

每日一题day16_6月6日

一. 单选

1.

```
void swap_int(int *a,int *b){
    *a=*a+*b;
    *b=*a-*b;
    *a=*a-*b;
}
```

以下说法正确的是：

- ☐ A 结果不正确，因为会溢出，用位与的方式就没问题
- ☐ B 结果正确，即使会溢出
- ☐ C 结果正确，不会溢出
- ☐ D 其他选项都不对

正确答案：B

2. 若有定义int (*pt) [3]；则下列说法正确的是：

- ☐ A 定义了基类型为int的三个指针变量
- ☐ B 定义了基类型为int的具有三个元素的指针数组pt
- ☐ C 定义了一个名为*pt、具有三个元素的整型数组
- ☐ D 定义了一个名为pt的指针变量，它可以指向每行有三个整数元素的二维数组

正确答案：D

3. 对于下面的说法，正确的是_____。

- ☐ A 对于 struct X { short s; int i; char c; }，sizeof(X) 的值等于 sizeof(s) + sizeof(i) + sizeof(c)
- ☐ B 对于某个double变量 a，可以使用 a == 0.0 来判断其是否为零
- ☐ C 初始化方式 char a[14] = "Hello, world!"; 和初始化方式 char a[14]; a = "Hello, world!"; 的效果相同
- ☐ D 在gcc编译器下，对于 int i = 3; printf("%d %d", ++i, ++i)，运行输出为：4 5
- ☐ E 选项A、B、C、D中至少有两个是正确的
- ☐ F 以上选项均不正确

正确答案：F

4. 下列代码的输出是？（注：print已经声明过）

```
main(){
    char str[]="Geneius";
    print (str);
}
print(char *s){
    if(*s){
        print(++s);
        printf("%c",*s);
    }
}
```

- ☒ A suiene
- ☐ B neius
- ☐ C run-time error
- ☐ D suieneG

正确答案：A

5. 写出下面程序的输出结果

```
class A
{
public:
    void FuncA()
    {
        printf( "FuncA called\n" );
    }
    virtual void FuncB()
    {
        printf( "FuncB called\n" );
    }
};
class B : public A
{
public:
    void FuncA()
    {
        A::FuncA();
        printf( "FuncAB called\n" );
    }
    virtual void FuncB()
    {
        printf( "FuncBB called\n" );
    }
};
void main( void )
```

```
{
    B b;
    A *pa;
    pa = &b;
    A *pa2 = new A;
    pa->FuncA(); ( 3 )
    pa->FuncB(); ( 4 )
    pa2->FuncA(); ( 5 )
    pa2->FuncB();
    delete pa2;
}
```

- ☐ A FuncA called FuncB called FuncA called FuncB called
- ☐ B FuncA called FuncBB called FuncA called FuncB called
- ☐ C FuncA called FuncBB called FuncAB called FuncBB called
- ☐ D FuncAB called FuncBB called FuncA called FuncB called

正确答案：B

6. 以下程序输出是_____。

```
#include <iostream>
using namespace std;
int main(void)
{
    const int a = 10;
    int * p = (int *)(&a);
    *p = 20;
    cout<<"a = "<<a<<" , *p = "<<*p<<endl;
    return 0;
}
```

- ☐ A 编译阶段报错运行阶段报错
- ☐ B a = 10, *p = 10
- ☐ C a = 20, *p = 20
- ☐ D a = 10, *p = 20
- ☐ E a = 20, *p = 10

正确答案：D

7. 以下关于STL的描述中，_____是错的。

- ☐ A STL容器是线程不安全的
- ☐ B 当容量不够时，vector内部内存扩展方式是翻倍

- ☐ C std::sort是稳定排序
- ☐ D std::bitset不是一个STL容器
- ☐ E std::stack默认是用deque实现的
- ☐ F std::string中可以存储多个'\0'字符

正确答案：C

8.

以下代码共调用多少次拷贝构造函数：

```
Widget f(Widget u)
{
    Widget v(u);
    Widget w=v;
    return w;
}
main(){
    Widget x;
    Widget y=f(f(x));
}
```

- ☐ A 1
- ☐ B 3
- ☐ C 5
- ☐ D 7

正确答案：D

9.

以下代码有什么问题？

```
struct Test
{
    Test( int ) {}
    Test() {}
    void fun() {}
};
void main( void )
{
    Test a(1);
    a.fun();
    Test b();
    b.fun();
}
```

- A b.fun () 会出错
- B Test结构的定义中应该加上public修饰符，这样才能main函数中调用改类的方法
- C Test(int){} 应该改成Test(int a){}
- D 以上说法都不正确

正确答案：A

10.

```
#include<iostream>
using namespace std;

class Base
{
public:
    virtual int foo(int x)
    {
        return x * 10;
    }

    int foo(char x[14])
    {
        return sizeof(x) + 10;
    }
};

class Derived: public Base
{
    int foo(int x)
    {
        return x * 20;
    }

    virtual int foo(char x[10])
    {
        return sizeof(x) + 20;
    }
};

int main()
{
    Derived stDerived;
    Base *pstBase = &stDerived;

    char x[10];
    printf("%d\n", pstBase->foo(100) + pstBase->foo(x));
}
```

```
return 0;
}
```

在32位环境下，以上程序的输出结果是？

- ☒ A 2000
- ☐ B 2004
- ☐ C 2014
- ☐ D 2024

正确答案：C

二. 编程

1. 标题：iNOC产品部--完全数计算 | 时间限制：1秒 | 内存限制：32768K

完全数（Perfect number），又称完美数或完备数，是一些特殊的自然数。

它所有的真因子（即除了自身以外的约数）的和（即因子函数），恰好等于它本身。

例如：28，它有约数1、2、4、7、14、28，除去它本身28外，其余5个数相加， $1+2+4+7+14=28$ 。

给定函数count(int n)，用于计算n以内(含n)完全数的个数。计算范围, $0 < n \leq 500000$

返回n以内完全数的个数。

异常情况返回-1

```
/**
```

```
*
```

```
*
```

完全数 (Perfect number) , 又称完美数或完备数, 是一些特殊的自然数。

*

它所有的真因子 (即除了自身以外的约数) 的和 (即因子函数) , 恰好等于它本身。

*

例如 : 28 , 它有约数1、 2、 4、 7、 14、 28 , 除去它本身28外, 其余5个数相加, $1+2+4+7+14=28$ 。

*

*

给定函数count(int n),用于计算n以内(含n)完全数的个数

* @param n

计算范围, $0 < n \leq 500000$

* @return n

以内完全数的个数, 异常情况返回-1

```
*
```

```
*/
```

```
public
```

```
static
```

```
int
```

```
count(
```

```
int
```

```
n)
```

输入描述：

输入一个数字

输出描述：

输出完全数的个数

示例1:

输入

1000

输出

3

正确答案：

2. 标题：扑克牌大小 | 时间限制：1秒 | 内存限制：32768K

style="">扑克牌游戏大家应该都比较熟悉了，一副牌由54张组成，含3~A、2各4张，小王1张，大王1张。牌面从小到大用如下字符和字符串表示（其中，小写joker表示小王，大写JOKER表示大王）：

3 4 5 6 7 8 9 10 J Q K A 2 joker JOKER

输入两手牌，两手牌之间用"-"连接，每手牌的每张牌以空格分隔，"-"两边没有空格，如：4 4 4 4-joker JOKER。

请比较两手牌大小，输出较大的牌，如果不存在比较关系则输出ERROR。

基本规则：

- (1) 输入每手牌可能是个子、对子、顺子（连续5张）、三个、炸弹（四个）和对王中的一种，不存在其他情况，由输入保证两手牌都是合法的，顺子已经从小到大排列；
- (2) 除了炸弹和对王可以和所有牌比较之外，其他类型的牌只能跟相同类型的存在比较关系（如，对子跟对子比较，三个跟三个比较），不考虑拆牌情况（如：将对子拆分成个子）；
- (3) 大小规则跟大家平时了解的常见规则相同，个子、对子、三个比较牌面大小；顺子比较最小牌大小；炸弹大于前面所有的牌，炸弹之间比较牌面大小；对王是最大的牌；
- (4) 输入的两手牌不会出现相等的情况。

输入描述：

输入两手牌，两手牌之间用"-"连接，每手牌的每张牌以空格分隔，"-"两边没有空格，如 4 4 4 4-joker JOKER。

输出描述：

输出两手牌中较大的那手，不含连接符，扑克牌顺序不变，仍以空格隔开；如果不存在比较关系则输出ERROR。

示例1:

输入

4 4 4 4-joker JOKER

输出

joker JOKER

正确答案：