

上海交通大学试卷 (A 卷)

(2016 至 2017 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 (工科大平台)

成绩 _____

一. 选择填空题 (每题 1 分, 共 20 分):

1. 执行以下语句: `int x; char s[100]; cin>>x; cin.get(); cin>>s;` 当用户输入: 3 (回车) A SjtU (回车) 后字符串 s 中的值为:

- A、“SjtU” B、“A SjtU” C、“A” ✓ D、‘A’

2. ch 为 char 类型, 判断 ch 为数字字符的表达式是 B。

- A、'0' <= ch <= '9' B、(ch >= '0') && (ch <= '9')
C、(ch >= '0') || (ch <= '9') D、(ch >= '0') & (ch <= '9')

3. cmath 库中一个求整数绝对值的函数原型为: `int abs(int x)`, 试问以下哪种调用会报错 B。

- A、`double x=3.5; int y=abs(int(x))` B、`int x=3; int y=abs(int x)`
C、`int x=3; int y; y=abs(x);` D、`int x=3; int y=abs(x)`

4. 以下哪个是 C++ 合法的变量名 D:

- A、%store B、12_store_5 C、12_store-5 D、_store_5

5. 执行 `double x; cin>>x;` 问 y 如何能获得 x 四舍五入的结果 A:

- A、`int y=x+0.5` B、`int y=(int)x` C、`int y=x;` D、`int y=x-0.5`

6. C++ 语言的跳转语句中, 对于 break 和 continue 说法正确的是 C:

- A、break 语句只应用于循环体中
B、continue 语句用于跳出循环语句
C、continue 语句用于跳出当前的循环周期
D、break 语句用于跳出当前的循环周期

7. 执行语句 `int a[10]={3,5};` 后 a[5] 的值为: D:

- A、5 B、3 C、随机值 D、0

8. 若有声明: `char *s;` 之后运行下列语句, 则哪个语句运行时不会出现错误 C (全错)。

- A、`s="SJTU"` B、`s={'S','J','T','U','\0'};`
C、`s=strcpy(s,"SJTU");` D、`s={'S','J','T','U'};`

9. 关于函数的定义, 下面哪个说法是错误的 D。

- A、一个函数最多只能返回一个值 B、void 类型的函数也可以使用 return 语句
C、函数可以有参数也可以没有 D、函数的返回类型一定和函数的某个参数类型一致

10. 在同一工程文件中, 一个文件想要共享使用在另外一个文件中声明的一个全局变量, 则需要在本文件声明该全局变量前加限定词 B。

- A、auto B、extern C、static D、register

[键

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)				

11. 一个整型数组 b[10]要作为参数传递给函数 void f(int a[], int n), 以下哪个调用是正确的 A, D。

- A、f(b,20) B、f(b[10],10) C、f(b[0],10) D、f(b+2,5)

12. 有如下定义语句: int a[] = {1,2,3,4,5};, 则对语句 int *p=a; 正确的描述是 B。

- A、语句 int *p = a; 定义不正确
B、语句 int *p=a; 初始化变量 p, 使其指向数组对象 a 的第一个元素
C、语句 int *p=a; 是把整个数组的值存放在变量 p 中
D、语句 int *p=a; 是把 a[0] 的值赋给变量 *p

13. 已知对指针 p 的定义, 以下哪个声明可以使得 *p 不得作为赋值语句的左值使用 A, C。

- A、const int *p; B、int * const p; C、int const *p D、**int *p const**

14. 已知一个字符串数组 char *city[]={"hello","moon","glass","tomorrow"}; cout<<city[2]+1 结果为:

- A、"lass" ☒ B、'l' C、"l" D、"glass"

15. 任意一个类, 关于构造函数和析构函数, 哪个说法是正确的 A。

- A、构造函数可以有多个, 析构函数只能有 1 个
B、构造函数可以有多个, 析构函数也能有多个
C、构造函数只能有 1 个, 析构函数也只能有 1 个
D、构造函数只能有 1 个, 析构函数可以有多个

16. 以下关于运算符重载的叙述中, 正确的是 D。

- A、C++ 已有的任何运算符都可以重载
B、可以重载 C++ 中没有的运算符
C、运算符重载时可改变其优先级
D、运算符重载时可改变其实现的功能

17. 以下哪个操作符必须重载成类的友元函数 A。

- A、>> B、* C、== D、++

18. 当使用 ofstream 流定义一个流对象并打开一个磁盘文件时, 文件的隐含打开方式为 B。

- A、ios::in B、ios::out C、ios::in|ios::out D、ios::binary

19. 当用 class 来定义一个类时, 下面有关类成员的叙述中, 不正确的是 C。

- A、当不指定类成员的访问权限时则为私有成员
B、友元函数可访问类中的任一成员
C、指定为 public 的成员不允许在类外访问
D、成员函数可访问类中的任一成员

20. 下面对类的静态成员函数描述中, 正确的是 B。

- A、静态成员函数有 this 指针作为函数的隐含参数
B、静态成员函数只能访问静态数据成员
C、静态成员函数不能访问类的公有数据成员
D、静态成员函数不能访问类的私有数据成员

二. 在每小题右侧给出程序的运行结果（每题 5 分， 共 30 分）:

1. #include <iostream>

using namespace std;

int main()

{

int a[5]={1,3,5,9,2};

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

switch (a[i]+1)

{ case 4:

cout<<"first ";

break;

case 2:

cout<<"second ";

break;

case 10:

cout<<"third ";

default:

cout<<"forth ";

cout<<endl;

}

return 0;

}

second first forth
third forth
forth

2. #include <iostream>

using namespace std;

int sum1(int a) {

int d = 0;

static int b = 5;

d++;

return (a + b + d);

}

int sum2(int a) {

int c = 2;

static int b = 15;

b += 1;

return (a + b + c);

}

int main() {

int aa = 10;

cout<<sum1(aa)<<','<<sum2(aa)<<endl;

cout<<sum1(aa)<<','<<sum2(aa)<<endl;

cout<<endl;

return 0;

}

16,28

16,29

3. #include <iostream>

using namespace std;

int main() {

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

int b[3] = { 'f','a','p' };

int c[3] = { 0 };

int j;

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

j = 0;

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

{ j++; c[i]++; }

}

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

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

cout << endl;

return 0;

}

1 3 0

4. #include <iostream>

using namespace std;

int main()

{

int s = 0;

int i;

for (i = 1;; i = i + 2) {

if (s > 10)

break;

if (i % 3 == 0)

s += i;

}

cout << "i,s=" << i << ", " << s << endl;

return 0;

}

i, s = 11, 12

5. #include <iostream>

using namespace std;

class A

{ private:

int x;

public:

A(int a=8){ x=a; }

A(const A&r){ x=r.x*5; }

A& operator=(const A &r)

{ x=2*r.x;

}

x:15
x:7
15 bye
3 bye
8 bye
7 bye

```

void disp(){cout<<"x:"<<x<<endl;}
~A() { cout<<x<<" bye "<<endl;}
};
A a0=7;
int main()
{
    A a1(3);
    A a2=a1;
    static A a3;
    a2.disp(); a0.disp();
    return 0;
}

```

6. #include <iostream>

```
using namespace std;
```

```
class Tdate
```

```
{
```

```
public:
```

```
    void set(int,int,int);
```

```
    int isLeapYear();
```

```
    void print();
```

```
private:
```

```
    int month; int day; int year;
```

```
};
```

```
void Tdate::set(int m,int d,int y=2000)
```

```
{    month=m;day=d;year=y;    }
```

```
int Tdate::isLeapYear()
```

```
{    return((year%4==0&&year%100!=0)||(year%400==0));    }
```

```
void Tdate::print()
```

```
{
```

```
    cout<<year<<"/"<<month<<"/"<<day<<endl;
```

```
}
```

```
int main()
```

```
{
```

```
    Tdate s,*pTdate=&s;
```

```
    s.set(12,2,2016);
```

```
    pTdate->print();
```

```
    if((*pTdate).isLeapYear())
```

```
        cout<<"error"<<endl;
```

```
    else
```

```
        cout<<"right"<<endl;
```

```
    return 0;
```

```
}
```

2016/12/2
error

三. 程序填空题 (每空 2 分, 共 20 分):

1. #include <iostream>

#include <cstring>

using namespace std;

bool search(const char *dept[], int n, const char *s)

```
{
    for (int i=0; i<n; i++)
        if (strcmp(dept[i], s)==0)
            return true;
    return false;
}
```

int main()

```
{
    const char *deptName[4]={ "computer", "automation", "electrical engineering",
                                "instrument engineering"};

    char name[50];

    cout<<" input the deptment name you want to search : ";
    cin>>name;

    if (search(deptName, 4, name))
        cout<<"found!"<<endl;
    else
        cout<<"not found!"<<endl;
    return 0;
}
```

2. 以下程序中的函数 fun(int *a, int n)的功能是:把形参 a 所指向的数组中的最小值放在 a[0]中, 接着把形参 a 所指数组中的最大值放在 a[1]中; 把 a 所指数组中的次小值放在 a[2]中, 把 a 所指数组中的次大值放在 a[3]中, 其余依次类推, 直至将数组中数据依次处理完为止。例如: 把 a 所指向数组中的数据最初排列为:3,5,9,2,6.按以上规则移动数组中的数据后, 数据的排列为:2,9,3,6,5

void fun(int a[], int n)

```
{
    int maxOrmin, pm, t;
    for (int i = 0; i<n - 1; i++)
    {
        maxOrmin = a[i] ;
        pm = i;
        for (int j = i + 1; j<n; j++)
        {
            if (((i%2==0)&&( a[j] <= maxOrmin ) || (( i%2 == 1 )&&(a[j]>maxOrmin)))
```

```

        {    maxOrmin = a[j];
            pm = j;
        }
    }
    if (pm!=i)
    {
        t=a[i];
        a[i]=maxOrmin;
        a[pm] =t;
    }
}

void fun(int a[], int n)
{
    int maxOrmin, i, j;

    for ( i = 0; i< n-1; i++)
    {
        maxOrmin = a[i];

        for ( j = i + 1; j<n; j++)
        {
            if (((i%2==0)&&(a[j] <= maxOrmin)) || ((i%2 ==1)&&(a[j]>maxOrmin)))
            {
                maxOrmin = a[j];
                a[j] = a[i];
                a[i]= maxOrmin;
            }
        }
    }
}

```

```

3. #include <iostream>
using namespace std;
void getResult(int a[], int n, int &max, int &min, double *p_ave)
{
    double total;
    max = min = a[0];
    for (int i=0; i<n; i++)
    {
        if (a[i]>max) max = a[i];
        if (a[i]<min) min = a[i];
        total += a[i];
    }
    *p_ave = total/n;
}

int main()
{
    int a[5];
    int max, min;
    double ave;
    for (int i=0; i<5; i++)
        cin>>a[i];
    getResult(a, 5, max, min, &ave);
    cout<<ave<<" "<<max<<" "<<min<<endl;
    return 0;
}

```

[键入叉子]

四、编程题（共 30 分）

1、输入一个字符串 str, 再输入一个字符 c, 试完成函数 void delChar (char str[], char c)的实现部分, 该函数可将 str 中的字符 c 全部删除。 要求该函数用递归的方法来实现。（10 分）

```
void delChar(char str[], char c )
{
    int j = 0;
    int i = 0;

    for (i=0; str[i] != '\0'; i++ )
    {
        if (str[i] != c)
            str[j++] = str[i];
    }
    str[j] = '\0';
}
```

```
void delChar(char str[], char c )    // 递归版本
{
    int i = 0;
    int len = strlen(str) ;
    if (len== 0) return;
    else
    {
        if (str[0] == c)
        {
            for (i=0; str[i] != '\0'; i++ ) str[i] = str[i+1];
            str[i] = '\0';
            delChar(str, c);
        }
        else
            delChar(str+1, c);
    }
}
```

```
int main()
{
    char str[] = "That's too foolish!";
    cout << str << endl;
    delChar(str, 'o');
    cout << str << endl;
    return 0;
}
```

四（2）设计一个商品类 `goods`，包括商品名称、进价、售价、个数，利用重载运算符”+“将 2 个商品的进价总额、售价总额、个数相加放在一个对象中，再对该对象求每件商品的平均进价、售价。当运行下面测试程序时，能得到相应的执行结果，这里价格和平均价格都要求为实数，个数为整数。（20 分）

```
int main()
{
    double inPrice, outPrice;
    goods g1("desk", 128, 150, 10), g2("chair", 25, 30, 10), g;
    cout << "输出结果" << endl;
    cout << g1 << endl;
    g = g1 + g2;           // 调用重载运算符
    avg(g, inPrice, outPrice); // 友元函数求平均价格
    cout << "平均进价: " << inPrice << endl;
    cout << "平均售价: " << outPrice << endl;
    return 0;
}
```

本测试程序的执行结果如下：

输出结果：

desk 128 150 10

平均进价: 76.5

平均售价: 90

```
class goods {
    string name;
    double inPrice;
    double outPrice;
    int num;
public:
    goods(string s, double inp, double outp, int n);
    goods();

    goods& operator+(const goods& g);

    friend ostream& operator<<(ostream& os, const goods& g);
    friend void avg(const goods g, double& inp, double& outp);
};
```

[键入叉子]

```

goods::goods(string s, double inp, double outp, int n)
{
    name = s;
    inPrice = inp;
    outPrice = outp;
    num = n;
}

goods::goods()
{
    name = "";
    inPrice = 0;
    outPrice = 0;
    num = 0;
}

goods& goods::operator+(const goods& g)
{
    inPrice += g.inPrice;
    outPrice += g.outPrice;
    num += g.num;

    return *this;
}

ostream& operator<<(ostream& os, const goods& g)
{
    os << g.name << " " << g.inPrice << " " << g.outPrice << " " << g.num << endl;
    return os;
}

void avg(const goods g, double& inp, double& outp)
{
    inp = g.inPrice / g.num;
    outp = g.outPrice / g.num;
}

```