

在软件开发中，我们经常要使用测试用例对程序进行检查，一次成功的测试是指运行测试用例后（）

正确答案: B 你的答案: 空 (错误)

证明程序正确性

发现了程序错误

改正了程序错误

未发现程序错误

下列 main（）函数执行后的结果是（）

```
int fun(int value)
{
    int cnt = 0;
    while (value)
    {
        cnt++;
        value = value & (value - 1);
    }
    return cnt;
}
void main(void)
{
    printf("%d", fun(65535));
}
```

正确答案: C 你的答案: 空 (错误)

17

15

16

1

下列代码之后的结果为（）

职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使

```
#include <iostream>
using namespace std;
struct Base
{
    int i;
    virtual int f()
    {
        cout << "a";
        return 1;
    }
    virtual const Base &f() const
    {
        cout << "b";
        return *this;
    }
    int g()
    {
        cout << "c";
        return 3;
    }
};
```

```
struct Derive: Base
{
    int i;
    int f()
    {
        cout << "d";
        return 4;
    }
    const Base &f() const
    {
        cout << "e";
        return *this;
    }
    int f(int = 0)
    {
        cout << "f";
        return 6;
    }
};
```

```
    }  
    virtual int g()  
    {  
        cout << "g";  
        return 7;  
    }  
};  
int main()  
{  
    Derive d;  
    const Derive d_const;  
    Base b, *p = &d;  
    const Base *p_const = &d_const;  
    b.f();  
    p->f();  
    p->g();  
    p_const->f();  
    d_const.f();  
}
```

正确答案: D 你的答案: 空 (错误)

abccf  
acdff  
abcde  
adcee

单例模式中, 两个基本要点 ( ) 和单子类自己提供单例

正确答案: A D 你的答案: 空 (错误)

构造函数私有  
静态工厂方法  
以上都不对  
唯一实例

为了查看不断更新的日志文件, 可以使用的指令是 ( )

正确答案: D 你的答案: 空 (错误)

cat -n  
vi  
more  
tail -f

在下面的类定义中, 横线上应填入的内容是 ( )

```
class Fred
{public:
    void print(){cout<<DATA<<endl;}
    void setData(double d){data=d;}
    static int count;
private:
    double data;};
    _____ count=0;
```

正确答案: B 你的答案: 空 (错误)

```
static int Fred: :
int Fred: :
int
static int
```

以下属于稳定排序的是 ( )

正确答案: A D 你的答案: 空 (错误)

归并排序和基数排序

快速排序和堆排序

选择排序和希尔排序

插入排序和冒泡排序

关于 `mysql_fetch_object` 说法中正确的是 ( )

正确答案: B 你的答案: 空 (错误)

`mysql_fetch_object` 返回值是个对象，在速度上和 `mysql_fetch_array` 及 `mysql_fetch_row` 差不多

`mysql_fetch_object` 返回值是个对象，所以在速度上比 `mysql_fetch_array` 要慢

`mysql_fetch_object` 返回值是个数组，所以在速度上和 `mysql_fetch_array` 及 `mysql_fetch_row` 差不多

`mysql_fetch_object` 和 `mysql_fetch_array` 一样，没什么区别

下列计算机不属于 RISC 计算机的是 ( )

正确答案: B 你的答案: 空 (错误)

IBM:Power PC620

IBM:PC/XT

SUN:Ultra SPARC

DEC:ALPHA21164

下列程序执行后的输出结果为 ( )

```
#include<stdio.h>
#include<stdlib.h>
#define sum(a,b,c) a+b+c
int main()
{
    int i = 3,j=2;
    printf("%d\n",i * sum(i,(i+j),j));
    return 0;
}
```

正确答案: C 你的答案: 空 (错误)

36

以上均不正确

16

30

以下意图用来描述 FACTORY METHOD(工厂方法)的是 ( )

正确答案: C 你的答案: 空 (错误)

表示作用于某对象结构中的各元素的操作，它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作

提供一个创建一系列相关或相互依赖对象的接口，而无需指定他们具体的类

定义一个用于创建对象的接口，让子类决定实例化哪一个类，该模式使一个类的实例化延迟到其子类

定义一系列的算法，把它们一个个封装起来，并且使它们可相互替换，本模式使得算法可独立于使用它的客户而变化

类 CBase 的定义如下： 在构造函数 CDerive 的下列定义中，正确的是 ( )

```
class CBase
{
    int x;
    public:
    CBase(int n){x=n;}
};
class CDerive:public CBase
{
    CBase y;
    int z
    public:
    CDerive(int a,int b,int c);
};
```

正确答案: B 你的答案: 空 (错误)

```
CDerive:CDerive(int a, int b, int c): x(a),y(b).z(c){}
CDerive:CDerive(int a, int b, int c): CBase(a),y(b).z(c){}
CDerive:CDerive(int a, int b, int c): CBase(a),CDerive(b).z(c){}
CDerive:CDerive(int a, int b, int c): x(a),CBase(b).z(c){}
```

设有以下定义程序;则以下合法的调用语句是 ( )

```
#include
class A1
{public:
    void show1()
    {cout << "class A1" << endl;};
class A2:public A1
{ public:
    void show2()
    {cout << "class A2" << endl;};
class A3:protected A2
{public:
    void show3()
    {cout << "class A1" << endl; };
void main()
{A1 obj1; A2 obj2; A3 obj3;}
```

正确答案: A B C 你的答案: 空 (错误)

```
obj3.show3();
obj2.show2();
obj1.show1();
obj3.show1();
```

在通常的语法分析中, ( ) 特别使用于表达式的分析

正确答案: B 你的答案: 空 (错误)

LL (1) 分析法  
算符优先分析法  
递归下降分析法  
LR 分析法

下推自动识别机的语言是 ( )

正确答案: C 你的答案: 空 (错误)

1 型语言  
3 型语言  
2 型语言  
0 型语言

大整数 845678992357836701 转译为 16 进制的表示, 最后两位字符是 ( )

正确答案: B 你的答案: 空 (错误)

AB  
9D

8B

EF

存在 int 类型变量 x, y, z, 其对应值为  $x=0x59$ ,  $y=0x39$ ,  $z=0x6E$ , 则  $x*y+z$  的值为 ( )

正确答案: B 你的答案: 空 (错误)

`1001001110011

`1010000111111

`1001001110111

`1011001111111

关于 RSA 加密过程和解密过程说法正确的是 ( )

正确答案: B C 你的答案: 空 (错误)

加密过程中, 使用私钥对信息进行加密

加密过程中, 使用公钥对信息进行加密

解密过程中, 使用私钥对信息进行解密

解密过程中, 使用公钥对信息进行解密

一个局域网中某台主机的 ip 地址为 176.68.160.12, 使用 22 位作为网络地址, 最多可以连接的主机数为 ( )

正确答案: D 你的答案: 空 (错误)

254

1024

512

1022

675、225、90、45、30、30、( )

正确答案: A 你的答案: 空 (错误)

60

124

38

27

问号中应填写的数字是 ( )

12	9	-6
2	3	10
1	3	?

正确答案: A 你的答案: 空 (错误)

- 11
- 26
- 17
- 13

定义新运算：对于任意自然数 A，B，若 A，B 奇偶性相同，则  $A*B=(A+B)/2$ ；若 A、B 奇偶性不同，则  $A*B=(A+B+1)/2$ ，那么  $1*2*5=()$ （“\*”代表是一个符号，不是乘积的意思。）

正确答案: A 你的答案: 空 (错误)

- 4
- 8
- 12
- 2

R浩B

☆

△

?

问号处应当是：

A

□

○

B

□

≡

C

□

△△

D

□

⊖

正确答案: D 你的答案: 空 (错误)

如图  
如图  
如图

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪offer。版权所有，违者必究。



如图

在某次税务检查后，四个工商管理人员各自做出了结论。甲说：“所有个体户都没纳税。”。乙说：“服装个体户陈老板没纳税。”丙说：“个体户不都没税”，丁说：“有的个体户没税”。如果四个人中只有一个断定属实，那么以下哪项是真的？（）

正确答案: D 你的答案: 空 (错误)

- 丁断定属实，但陈老板交了税
- 甲断定属实，陈老板没纳税
- 乙断定属实，陈老板没纳税
- 丙断定属实，但陈老板交了税

看下列资料描述，回答 1-4 题

根据统计，1985 年北京市建筑企业共完成生产总值 47.2 亿元，比上年增长 31.5%（扣除价格因素的影响，实际增长 22.2%），其中，中央在京施工企业完成 8.2 亿元，比上年增长 28%，地方全民所有制工企完成 24.1 亿元，比上年增长 24.4%，特别是地方集体所有制施工企业发展更快，完成总产值 14.9 亿元，比上年增长 47.3%，全市建筑企业按总产值计算的全员劳动生产率为 7743 元，比上年提高 13.4%（扣除价格因素影响则提高 6%），全员平均产值达到万元以上的企业，由上年的 20 个增加到 55 个。市建筑工程总公司所属二公司，四公司 and 设备安装公司，门头沟区建筑公司，市水利工程二处，水利机械施工等 12 个企业的全员劳动生产超过 15000 元，扣除价格因素影响，

1、北京市 1985 年建筑企业生产总值比 1984 年实际增长多少？

- A 22.2%
- B 24.4%
- C 31.5%
- D 28%

2、与 1984 年相比，1985 年北京市哪种类型的施工企业其产值增长速度最快？（）

- A 地方全民所有制企业
- B 无法确定
- C 地方集体所有制施工企业
- D 中央在京施工企业

3、1985 年，全员平均产值达到万元以上的施工企业比 1984 年增加了多少个？（）

- A 20
- B 55
- C 35
- D 12

4、北京市建筑工程总公司所属公司 1985 年全员劳动生产率为（）。

- A 10000 元
  - B 7743 元
  - C 接近 15000 元
  - D 15000 元以上
- ACCC

**题目描述：**英文单词有很多非常相似，比如：**see** 和 **seek**、**cat** 和 **cut** 等，现在提供 3 种编辑操作：**insert**、**remove**、**replace**，通过在单词 1 上进行这些操作，可以让单词 1 变成单词 2

那么问题来了，如何只用最小次数的编辑操作，可以让字符串 1 变成字符串 2？

说明：

- 1) 3 种编辑操作的代价是一样的
- 2) 并且每次只能操作一个字符串的一个字母
- 3) 只需要考虑在字符串 1 上进行编辑操作即可

### 输入

输入一行, 有两个字符串, 以空格分隔。

### 输出

输出为最小编辑次数。

### 样例输入

geek gesek

### 样例输出

1

# coding:utf8

```
def count_opera():
```

```
    hey = raw_input()
```

```
    while (not hey or len(hey.split(' '))!=2):
```

```
        print "Input Correct String"
```

```
        hey = raw_input()
```

```
    need_change, porpose = hey.split(' ')
```

```
    need_change_length = len(need_change)
```

```
    porpose_length = len(porpose)
```

```
    if need_change_length == porpose_length:
```

```
        print same_length(need_change, porpose)
```

```
    if need_change_length < porpose_length:
```

```
        print need_insert(need_change, porpose)
```

```
    if need_change_length > porpose_length:
```

```
        print need_delete(need_change, porpose)
```

```
def same_length(string_A, string_B):
```

```
    count = 0
```

```
    for index in xrange(len(string_B)):
```

```
        if string_A[index] != string_B[index]:
```

```
            count += 1
```

```
    return count
```

```
def need_insert(string_A, string_B):
```

```
    count = 0
```

```
    a_list = list(string_A)
```

```
    for index in xrange(len(string_B)):
```

```
        if a_list[index] != string_B[index]:
```

```
            a_list.insert(index, string_B[index])
```

```
            count += 1
```

```
    if len(a_list) == len(string_B):
```

```
        string_A = ''.join(a_list)
```

```
        break
    change_cnt = same_length(string_A, string_B)
    count += change_cnt
    return count

def need_delete(string_A, string_B):
    count = 0
    a_list = list(string_A)
    while (len(a_list) != len(string_B)):
        a_list = delete_and_judge(a_list, string_B)
        count += 1
    string_A = ''.join(a_list)
    change_cnt = same_length(string_A, string_B)
    count += change_cnt
    return count

def delete_and_judge(a_list, string_B):
    length = len(a_list)
    for index in xrange(length):
        if a_list[index] != string_B[index]:
            del a_list[index]
    return a_list

if __name__ == "__main__":
    count_opera()
```