostream>
using namespace std;
int a=0;
void f ( ) { a++; }
class S {
public:
    int a;
    S(): a(100) {}
    void f ( ) {
        ::a++;
        ::f ( );
        a--;
    };
};
void main ( )
{
    S s;
    s.f();
    cout << s.a << " " << a << endl;
```

- A. 101 99
- B. 2 99
- C. 99 1
- D. 99 2

}

6. 下面程序的运行结果是 ?

```
#include <iostream>
using namespace std;
class Base {
public:
    void print() { cout << "Base" << endl; }
};
class Derived: public Base {
public:
    void print() {cout << "Derived" << endl; }
};
void func( Base & r) {r.print();
}
void main(){
    Derived d;
    func(d);
}
```

- A. Base
- B. Derived
- C. Base  
Derived
- D. Derived  
Base

7. 下面程序的运行结果是 ?

```
#include <iostream>
class complex {
    float real, image;
public:
    complex(float r=0,float i=0){
        real=r; image =i;}
    complex operator+(complex & cmpx) {
        return complex( real+cmpx.real, image+cmpx.image);}
    complex operator+(float num){
        return complex(real+num, image+num);}
    void operator=(complex &cmpx){
        real=cmpx.real; image=cmpx.image;}
    void show(){
        std::cout<<real<<" + "<<image<<" i"<<std::endl;}
};
```

- A.  $109 + 106 i$
- B.  $109 + 100 i$
- C.  $106 + 109 i$
- D.  $100 + 100 i$

```
void main(){
    complex p1(1,2), p2(2,3), p3(3,4), p4;
    p4=p1 + p2 + p3 + 100;
    p4.show();
}
```

8.下面程序的运行结果是 ?

```
#include <iostream>
class A {
public:
    virtual void fun()=0;
};
class B:public A {
public:
    void fun(){
        std::cout << " B ";
    }
};
class C: public A {
public:
    void fun() {
        std::cout << " C ";
    }
};
```

```
class D:public A {
public:
    void fun(){
        std::cout<< " D ";
    };
};
void main(){
    A *p;
    B b;
    C c;
    D d;
    p=&b;
    p->fun();
    p=&c;
    p->fun();
    p=&d;
    p->fun();
}
```

- A. B C D
- B. C D B
- C. B D C
- D. 编译出错，程序不能够运行

9.下面程序的运行结果是 ?

```
#include <iostream>
template <class T>
T max( T x, T y)
{ return ( x>=y)? x:y;
}
template <class T>
T max(T x, T y, T z)
{ T t;
    t = (x>=y)?x:y;
    return (t>=z)?t:z;
}
```

- A. 40
- 50
- B. 50
- 40
- C. 30
- 40
- D. 40
- 30

```

void main(){
    int x=10, y=20, max1;
    float a=30, b=40, c=50, max2;
    max1=max(x, (int) c);
    max2=max(a,b,(float) y);
    std::cout << max1 << endl;
    std::cout << max2 << endl;
}

```

---

10. 下面程序的运行结果是 ?

```

#include <iostream>
namespace BASE {
    int i;
}
namespace X {
    using BASE::i;
}
namespace Y {
    int i;
}
main() {
    using namespace X;
    i=0;
    using namespace Y;
    Y::i=1;
    printf("%d %d %d\n", BASE::i, X::i, Y::i);
}

```

- A. 0 1 0
- B. 0 1 1
- C. 1 0 0
- D. 0 0 1

11. 有如下的程序，其运行结果是？

```

#include <iostream>
using namespace std;
class Base{
public:
    virtual void f() { cout << "Base\n"; };
};
class Derived1: public Base{
public:

```

- A. Base
- B. Derived1
- C. Derived2
- D. q is NULL

```
void f() {cout << "Derived1\n"; };  
};  
class Derived2: public Derived1{  
public:  
    void f() { cout << "Derived2\n"; };  
};  
int main(){  
    Base * p = new Derived2;  
    Derived1 * q = dynamic_cast<Derived1*> (p);  
    if (q){  
        q->f();  
    } else{  
        cout << "q is NULL\n";  
    }  
    return 0;  
}
```

12. 多重继承的构造顺序一般可分为 3 步，下面给出这 3 个步骤：

Step1:基类的构造函数按照它们声明被继承的顺序构造。

Step2: 类成员对象的构造函数按照它们声明的顺序构造。

Step3: 类自己的构造函数。

这 3 个步骤的正确顺序是 ( )。

A. step3、step2、step1      B. step2、step3、step1

C. step1、step2、step3      D. step3、step1、step2

得 分

二、编程题（本题共 52 分）

1 (20 分) 设有两个文本文件 file1.txt 以及 file2.txt。file1.txt 存放学生英语课程的成绩，每行存放一个学生的姓名，

然后是其英语课程的成绩（姓名与成绩间使用空格进行分割），内容如下：

```
Student_1_name English_score_1
Student_2_name English_score_2
Student_3_name English_score_3
....
```

例如：

```
stdname1 80
stdname2 82
```

file2.txt 存放学生数学课程的成绩，每行存放一个学生的姓名，

然后是其数学课程的成绩（姓名与成绩间使用空格进行分割），内容如下：

```
Student_3_name Math_score_3
Student_1_name Math_score_1
Student_2_name Math_score_2
....
```

例如：

```
stdname3 91
stdname2 82
stdname1 78
```

两个文件中的学生是同一组学生。也就是说 file1.txt 学生与 file2.txt 的学生完全一致，但两个文件中学生姓名的出现

顺序不同（姓名中不存在同名）。编写一个程序将这两个文件中的信息合并起来，存放在文件 file3.txt 中，每行依次

存放一个学生的姓名、英语课程成绩以及数学课程成绩，姓名与成绩间使用空格进行分割，内容如下：

```
Student_1_name English_score_1 Math_socre_1
Student_2_name English_score_2 Math_socre_2
Student_3_name English_score_3 Math_socre_3
....
```

例如：

```
stdname2 82 82
stdname3 76 91
stdname1 80 78
```

file3.txt 中学生姓名的出现顺序可以和 file1.txt 以及 file2.txt 中的都不同。

注：学生的姓名中不含空格，成绩均为整数。可使用 STL 及 string 类。





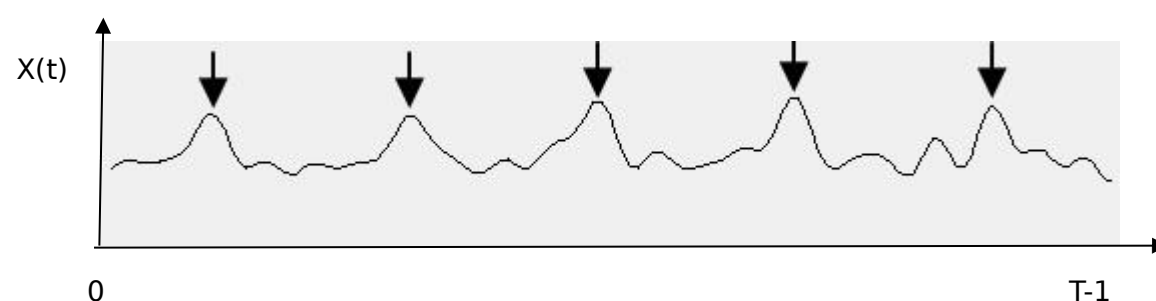
2 ( 20 分 ) 编写一个类模板 CPoint , 存放平面坐标系中一个点的 x 和 y 坐标。它具有一个模板参数 T , 表示 x 以及 y 的类型 , 可以是 int, float, double 等基本数字类型。该类模板具有成员函数 x( )、成员函数 y( ) , 分别返回其 x 和 y 坐标。再编写一个全局的函数模板 abs( ) , 求取 CPoint 所表示的点到坐标原点的距离 , 也就是  $\sqrt{x^2 + y^2}$ 。在 main() 中编写以下代码 , 测试这两个模板的功能。创建数据成员类型为 int, float, double 类型的模板类对象 , 并调用 abs, 输出对象到原点的距离。

注 : 求开方可以使用 : <math.h> 内 double sqrt(double)



3 ( 12 分 ) 有一段信号  $x(t)$ ,  $t=0..T-1$ 。时刻  $t$  以及该时刻信号的取值被称为信号的一个样本。这段信号含有若干个峰。所谓峰指的是一个样本的值比其相邻的两个样本的值要大。从这段信号中任取  $K=5$  个峰形成一个峰群。有的峰群, 比如图中黑色箭头所指的, 能刻画这段信号的周期性。这些峰具有更大的幅度, 相邻峰之间的距离 (被称为峰距) 也相差不大。我们使用下面的评估函数来评判一个峰群: 峰的幅度的和, 减去峰距的离散度。此处的离散度被定义为各峰距与峰距均值的差的绝对值的和。

约定这  $K$  个峰中, 第  $i$  个峰,  $i=0..K-1$ , 位于  $i * T/K$  以及  $(i+1)*T/K$  的时间范围内。给定一段信号 (以 vector 存储, 定为 `vector<double>signal`, 长度为  $T$ ) 以及  $K$ , 求取使得评估函数值最大的峰群, 输出该峰群的峰的位置。



- (1) 实现求取 `signal` 中峰位置的函数, 返回 `vector<int> peak`。(6 分)
- (2) 根据  $K$  值获取候选峰群列表(3 分)
- (3) 求取候选峰群的评估函数值, 并求取最大的峰群(3 分)



