

# 上海交通大学试卷 (A 卷)

( 2021 至 2022 学年 第 1 学期 )

班级号 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

课程名称 \_\_\_\_\_ 程序设计思想与方法 (C++) \_\_\_\_\_ 成绩 \_\_\_\_\_

## 一. 单项选择题 (每题 1 分, 共 15 分)

( ) 1 经典的计算机硬件结构是由冯·诺依曼提出的, 因此被称为冯·诺依曼体系结构, \_\_\_\_\_ 不属于其中的一部分:

- A. 运算器      B. 控制器      C. 编译器      D. 存储器      E. 以上都属于

( ) 2 设有整型数组 a[10], 要输出数组 a 的 10 个元素, 错误的是 \_\_\_\_\_

- A. for ( int \*p=a, int i=0; i<10; ++i )      cout << p[i] ;  
B. for ( int i=0; i<10; ++i )      cout << \*(a+i);  
C. for ( int \*p=a; p<a+10; ++p )      cout << \*p ;  
D. for ( int i=0; i<10; ++i )      { cout << \*a ;    ++a; }

( ) 3 以下关于函数参数的叙述不正确的是 \_\_\_\_\_

- A. 函数的参数不仅仅可以是函数的输入, 还可以是函数的输出  
B. 函数的形参命名可以任意, 只要符合标识符规则  
C. 对象作为实参时系统将自动调用拷贝构造函数  
D. 函数参数的值可以是内存单元地址

( ) 4 已知程序

```
#include <iostream>

using namespace std;

int main()
{
    for(int i=5; i>1; --i) cout<<i; cout<<i; return 0; }
```

则运行该程序之后, 输出 \_\_\_\_\_

- A. 543210      B. 54321      C. 55443322      D. 5432      E. 以上都不对

我承诺，我将严格遵守考试纪律。

承诺人：\_\_\_\_\_

题号	一	二	三	四
得分				
批阅人(流水阅卷教师签名处)				

( ) 5 下列关于 C++ 类中 private 成员和 public 成员的说法中，正确的是\_\_\_\_\_

- A. 必须先定义 private 成员，再定义 public 成员
- B. 访问控制属性 private 和 public 只能出现一次，不能出现多次
- C. 私有成员可以由类的成员函数访问
- D. 数据成员一般说明为 public，需要被用户调用的函数说明为 private

( ) 6 考虑以下程序段，则\_\_\_\_\_

```
class base {  
protected:  
    int x;  
public:  
    void setx(int k) { x=k; }  
};  
  
class derived:public base {  
    int y;  
public:  
    void setxy(int j, int k) { x=j; y=k; }  
};
```

- A. 除了系统自动生成的成员函数，derived 只有 1 个数据成员和 1 个成员函数
- B. 除了系统自动生成的成员函数，derived 只有 1 个数据成员和 2 个公有成员函数
- C. derived 的成员函数 setxy 是错的，必须通过 base 的公有成员函数修改数据成员 x 的值
- D. 以上论述都是错误的

( ) 7 若有定义语句: `int a[5][10], *p, *q[5];`且  $0 \leq i < 5$ , 则错误的赋值是\_\_\_\_\_

- A. `p=a;`      B. `q[i]=a[i];`      C. `p=a[i];`      D. `p=&a[2][1];`

( ) 8 已知语句 `myfun(x) = yourfun(y);`可以正常运行, 以下说法中正确的是\_\_\_\_\_

- A. `myfun` 函数不能返回非静态的局部变量      B. `yourfun` 函数不能返回非静态的局部变量  
C. `myfun` 函数的参数只能采用引用传递      D. `yourfun` 函数的参数只能采用引用传递

( ) 9 若变量已正确定义, 以下和语句 `if(a>b) k=0;else k=1;`等价的是\_\_\_\_\_

- A. `k=(a>b)?1:0;`      B. `k=a>b;`      C. `k=(a<=b)?1:0;`      D. `k=!a<b;`

( ) 10 下列关于模板的说法正确的是\_\_\_\_\_

- A. 在 C++ 中, 模板包括函数模板与类模板      B. 模板的形式参数不能为类型  
C. 模板类实例化后得到类模板      D. 类模板的成员函数可以是普通函数, 还可以是函数模板

( ) 11 对于 `switch-case`, 说法不正确的是\_\_\_\_\_

- A. 在 `case` 后的各常量表达式的值可以相同, 程序将执行其中第一个 `case` 后的语句  
B. 在 `case` 后, 允许有多个语句, 并且可以不用 `{ }`括起来  
C. 一个 `switch` 中可以有任意数量的 `case` 语句  
D. `default` 子句可以省略不用

( ) 12 关于构造函数和析构函数, 他们的共同特性是\_\_\_\_\_

- A. 都没有返回类型      B. 都可以重载  
C. 都可以有参数      D. 都必须自己定义

( ) 13 以下关于异常的叙述, \_\_\_\_\_是错误的

- A. 程序员可以将数组下标越界定义为一种异常  
B. 如果 `try` 语句块中包含了可能抛出异常的代码, 那么一旦抛出了异常, 则程序退出 `try` 语句块, 进入 `try` 后面的异常捕获和处理  
C. 类的创建者必须能够发现和处理该类在使用过程中可能出现的异常  
D. 使用 `sin` 函数但没有写 `#include <cmath>` 这种情况不属于异常

( ) 14 以下程序均由多个文件构成，\_\_\_\_\_是正确的。其中//后面为文件名，而...表示该处程序省略，无错误

A	B	C	D
<pre>//test.h extern int a; void f();  //test1.cpp #include "test.h" int a; int main() {...}  //test2.cpp #include&lt;iostream&gt; using namespace std; #include "test.h" void f() {     cout&lt;&lt;a; }</pre>	<pre>//a.h #ifndef _A #define _A #include "c.h" #endif  //b.h #ifndef _B #define _B #include "c.h" #endif  //c.h struct A{};  //main.cpp #include "a.h" #include "b.h" int main() {...}</pre>	<pre>//file1.cpp void f(); extern int x;  int main() {     f();     ... }  //file2.cpp #include&lt;iostream&gt; using namespace std; static int x;  void f() {     cout &lt;&lt; x; }</pre>	<pre>//m1.h #ifndef _M1 #define _M1 int a=1; #endif  //m1.cpp #include "m1.h" void f() {...}  //m2.cpp #include "m1.h" void g() {     a++; }  //m3.cpp #include "m1.h" int main() {...}</pre>

( ) 15 以下程序段中，\_\_\_\_\_可以运行并且得到合理正确的结果

- A. int n; cin>>n; if(n=0) cout<<"It is Zero"; else cout<<"It is not zero";
- B. int x=12345678; x\*=x; cout<<x;
- C. char s[]={ '1','2','3' }; cout<<s;
- D. 以上选项都不能得到合理正确的结果

二 . 给出下列程序段的运行结果（请将结果写在程序右侧空白处，每题 5 分， 共 35 分）

1.

```
#include <iostream>

using namespace std;

int main( ){

    char a[3] = {'d','c','h'};

    float b[3] = {1, 2, 3};

    int c[3] = {0};

    int j;

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

        j = 0;

        while ((j<3)&&(a[i] <= a[j]))

        {

            c[i]+=((a[j]-'a')/b[j]);

            j++;

        }

    }

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

        cout << c[i] << " ";

    return 0;

}
```

2

```
#include <iostream>

using namespace std;

void f(int a, int &b)

{

    static int c=3;

    c+=c*2;
```

```

        b=(a++)+c;
    }

int main()
{
    int a[10]={3,15,9,8,12,35};
    int b[40]={0};
    for (int i=0; i<3; i++)
    {
        f(a[i], b[i]);
        b[i]++;
        cout<<b[i]<<endl;
    }
    return 0;
}

```

3

```

#include <iostream>
using namespace std;
int main( )
{
    char ch[2][5]= {"6967","8284"},*p[2];
    int i,j,s=0;
    for(i=0; i<2; i++)
        p[i]=ch[i];
    for(i=0; i<2; i++)
        for(j=0; p[i][j]>'\0'; j+=2)
            s=10*s+p[i][j]-'0';
    cout<<s<<endl;
    return 0;
}

```

```
}
```

4

```
#include <iostream>

#include <cstring>

using namespace std;

char* myfun (char *s1, char s2[])
{
    for (int i=0; s1[i] != '\0'; ++i) s1[i] ++;

    strcat (s2, s1);

    return (s2+1);
}

int main()
{
    char *a = new char[10], *p = new char[10], b[5] = "AB";

    *p = *a = 0;

    for (int i=0; i < 3; ++i)
    {
        strcat (a, myfun(b, p));

        cout << a << endl;
    }

    delete []a;

    delete []p;

    return 0;
}
```

5

```
#include<iostream>
```

```

using namespace std;

class CMoney
{
private:
    int yuan,jiao,fen;
public:
    CMoney(int y=0, int j=0, int f=0);
    void disp();
};

CMoney::CMoney(int y, int j, int f)
{
    yuan=y;
    jiao=j;
    fen=f;
}

void CMoney::disp()
{
    cout<<"我有"<<yuan<<"元"<<jiao<<"角"<<fen<<"分"<<endl;
}

int main()
{
    CMoney m,m1(100),m2(100,5),m3(100,5,2);
    m.disp();
    m1.disp();
    m2.disp();
    m3.disp();
    return 0;
}

```



6

```
#include <iostream>
```

```
#define N 4
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a[N], n = 1234, i = 0;
```

```
    while (n && i < N)
```

```
    {
```

```
        a[i] = n % 10;
```

```
        n /= 10;
```

```
        i++;
```

```
    }
```

```
    for (i = 0; i < N; i++)
```

```
    {
```

```
        switch (a[i])
```

```
        {
```

```
            case 1:
```

```
                cout << "One ";
```

```
                break;
```

```
            case 2:
```

```
                cout << "Two ";
```

```
                break;
```

```
            case 3:
```

```
                cout << "Three ";
```

```
            default:
```

```
                cout << "Default" << endl;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

7

```
#include<iostream>

using namespace std;
```

```
void fun(int A[],int n)
```

```
{
    if(n>0)
    {
        fun(A,n-1);
        int tmp = A[n-1];
        for (int i = n-1; i >= 0; i--) {
            if (i == 0 || A[i-1] <= tmp) {
                A[i] = tmp;
                break;
            }
            else
                A[i] = A[i-1];
        }
    }
}
```

```
int main()
```

```
{
    int num[5]={1,3,5,0,2};
    fun(num, 5);
    for (int i = 0; i < 5; i++)
        cout << num[i] << " ";
    return 0;
}
```

### 三 . 程序填空（每空 1 分， 共 10 分）

1. 二分查找 (Binary Search) 也称折半查找，是一种时间复杂度为  $O(\log n)$  的查找算法。请完成下面的二分查找元素 target 在数组中下标的程序。

```
#include <iostream>

using namespace std;

int search( _____ ) {
    int left = 0, right = len - 1, mid;
    while( _____ ){
        mid = left + (right - left)/2;
        if(nums[mid] == target){
            return mid;
        }else if( _____ ){
            _____
        }else{
            left = mid + 1;
        }
    }
    return -1; //若未找到返回 -1
}
```

```
int main(){
    int len, target;
    int * nums;
    cout<<"请输入数组长度: "<<endl;
    cin>>len;
    nums = new int[len];
    cout<<"请输入数组元素: "<<endl;
    for(int i=0;i< len; i++)
        cin>>nums[i];
    cout<<"请输入要查找的元素: ";
    cin>>target;
```

```

        cout<<target<<"的下标为"<<search(nums, len, target)<<endl;
        return 0;
    }

```

2 将以下计算两点之间距离的程序补充完整。

```

#include <iostream>

using namespace std;

class Distance{
    double x1,y1,x2,y2;
    double dist;

public:
    Distance(int, int, int, int);
    double getdist(){ return _____; }
    void showfirstpoint() {
        cout<<x1<<" " <<y1<<endl;
    }
};

_____ {
    double x=_____;
    double y=_____;
    dist=sqrt(y*y+x*x);
}

int main(){
    Distance dist(0,0,1,1);
    cout<<"两点之间的距离是:"<<dist.getdist()<<endl;
    return 0;
}

```

3. 将下面的程序补充完整，使得程序运行结果如下：

Nemo says to Tony,"It's ok."

Nemo says to Tim,"You are too small."

```
#include <iostream>
```

```
#include <cstring>
```

```
using namespace std;
```

---

```
class Clownfish
```

```
{
```

```
    char name[10];
```

```
    int len;
```

```
public:
```

```
    Clownfish(const char *n, int l) {
```

```
        strcpy(name, n);
```

```
        len = l;
```

```
    }
```

```
    void disp(const SeaAnemone &x);
```

```
};
```

```
class SeaAnemone
```

```
{
```

---

```
    char name[10];
```

```
    int len;
```

```
public:
```

```
    SeaAnemone(const char *n, int l) {
```

```
        strcpy(name, n);
```

```
        len = l;
```

```
    }
```

```

};

void Clownfish::disp(const SeaAnemone &x)
{
    if(len <= x.len)
        cout << name << " says to " << x.name << ",\It's ok.\"" << endl;
    else
        cout << name << " says to " << x.name << ",\You are too small.\"" << endl;
}

int main()
{
    SeaAnemone sa1("Tony", 20);
    SeaAnemone sa2("Tim", 2);
    Clownfish cl("Nemo", 4);
    cl.disp(sa1);
    cl.disp(sa2);
    return 0;
}

```

#### 四. 编程题（每题 10 分，共 40 分）

1 设计一个函数 double Sin(double x, double epsilon)，使用以下无穷级数计算 sinx 的值

$$\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

要求 1) 尽量减少计算量，2) 当  $\frac{x^n}{n!}$  绝对值小于事先指定的 epsilon 时结束计算

2 设计一个支持整型、实型和字符型的冒泡排序的函数模板。

调用方式参考以下代码片段：

```
int a[10]={0,2,1,9,6,7,4,8,5,3};
```

```
char b[4]={'a','s','d','f'};
```

```
bubbleSort(a,10);
```

```
bubbleSort(b,4);
```

3、设计一个程序，先让用户输入一个整数 num，然后让用户输入 num 行句子，每行句子的长度都小于 50。已知各行句子中只包含空格以及由字母组成的单词，句子的首个字符和末尾字符都不是空格，相邻两个单词之间有且只有一个空格。要求该程序能将所有句子中最长的单词和最短的单词输出。如果有多个长度相同的最长单词或最短单词，输出最前面出现的那一个单词。部分程序代码如下，请写出函数 twoWords 的定义：

```
#include <iostream>

#include <cstring>

using namespace std;

//此处为函数 twoWords 的定义

int main() {
    int num;

    char *longest, *shortest;

    cout << "Please input the number of sentences: ";

    cin >> num; cin.get();

    cout << "Please input the sentences below: " << endl;

    twoWords(num, longest, shortest);

    cout << "The longest word is: " << longest << endl;

    cout << "The shortest word is: " << shortest << endl;

    //以下省略
}
```

运行示例：（下划线部分为用户的输入）

Please input the number of sentences: 3

Please input the sentences below:

What are you doing here

I am taking an exam

It might be an opportunity for review

The longest word is: opportunity



The shortest word is: I

4 定义一个二维数组类 Array，实现二维数组的循环上移，循环下移和输出，使用该类的 main 函数如下所示

```
#include<iostream>
```

```
using namespace std;
```

假设输入如下

3

```
int main()
```

1 2 3

```
{
```

4 5 6

```
    int n;
```

7 8 9

```
    cin>>n;
```

则输出为：

```
    Array arr(n); //构造一个 n*n 的矩阵
```

1 2 3

```
    arr.input(); //输入该矩阵的元素
```

4 5 6

```
    arr.print(); //输出该矩阵
```

7 8 9

```
    cout<<endl;
```

```
    arr.upShift(); //将该矩阵循环上移
```

4 5 6

```
    arr.print();
```

7 8 9

```
    cout<<endl;
```

1 2 3

```
    Array brr(arr);
```

```
    downShift(brr); //将矩阵循环下移
```

1 2 3

```
    brr.print();
```

4 5 6

```
    return 0;
```

7 8 9

```
}
```



