

### 一、百度(武汉地区)

#### 第一部分：

- 1、描述数据库的简单操作。
- 2、描述 TCP\IP 四层模型，并简述之。
- 3、描述 MVC 的内容。

#### 第二部分：

- 1、给出 a-z0-9，在其中选择三个字符组成一个密码，输出全部的情况，程序实现。
- 2、字符串的反转，比如 abcde，输出 edcba。

3、许多程序会大量使用字符串。对于不同的字符串，我们希望能够有办法判断其相似程序。我们定义一套操作方法来把两个不相同的字符串变得相同，具体的操作方法为：

- 1.修改一个字符（如把“a”替换为“b”）；
- 2.增加一个字符（如把“abdd”变为“aebdd”）；
- 3.删除一个字符（如把“travelling”变为“traveling”）；

比如，对于“abcdefg”和“abcdef”两个字符串来说，我们认为可以通过增加/减少一个“g”的方式来达到目的。上面的两种方案，都仅需要一次。把这个操作所需要的次数定义为两个字符串的距离，而相似度等于“距离+1”的倒数。也就是说，“abcdefg”和“abcdef”的距离为 1，相似度为  $1/2=0.5$ 。

给定任意两个字符串，你是否能写出一个算法来计算它们的相似度呢？

#### 第三部分：

一般上网买东西，都会用得到图 1 设备里面生成的 6 位动态密码（每一分钟可以刷新一次），然后输入到网页上的相应文本框里面，问服务器怎么验证用户输入的 6 位密码是正确的？如果系统升级，怎么去维护？如果系统用户比较多，又该如何无

设计？



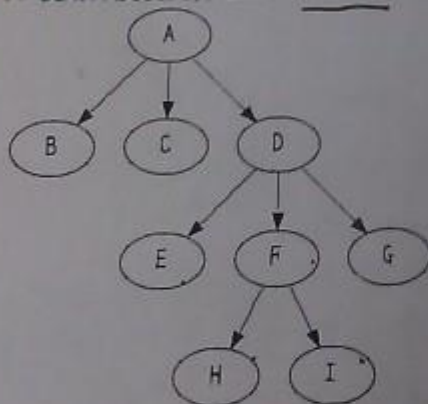
## 二、腾讯

## 腾讯 2013 校园招聘笔试试题

试卷类型：软件开发 T1  
考试时长：120 分钟

一、不定项选择题（共 20 题，每题 3 分，共 60 分，少选、错选、多选不得分）

- 以下表的设计，最合理的是（）  
 A. 学生：(id, name, age), 学科：(id, name), 分数：(学生 id, 学科 id, 分数)  
 B. 学生：(id, name, age), 分数：(学生 id, 学科名称, 分数)  
 C. 分数：(学生姓名, 学科名称, 分数)  
 D. 学科：(id, name), 分数：(学生姓名, 学科 id, 分数)
- 在数据库系统中，产生不一致的根本原因是（）  
 A. 数据存量太大 B. 没有严格保护数据 C. 未对数据进行完整性控制 D. 数据冗余
- 用容积分别为 15 升和 27 升的两个杯子向一个水桶中装水，可以精确向水桶中注入（）升水？  
 A. 53 B. 25 C. 33 D. 52
- 考虑左递归文法  $S \rightarrow Aa|b$   $A \rightarrow Ac|Sd|e$ , 消除左递归后应该为（）？  
 A.  $S \rightarrow Aa|b$   $A \rightarrow bdA'|A'$   
 $A' \rightarrow cA'|adA'|\epsilon$   
 B.  $S \rightarrow Ab|a$   $A \rightarrow bdA'|A'$   
 $A' \rightarrow cA'|adA'|\epsilon$   
 C.  $S \rightarrow Aa|b$   $A \rightarrow cdA'|A'$   
 $A' \rightarrow bA'|adA'|\epsilon$   
 D.  $S \rightarrow Aa|b$   $A \rightarrow bdA'|A'$   
 $A' \rightarrow caA'|dA'|\epsilon$
- 下面的排序算法中，初始数据集的排列顺序对算法的性能无影响的是（）  
 A. 插入排序 B. 堆排序 C. 冒泡排序 D. 快速排序
- 使用二分查找算法在一个有序序列中查找一个元素的时间复杂度为（）  
 A.  $O(N)$  B.  $O(\log N)$  C.  $O(N*N)$  D.  $O(N*\log N)$
- 路由器工作在网络模型中的哪一层（）？  
 A. 数据链路层 B. 物理层 C. 网络层 D. 应用层
- 对于满足 SQL92 标准的 SQL 语句：SELECT foo, count(foo) FROM pokes WHERE foo>10 GROUP BY foo HAVING ORDER BY foo, 其执行的顺序应该为（）  
 A. FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY  
 B. FROM -> GROUP BY -> WHERE -> HAVING -> SELECT -> ORDER BY  
 C. FROM -> WHERE -> GROUP BY -> HAVING -> ORDER BY -> SELECT  
 D. FROM -> WHERE -> ORDER BY -> GROUP BY -> HAVING -> SELECT
- 使用深度优先算法遍历下面的图，遍历的顺序为（）



- A. ABCDEFGHI、 B. BCEHIFGDA CABCEFHIGD D. HIFEGBCDA

- 在 UNIX 系统中，目录结构采用（）  
 A. 单级目录结构 B. 二级目录结构 C. 单纯树形目录结构 D. 带链接树形目录结构

11. 题目：请问下面的程序一共输出多少个“-"？

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
```

```
{
    int i;
    for(i=0; i<2; i++){
        fork();
        printf("-");
    }
    return 0;
}
```

(A) 2 个      B. 4 个      C. 6 个      D. 8 个

12. 题目：请问下面的程序一共输出多少个“-"？

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
```

```
int main(void)
{
    int i;
    for(i=0; i<2; i++){
        fork();
        printf("\n");
    }
    return 0;
}
```

A. 2 个      (B) 4 个      C. 6 个      D. 8 个

13. 避免死锁的一个著名的算法是 ( )

A. 先入先出法      (B) 银行家算法      C. 优先级算法      D. 资源按序分配法

14. 你怎么理解的分配延迟(dispatch latency)

A. 分配器停止一个进程到开启另一个进程的时间      (B) 处理器将一个文件写入磁盘的时间  
C. 所有处理器占用的时间      D. 以上都不对

15. 以下那一个不是进程的基本状态 ( )

A. 阻塞态      B. 执行态      C. 就绪态      (D) 完成态

16. 假定我们有 3 个程序，每个程序花费 80%的时间进行 I/O, 20%的时间使用 CPU。每个程序启动时间和其需要使用进行计算的分钟数如下，不考虑进程切换时间：

程序编号	启动时间	需要 CPU 时间 (分钟)
1	00:00	3.5
2	00:10	2
3	00:15	1.5

请问，在多线程/进程环境下，系统的总响应时间为 ( )

A. 22.5      B. 23.5      C. 24.5      D. 25.5

17. 在所有非抢占 CPU 调度算法中，系统平均响应时间最优的是 ( )

A. 实时调度算法      (B) 短任务优先算法      C. 时间片轮转算法      D. 先来先服务算法

18. 什么是内存抖动(Thrashing) ( )

(A) 非常频繁的换页活动      B. 非常高的 CPU 执行活动      C. 一个极长的执行进程      D. 一个极大的虚拟内存

19. Belady's Anomaly 出现在哪里 ( )

A. 内存管理算法      (B) 内存换页算法      C. 预防锁死算法      D. 磁盘调度算法

20. 以下的生产者消费者程序中，那个不会出现锁死，并且开销最少？

注：  
down()

- 1 判断信号量的取值是否大于等于 1
  - 2 如果是，将信号量的值减去 1，继续向下执行
  - 3 否则，在该信号量上等待（进程被挂起）
- up()  
1 将信号量的值增加 1（此操作将唤醒一个在信号量上面等待的进程）  
2 线程继续往下执行  
down()和 up()是一组原子操作

选项：

A:

```
#define N 100 //定义缓冲区大小
typedef int semaphore; //定义信号量类型
semaphore mutex = 1; //互斥信号量
semaphore empty = N; //缓冲区计数信号量，用来计数缓冲区里的空位数量
semaphore full = 0; //缓冲区计数信号量，用来计数缓冲区里的商品数量
void producer(void)
{
    int item;
    while(TRUE) {
        item = produce_item();
        down(&empty);
        down(&mutex);
        insert_item(item);
        up(&mutex);
        up(&full);
    }
}
void consumer(void)
{
    int item;
    while(TRUE){
        down(&full);
        down(&mutex);
        item = remove_item();
        up(&mutex);
        up(&empty);
        consume_item(item);
    }
}
```

B:

```
#define N 100
typedef int semaphore;
semaphore empty = N;
semaphore full = 0;
void producer(void)
{
    int item;
    while(TRUE) {
        item = produce_item();
        down(&empty);
        insert_item(item);
        up(&full);
    }
}
void consumer(void)
{
    int item;
    while(TRUE){
        down(&full);
        item = remove_item();
        up(&empty);
        consume_item(item);
    }
}
```



```
}
c
#define N 100
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;
void producer(void)
{
    int item;
    while(TRUE) {
        item = produce_item();
        down(&empty);
        down(&mutex);
        insert_item(item);
        up(&mutex);
        up(&full);
    }
}
void consumer(void)
{
    int item;
    while(TRUE){
        down(&mutex);
        down(&full);
        item = remove_item();
        up(&mutex);
        up(&empty);
        consume_item(item);
    }
}
```

```
D:
#define N 100
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;
void producer(void)
{
    int item;
    while(TRUE) {
        item = produce_item();
        down(&empty);
        down(&mutex);
        insert_item(item);
        up(&full);
        up(&mutex);
    }
}
void consumer(void)
{
    int item;
    while(TRUE){
        down(&full);
        down(&mutex);
        item = remove_item();
        up(&empty);
        up(&mutex);
        consume_item(item);
    }
}
```

二、填空题 (共 7 题 10 空, 每空 4 分, 共 40 分)

21. 将下图进行拓扑排序后, 对应的序列为 111 ABCFD



22. 下面的函数使用二分查找算法, 对已按升序排序的数组返回所要查找数值的数据位置, 请填写缺少的两行语句:

```

int *BinarySearch(int* arrayAddress, int arrayLength, int valueToSearch)
{
    int head = 0;
    int tail = arrayLength - 1;
    int mid;
    while(head < tail)
    {
        mid = (head + tail) / 2;
        if(arrayAddress[mid] > valueToSearch)
            tail = mid - 1
        else
            head = mid + 1
    }
    if(tail < arrayLength && arrayAddress[tail] == valueToSearch)
        return &arrayAddress[tail];
    else
        return NULL;
}

```

23. 一个有 N 个整数元素的一维数组  $A[0], A[1], \dots, A[N-1]$ , 求连续子数组和的最大值。

int max(int a, int b); //return the maximum number between a and b

int MaxSum(int \*A, int length){

int nStart = A[0];

int nAll = A[0];

for(int i=1; i<length; i++){

4

5

}

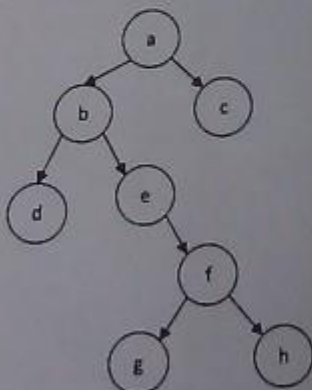
return nAll;

}

$nAll = \max(nAll, A[i])$

$nAll = \max(nAll, 0)$

24. 请给出二叉树:



abdefghc

的前序遍历: ca b d e f g h c

25. 最长递增子序列(LIS)表示在一个序列中, 保持递增的最长子序列。比如:

{2,1,4,2,3,7,4,6}的 LIS 是 {1,2,3,4,6}, 则 LIS 的长度是 5。

对于一个有  $N$  个元素的序列, 得到 LIS 的长度的最优时间复杂度是  $O(n \log n)$  空间复杂度是  $O(1)$

26. 给一系列的数  $1, 2, 3, \dots, n$  (有序的) 和一个栈(stack)。这个栈无限大。将这  $n$  个数按照顺序放入栈中。但是随机的从栈中弹出。比如:  $n=2$  时, 输出序列有可能是 1,2 或者 2,1, 所以我们得到 2 中可能的出栈序列。假设  $n=5$ , 请问有多少种出栈序列 121

27. 请给出表达式  $a+b*(c-d)/e-f$  的逆波兰式:  $a\ b\ c\ d\ -\ * \ e\ / \ f\ -$

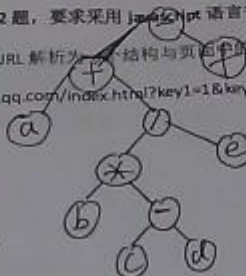
三、Web 前端方向附加题 (共 2 题, 要求采用 JavaScript 语言答题, 作为面试参考, 不计入总分)

1. 编写一个 JavaScript 函数把 URL 解析为一个结构与页面中的 javascript:location 的对象相似的对象实体

如: 输入 url 串为 "http://www.qq.com/index.html?key1=1&key2=2&key3=3",

最后输出的对象是:

```
{
  'protocol': 'http',
  'hostname': 'www.qq.com',
  'pathname': '/index.html',
  'query': 'key1=1&key2=2&key3=3'
}
```



(注: 如果熟悉 URL 规范, 对于更复杂的 URL, 可以尝试让解析输出的对象有更多便于使用的属性字段)

2. 写一个函数对字符串数组进行排序, 排序的规则是根据每个字符串中重复出现次数最多的字符出现的次数, 在次数相同的情况下根据出现次数第二多的字符排序,

比如  
"abcaba" 中重复出现次数最多的字符是 a, 次数是 3, 第二多的字符是 b, 次数是 2, 第三是 c, 次数是 1

因此 mySort(["abcaba", "asdfasdf", "asdfasdfasdf"])的结果是:

["asdfasdfasdf", "abcaba", "asdfasdf"]

四、其他方向附加题 (共 2 题, 每题 20 分, 选做 1 题, 不计入总分)

1. 微博广告投放是腾讯收入来源之一。为了保证投放的广告对用户更有帮助, 必须分析用户对什么最感兴趣。用户的每条微博都可以拆分成几个关键词, 腾讯微博每个月都会收集到上 T(Terabyte)的关键词, 请你分析出其中出现次数最多的十个关键词。

2. 腾讯新闻首页改版之后, 为了精确掌握改版效果, 需要准实时统计访问每篇文章的 IP 数量, 即从文章发表之后, 有多少个不同 IP 的用户读过这篇文章。每个用户访问请求都会被 web 服务器解析, 并实时传输到后台统计系统, 请你设计该“后台统计系统”, 以完成统计。



上面的题目答案来自网友，不一定准确。

### 三、欢聚时代

## 欢聚时代 2013 校园招聘笔试题 C++ A 卷

答题时间 120 分钟

提示：请合理分配答题时间，答案请写到答题纸上。

### 一、选择题 (20 分)

1. For automatic objects, constructors and destructors are called each time the objects \_\_\_\_\_. (1 分)  
☒ A. enter and leave scope      B. inherit parent class  
C. are constructed      D. are destroyed
2. Which of the following statement is correct regarding destructor of base class? (1 分)  
A. Destructor of base class should always be static.  
☒ B. Destructor of base class should always be virtual.  
C. Destructor of base class should not be virtual. ✗  
D. Destructor of base class should always be private. ✗
3. How can we make a class abstract? (1 分)  
A. By making all member functions constant.  
☒ B. By making at least one member function as pure virtual function.  
C. By declaring it abstract using the static keyword.  
D. By declaring it abstract using the virtual keyword.
4. Which of the following statements is correct when a class is inherited publicly? (1 分)  
A. Public members of the base class become protected members of derived class. ✗  
B. Public members of the base class become private members of derived class. ✗  
C. Private members of the base class become protected members of derived class.  
☒ D. Public members of the base class become public members of derived class.
5. Which of the following statements is correct about the constructors and destructors? (1 分)  
A. Destructors can take arguments but constructors cannot. ✗  
☒ B. Constructors can take arguments but destructors cannot.  
C. Destructors can be overloaded but constructors cannot be overloaded. ✗  
D. Constructors and destructors can both return a value.
6. Which of the following access specifier is used in a class definition by default? (1 分)  
A. protected      B. public      ☒ C. private      D. friend
7. Which of the following concepts means determining at runtime what method to invoke? (1 分)  
A. Data hiding      B. Dynamic Typing  
☒ C. Dynamic binding      D. Dynamic loading
8. How many instances of an abstract class can be created? (1 分)  
A. Infinite      B. 1      C. Variable      ☒ D. 0

9. Which of the following cannot be friend? ( 1 分 )

- A. Function    ☒ B. Class C. Object    D. Operator function

10. `cout` is a/an \_\_\_\_\_. ( 1 分 )

- A. operator    B. function    ☒ C. object    D. macro

11. Which of the following is correct about class and structure? ( 1 分 )

- A. class can have member functions while structure cannot. ☒ X  
B. class data members are public by default while that of structure are private. ☒ X  
C. Pointer to structure or classes cannot be declared.  
☒ D. class data members are private by default while that of structure are public by default.

12. What will be the output of the following program? ( 2 分 )

- A. AABaa-B-A-A    ☒ B. AABba-B-A-A  
C. ABba-B-A    D. ABaa-B-A

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

class FooBase
{
public:
    FooBase()
    {
        cout << "A";
    }
    ~FooBase()
    {
        cout << "~A";
    }
    virtual void SayHello()
    {
        cout << "a";
    }
};

class FooDerived: public FooBase
{
public:
    FooDerived()
    {
        cout << "B";
    }
    ~FooDerived()
    {
        cout << "~B";
    }
    void SayHello()
    {
        cout << "b";
    }
};

int main()
{
    FooBase baseObj;
    FooDerived derivedObj;
    FooBase* fooObjArray[] = { &derivedObj, &baseObj };
    for ( int i = 0; i < sizeof( fooObjArray ) / sizeof( fooObjArray[0] ); i++ )
    {
        fooObjArray[i]->SayHello();
    }
    return 0;
}
```

AABba~B~A~A



13. Which of the following is the correct way of declaring a function as constant? (1分)
- ☒ A. `const int ShowData(void) { /* statements */ }`  
☐ B. `int const ShowData(void) { /* statements */ }`  
☐ C. `int ShowData(void) const { /* statements */ }`  
☐ D. Both B and C
14. Which of the following are available only in the class hierarchy chain? (1分)
- ☐ A. Public data members    ☐ B. Private data members  
☐ C. Protected data members    ☒ D. Member functions
15. Which of the following statements regarding inline functions is correct? (1分)
- ☐ A. It speeds up execution.    ☐ B. It slows down execution.  
☐ C. It increases the code size.    ☒ D. Both A and C.
16. Which of the following concepts is used to implement late binding? (1分)
- ☒ A. Virtual function    ☐ B. Operator function  
☐ C. Const function    ☐ D. Static function
17. What is deep copy? (1分)
- ☒ A. A deep copy creates a copy of the dynamically allocated objects too.  
☐ B. A deep copy just copies the values of the data as they are.  
☐ C. A deep copy creates a copy of the statically allocated objects too  
☐ D. Both B and C above
18. Which one of the following is the correct way to declare a pure virtual function? (1分)
- ☐ A. `virtual void Foo() {0};`    ☒ B. `virtual void Foo() = 0;`  
☐ C. `void Foo() virtual = 0;`    ☐ D. `void Foo() = 0;`
19. Which bitwise operator is suitable for checking whether a particular bit is on or off? (1分)
- ☐ A. `&&` operator    ☒ B. `&` operator  
☐ C. `||` operator    ☐ D. `!` operator

## 二. 综合 (40分)

1. 函数 `ReverseList` 将单链表反转，填空完成如下代码：(4分)

```
struct ListNode
{
    int data;
    struct ListNode* next;
};

struct ListNode* ReverseList( struct ListNode* head )
{
    struct ListNode* newHead = NULL;
    while ( head )
    {
        struct ListNode* next = head -> next;
    }
}
```

```

    head->next
    = newHead;
    newHead head;
    head = next;
}
return newHead;
}

```

2. 如下程序用于判断二叉树是否平衡二叉树，完成 // todo 部分；(9分)

```

struct BinTreeNode
{
    struct BinTreeNode* left;
    struct BinTreeNode* right;
};

int MinDepth( const BinTreeNode* aBinTree )
{
    // todo
}

int MaxDepth( const BinTreeNode* aBinTree )
{
    // todo
}

bool IsBalanced( const BinTreeNode* aBinTree )
{
    // todo
}

```

3. 在 8 位整数里，十进制整数-47 的补码是 11010001，补码 1010 0100 表示十进制整数 -92。(4分)

4. 构造函数中如何处理构造失败？为什么？(5分)

5. 什么情况下会产生“野指针”？举例说明。(6分)

6. 死锁什么情况下发生？(3分)

7. 如下的函数如果被调用，会发生什么？(3分)

```

void Foo( int aNum )
{
    aNum++;
    Foo( aNum );
}

```

8. 下面的程序使用分治的方法计算 32 位整数的二进制码中含有 1 的位数，填空完成：(6分)

```

uint8_t Count1bit( uint32_t num )
{
    num = ( num & 0x55555555 ) + ( ( num >> 1 ) & 0x55555555 );
    num = ( num & 0x00000000 ) + ( ( num >> 16 ) & 0x00000000 );
    num = ( num & 0x0F0F0F0F ) + ( ( num >> 8 ) & 0x0F0F0F0F );
    num = ( num & 0x00000000 ) + ( ( num >> 24 ) & 0x00000000 );
    return ( uint8_t )( num );
}

```

## 四. 程序实践 (40分)

1. 函数 void PrintCharSet( const char\* aString ) 按出现次序无重复打印 aString 中含有的字符，比如输入 aString 为 "ab23bd2" 时，屏幕输出为 ab23d，编程实现之。(12分)

2. 多叉树的遍历方式有先序、后序等方式，编程实现先序遍历时的 `NextTreeNode` 函数，其原型定义为：

```
/**
 * NextTreeNode
 * @param aNode 给定的树节点
 * @return 返回按先序遍历时 aNode 的下一个节点
 */
TreeNode* NextTreeNode( TreeNode* aNode );
```

要求：不能使用递归方式实现，同时需要自行定义 `TreeNode` 类型。（13分）

3. 函数 `void SplitString( const char* aString, char aSeparator )` 以 `aSeparator` 为分割符将 `aString` 拆分打印输出，但注意：当 `aSeparator` 存在于 " 引用的字符串中时、或两个 `aSeparator` 在一起时，将不作为分隔符，比如假定分割符是 + 时，如下为不同情形下的输出示例：

i) `abc+"quoted+quoted"+plus++plus`

输出：

`abc`

`"quoted+quoted"`

`plus+plus`

ii) `abc+"123++123"+def`

输出：

`abc`

`"123++123"`

`def`

iii) `abc+"123+test`

输出：

`abc`

`"123+test`

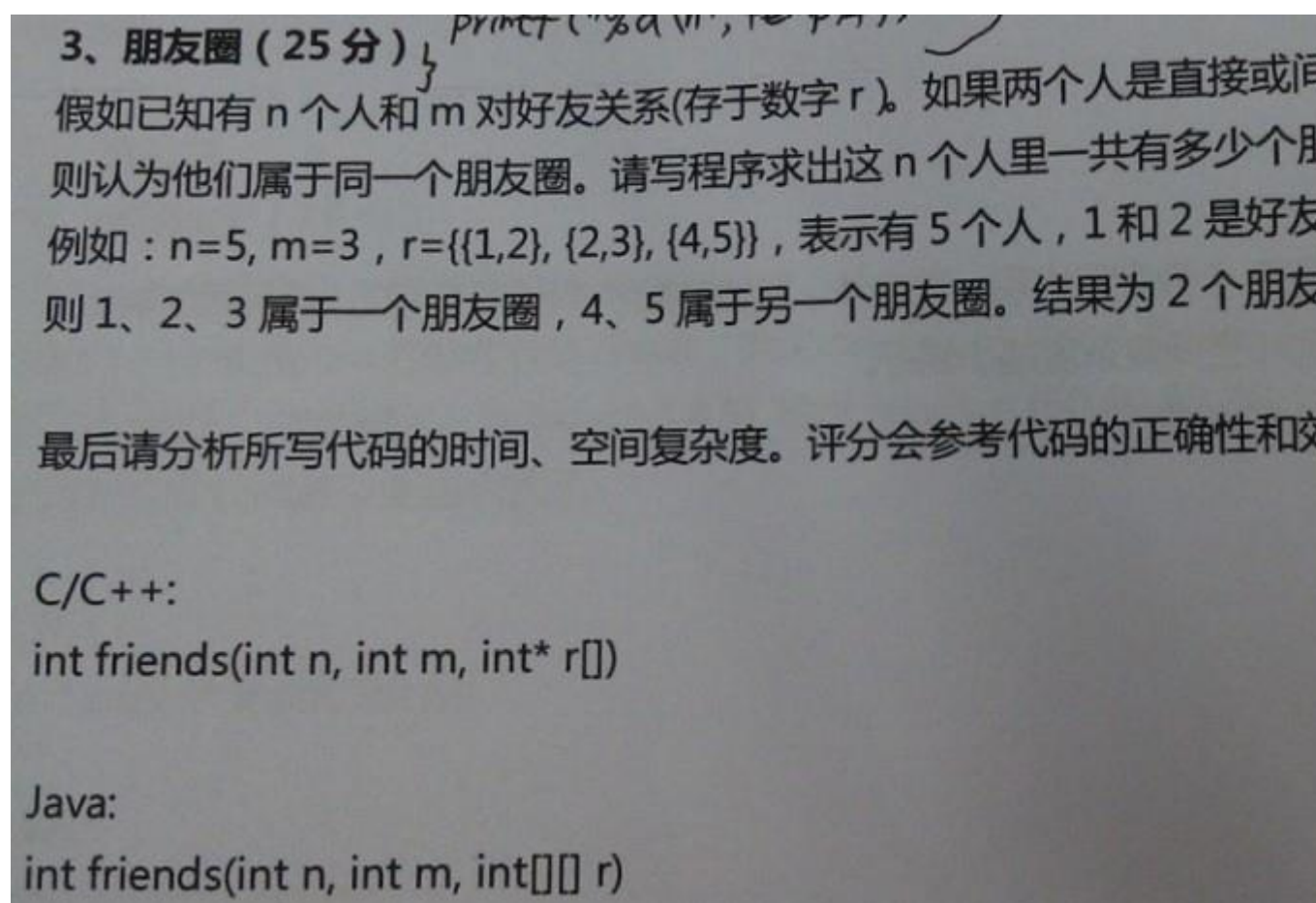
编程实现函数 `SplitString`。（15分）

## 四、小米

**第一大题：**已知数组  $a[n]$ , 求数组  $b[n]$ . 要求:  $b[i] = a[0] * a[1] * \dots * a[n-1] / a[i]$ , 不能用除法。a. 时间复杂度  $O(n)$ , 空间复杂度  $O(1)$ 。 b. 除了迭代器  $i$ , 不允许使用任何其它变量 (包括栈临时变量等) ([本题在我博客: 2012 年腾讯招聘实习生笔试题 有解法](#))

**第二大题：**一个数组里, 除了三个数是唯一出现的, 其余的都出现偶数个, 找出这三个数中的任一个。比如数组元素为【1, 2, 4, 5, 6, 4, 2】, 只有 1, 5, 6 这三个数字是唯一出现的, 我们只需要输出 1, 5, 6 中的一个就行。([本题在我博客: 小米 2013 校园招聘笔试题 有解法](#))

**第三大题：**



3. 朋友圈 (25 分)

假如已知有  $n$  个人和  $m$  对好友关系(存于数字  $r$ )。如果两个人是直接或间接认识(即在好友圈中), 则认为他们属于同一个朋友圈。请写程序求出这  $n$  个人里一共有多少个朋友圈。

例如:  $n=5, m=3, r=\{\{1,2\}, \{2,3\}, \{4,5\}\}$ , 表示有 5 个人, 1 和 2 是好友, 2 和 3 是好友, 4 和 5 是好友。则 1、2、3 属于一个朋友圈, 4、5 属于另一个朋友圈。结果为 2 个朋友圈。

最后请分析所写代码的时间、空间复杂度。评分会参考代码的正确性和复杂度。

C/C++:

```
int friends(int n, int m, int* r[])
```


Java:

```
int friends(int n, int m, int[][] r)
```

## 五、创新工场



# 云帆教育大数据学院

 **创新工场**  
INNOVATION WORKS

2013 创新工场校园招聘笔试题试卷八 A 卷

姓名 \_\_\_\_\_ 学校 \_\_\_\_\_ 手机 \_\_\_\_\_ 毕业时间 2012

\_\_\_\_\_ 学历 \_\_\_\_\_ 邮箱 \_\_\_\_\_

地点 北京 ( ) 上海 ( ☒ ) 成都 ( ) 广州 ( ☒ ) 笔试时间 2012

一部分单选题，把答案写在题目前括号中

3 ) 1. 与 10.110.12.29 mask 255.255.255.224 属于同一网段的主机 IP 地址是 ( ):

☒ A. 10.110.12.0 ☒ B. 10.110.12.30 C. 10.110.12.31 D. 10.110.12.32

) 2. 若系统中有五台绘图仪，有多个进程均需要使用两台，规定每个进程一次仅允许申请一台，则至多允许 ( ) 于竞争，而不会发生死锁。

A. 2 ☒ B. 3 C. 4 D. 5

) 3. 下述代码的输出结果(假定为 32 位机器)为 ( )。

```
struct s
{
    int x: 8;
    int y: 4;
    double a: 8;
    int z: 3;
};
printf("%d\n", sizeof(s));
```

A. 8 B. 16 C. 20 ☒ D. 24

4. 一棵二叉树高度为 h, 所有结点的度或为 0, 或为 2, 则这棵二叉树最少有 ( ) 结点

A. 2h ☒ B. 2h-1 ☒ C. 2h+1 D. h+1

5. 2012! 的末尾有多少个 0?

☒ A. 501 B. 2004 C. 1994 D. 40

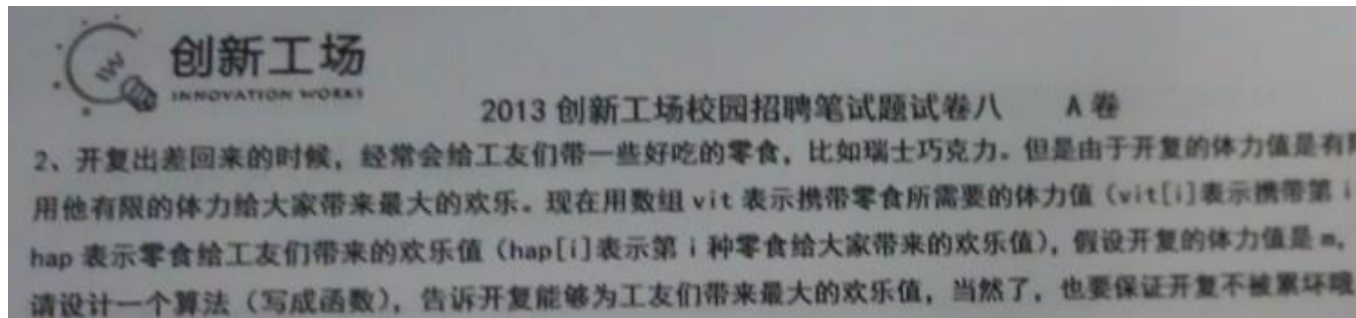
6. #define N 2  
#define Y(n) ((N+1)\*n)  
则执行语句 z=2 \* (N+Y(5+1)); 后, z 的值为 ( )。

A. 32 ☒ B. 36 C. 40 D. 出错

填空题

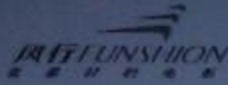
淘宝搜索时，每键入字符都会出现搜索建议，实现这类技术后台采用的数据结构是 字典树+Top K

2. 定义一个宏实现一整数中的偶比特位与奇比特位的交换。



## 六、风行网

# 云帆教育大数据学院



北京风行在线技术有限公司

## 风行网 2013 年校园招聘工程师笔试题

姓名 学校 专业 学历 手机号

申请第一职位 申请第二职位

欢迎参加风行网笔试，注意事项：

一、笔试总时间为 2 小时，请在 2 小时内回答尽量多的题目：

1. 共三部分：第一部分基础题（参考答题时间 1 小时），第二部分附加题（参考答题时间 50 分钟），第三部分综合测评题（参考答题时间 10 分钟）；
2. 附加题分为：软件开发、测试开发、系统运维开发、数据挖掘、前端开发，根据申请职位做相应的附加题，每人只需做 1 份附加题；
3. 申请职位可以填写 2 个，分为：

软件开发：使用 C++ 或 Java、python、object c、php 等语言开发客户端或移动客户端、服务器类软件

测试开发：使用 python 或 java、lua 等语言开发各种自动化测试工具软件

系统运维开发：使用 C 或 python、shell、lua 等语言开发服务器、系统软件

数据挖掘：使用 java 或 python、shell 等语言开发 hadoop 程序或进行各类数据分析

前端开发：使用 java script、CSS 开发 WEB 前端

二、正背面均有题目，请在试卷指定位置填写正确答案；

三、请严格遵守考场纪律，如发现作弊行为将取消面试资格；

四、交卷时，从上至下顺序为：第一部分、第二部分、第三部分、简历和成绩单复印件。

### 第一部分基础题

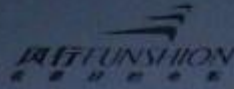
一、编程题(可使用 C/C++ 或者 java、java script 等语言实现)

1、编程实现字符串查找函数。

C/C++: `char * strstr (const char * str1, const char * str2);` // 找到返回子串指针，没找到返回 NULL

Java: `int strstr(String str1, String str2);` // 若找到则返回子串在原串中的位置，没有找到则返回 -1





北京风行在线技术有限公司

2、已知两个链表 head1 和 head2 各自有序，请写一个程序把它们合并成一个链表依然有序。数据结构定义如下。

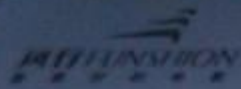
C/C++ 结构：

```
struct node
{
    int _data;
    node* _next;
};
```

JAVA 结构：

```
class Node
{
    int data;
    Node next;
}
```

3、已知两个数组 x[] 与 y[], 各有 n 个元素, 并且也已经以从小到大的顺序排好, 用代码一个程序, 用少于 n 次比较找出 x[] 与 y[] 合并后的中位数



北京风行在线技术有限公司

## 二、简答题

- 1、请简述面向对象软件开发的要素和基本原则
- 2、请简述“死锁”产生原因及如何避免“死锁”
- 3、请简述 TCP 协议栈是如何进行流量控制的？
- 4、请简述操作系统如何解决程序可寻址逻辑空间大而物理内存小的矛盾



北京风行在线技术有限公司

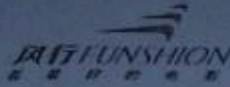
5、目前风行网为用户提供了数十万部可在线点播的影片，每天这些影片的总观看次数在千万以上，请设计一个算法，从原始的观看日志中找出每天观看次数排名前 10 的影片，并分析算法的时间复杂度。（可使用伪代码，或者分步骤进行描述）

6、设计一个棋类游戏的程序（中国象棋、国际象棋、五子棋、跳棋或围棋，五选一），请描述出各个类的设计及之间相互关系（可用 UML 进行描述）





观看次数在千  
影片，并分



北京风行在线技术有限公司

### 三、综合题

1、你觉得成为一个优秀开发工程师或测试工程师应该具有哪些职业素养？你看过哪些软件开发或测试方面的书籍？

2、请罗列 10~20 个你常用的软件(喜欢的电影或常登陆的网站)，并对这些软件使用 2 种以上的方法对其进行分类，每种分类方法均应满足以下条件：

(1)分类可以有多级

(2)除有亲子关系的分类外，不同的分类间不应存在交集

(3)每一级分类均应覆盖所有列出的软件

(4)对每一种软件，应可很容易的判定其属于哪个分类，不存在明显的歧义性或较大的不确定性



北京风行在线技术有限公司

3、今年中日之间就钓鱼岛归属权问题再次爆发争端，由此引发了一轮抵制日货风潮。就此事有以下几种观点：

（1）购买日货就是资助日本政府，资助其军事实力。每买 100 元日货，相当于让日本可以多造 10 颗子弹用来打我们，故应坚决抵制日货。不仅我自己不买，还会用各种方法阻止身边的人购买日货。

（2）抵制日货属于伤敌八百、自损一千的行为。抵制日货后，我国在税收、就业等方面的损失比日本更大，故不应抵制日货。

（3）任何情况下都应尊重他人的自由和财产。抵制日货不应通过阻止他人购买日货实现，更不应打砸他人的财产。我们应把自己的产品做得更好，全面超越日货，从而让人们不再需要购买日货。

（4）自由贸易是财富的源泉，抵制日货会导致对社会分工和自由贸易的破坏，从而减低社会生产率，是一种损人不利己的行为，不应提倡。此外抵制日货及其它外国制品会起到保护落后的本国产品的作用，不利于振兴民族产业。

（5）由于中日韩今年在自由贸易区建设上取得了较大进展，有可能影响到美国在亚太及全球经济体系中的地位，故美国在幕后操纵挑动中日及日韩间的领土争端，以达到破坏中日韩自由贸易区的目的。我们如果抵制日货，正好落入美国圈套，对中国的发展是不利的。

请问：

（1）. 以上观点在哪些方面存在相同点，为什么？

（2）. 以上观点在哪些方面存在不同点，为什么？

以上两题可分别回答也可合并回答。

# 云帆教育大数据学院

云帆教育大数据学院 [www.cloudyhadoop.com](http://www.cloudyhadoop.com)

通过最新实战课程，系统学习 **hadoop2.x** 开发技能，在云帆教育，课程源于企业真实需求，最有实战价值，成为正式会员，可无限限制在线学习全部教程；培训市场这么乱，云帆大数据值得你选择!! 详情请加入 **QQ 群：374152400**，咨询课程顾问！



关注云帆教育微信公众号 **yfteach**，第一时间获取公开课信息。