

1.1 有 A、B、C、D 四个人，要在夜里过一座桥。

他们通过这座桥分别需要耗时 1、2、5、10 分钟，只有一支手电，并且同时最多只能两个

人一起过桥。

请问，如何安排，能够在 17 分钟内这四个人都过桥？

答案：A & B --> 2 mins

1 mins <-- A

C & D --> 10 mins

2 mins <-- B

A & B --> 2 mins

一共 $2 + 1 + 10 + 2 + 2 = 17$ mins

1.2 1-20 的两个数把和告诉 A，积告诉 B，A 说不知道是多少，

B 也说不知道，这时 A 说我知道了，B 接着说我也知道了，问这两个数是多少？

答案：2 和 3

1.3 爸爸，妈妈，妹妹，小强，至少两个人同生肖的概率是多少？

$$1 - \frac{12 \times 11 \times 10 \times 9}{12 \times 12 \times 12 \times 12} = 1 - \frac{55}{96} = \frac{41}{96}$$

1.4 某人去玩具店买小熊，单价 30 元。付给玩具店老板 100 元 玩具店老板没零钱，

去水果店换了 100 元零钱回来找给那人 70 元。那人走后，水果店老板找到玩具店老板说刚

才的 100 元是假币，

玩具店老板赔偿了水果店老板 100 元

问：玩具店老板损失了多少钱？

答案：70+小熊的进价

2 请定义一个宏，比较两个数 a、b 的大小，不能使用大于、小于、if 语句

答案：#define max(a,b) (((long)((a)-(b)))&0x80000000)?b:a)

若 $a > b$ ，则 $a-b$ 的二进制最高位为 0，与上任何数还是 0，所以大数为 a；

否则， $a-b$ 为负数，二进制最高位为 1，与上 0x80000000 (最高位为 1 其他为 0) 之后为 1，

所以此时的大数为 b。

3 计算 $a \wedge b \ll 2$

答案：爱思益求职

运算符优先级：括号，下标， \rightarrow 和 \cdot (成员) 最高；

单目的比双目的高；

算术双目的比其他双目的高；

位运算 高于 关系运算；

关系运算 高于 按位运算(与，或，异或)；

按位运算 高于 逻辑运算；

三目的只有一个 条件运算，低于逻辑运算；

赋值运算仅比 $,$ (顺序运算) 高。

在此题中，位左移 " \ll " 优先级高于 按位异或 " \wedge "，所以 b 先左移两位 (相当于乘以 4)，

再与 a 异或。

例如：当 $a = 6$ ； $b = 4$ 时；则 $a \wedge b \ll 2 = 22$

4 如何输出源文件的标题和目前执行行的行数？

答案：printf("The file name: %d\n", __FILE__);

printf("The current line No:%d\n", __LINE__);

ANSI C 标准预定义宏：

__LINE__

__FILE__

__DATE__

__TIME__

__STDC__ 当要求程序严格遵循 ANSI C 标准时该标识符被赋值为 1

__cplusplus__ 当编写 C++ 程序时该标识符被定义

5 a[3][4] 哪个不能表示 a[1][1]: *(&a[0][0]+5) *(*(a+1)+1) *(&a[1]+1) *(&a[0][0]+4)

6 fun((exp1, exp2), (exp3, exp4, exp5)) 有几个实参?

答案：两个。

形式参数：在声明和定义函数时，写在函数名后的括号中的参数。

实参是调用参数中的变量，行参是被调用函数中的变量。

7. 希尔 冒泡 快速 插入 哪个平均速度最快?

答案：快速排序

快速排序、归并排序和基数排序在不同情况下都是最快最有用的。

8. enum 的声明方式

答案：enum 枚举类型名 {

枚举常量 1,

枚举常量 2,

...

枚举常量 n

};

For example:

```
enum weekday { sunday, monday, tuesday, wednesday, thursday, friday,
saturda
```

```
y};
```

enum weekday week_day;//week_day 就是一个枚举类型变量

9. 频繁的插入删除操作使用什么结构比较合适，链表还是数组？

答案：链表

10. *p=NULL *p=new char[100] sizeof(p)各为多少？

答案：都为 4。因为都是指针类型，所占存储空间必然为 4。

11. 顺序查找的平均时间

答案： $(1+2+3+\dots+n)/n = (n+1)/2$

12. for(i=0,sum=0; i<10; ++i,sum+=i);的运行结果

答案：sum = 55

13. 不能做 switch()的参数类型是：

答案：switch 的参数不能为浮点型。

14. 不使用其他变量，交换两个整型 a，b 的值

答案： $x = x+y; y = x-y; x = x-y$

15. 写出 float x 与“零值”比较的 if 语句。

if(x>=0.000001 && x<=-0.000001) (x 不为 0 的比较)

float: 6 位精度

double: 16 位精度

16. 两个数相乘，小数点后位数没有限制，请写一个高精度算法

数据库

1. 有个表 tableQQ，有整型的 ID 项和字符类型的 Nickname 项，这两个项都不允许为空

(1) 写出建立该表的 SQL 语句

(2) 找出 Nickname 为 QQ 的用户，按 ID 降序排列的 SQL 语句

(3) 写出删除 ID 为 1234 用户记录的 SQL 语句

(4) 写出添加 ID 为 5555，Nickname 为 '1234' 的 SQL 语句

答案：

(1) CREATE TABLE tableQQ

(

ID NUMBER(12) NOT NULL,

Nickname Varchar2(30) NOT NULL

);

(2) select * from tableQQ where Nickname = 'QQ' order by ID desc;

(3) from tableQQ where >

(4) into tableQQ values(5555,'1234');

//删除表

(5) tableQQ;

2. 有关系 s(sno, sname) c(cno, cname) sc(sno, cno, grade)

1 问上课程 "db" 的学生

2 成绩最高的学生号

3 每科大于 90 分的人数

答案：

(1) select a.sno, a.cno, b.cno, b.cname from sc a, c b where a.cno = b.cn

o and b.cname = 'db';

(2) select sno, max(grade) from sc group by sno;

(3) select cno, count(sno) from sc where grade > 90 group by cno;

操作系统 网络

1. 描述实时系统的基本特性

答案：在特定时间内完成特定的任务，实时性与可靠性。

2. Internet 采用哪种网络协议？该协议的主要层次结构？

答案：TCP/IP 协议。应用层、传输层、网络层、数据链路层和物理层。

3. Internet 物理地址和 IP 地址转换采用什么协议？

答案：地址解析协议 ARP address resolution protocol

4. IP 地址的编码分为哪俩部分？

答案：网络号和主机号。不过是要和“子网掩码”按位与上之后才能区分哪些是网络位

哪些是主机位。

搜集的一些腾讯笔试题，大家一起努力~

2005 年腾讯招聘

选择题 (60)

程序填空 (40)

1. (20) 4 空 x5

不使用额外空间，将 A, B 两链表的元素交叉归并

2. (20) 4 空 x5

MFC 将树序列化 转存在数组或 链表中！

1, 计算 $a^b \ll 2$ (运算符优先级问题)

2 根据先序中序求后序

3 a[3][4] 哪个不能表示 a[1][1]: $*(\&a[0][0]) * (*(a+1)+1) * (\&a[1]+1) * (\&a[0][0]+4)$

4 for(int i...)

for(int j...)


```
printf(i, j);
```

```
printf(j)
```

会出现什么问题

5 for(i=0;i<10;++i, sum+=i); 的运行结果

6 10 个数顺序插入查找二叉树，元素 62 的比较次数

7 10 个数放入模 10hash 链表，最大长度是多少

8 fun((exp1, exp2), (exp3, exp4, exp5)) 有几个实参

10 二分查找是 顺序存储 链存储 按 value 有序中的哪些

11 顺序查找的平均时间

12 *p=NULL *p=new char[100] sizeof(p) 各为多少

15 1-20 的两个数把和告诉 A, 积告诉 B, A 说不知道是多少, B 也说不知道, 这时 A 说我知道了, B 接着说我也知道了, 问这两个数是多少

大题:

1 把字符串转换为小写, 不成功返回 NULL, 成功返回新串

```
char* toLower(char* sSrcStr)
```

```
{
```

```
char* sDest= NULL;
```

```
if(____1____)
```

```
{
```

```
int j;
```

```
sLen = strlen(sSrcStr);
```

```
sDest = new [____2____];
```

```
if(*sDest == NULL)
```

```
return NULL;
```

```
sDest[sLen] = '\0';
```

```
while(____3____)
```

```
sDest[sLen] = toLowerChar(sSrcStr[sLen]);
```

```
}
```

```
return sDest;
```

```
}
```

2 把字符串转换为整数 例如：“-123” -> -123

```
main()
```

```
{
```

```
.....
```

```
if( *string == '-' )
```

```
n = ____1____;
```

```
else
```

```
n = num(string);
```

```
.....
```

```
}
```

```
int num(char* string)
```

```
{
```

```
for(; !(*string==0); string++)
```

```
{
```

```
int k;
```

```
k = __2__;
```

```
j = --sLen;
```

```
while( __3__ )
```



```
k = k * 10;
```

```
num = num + k;
```

```
}
```

```
return num;
```

```
}
```

```
struct st{
```

```
int i;
```

```
short s;
```

```
char c;
```

```
};
```

```
sizeof(struct st);
```

两个数相乘，小数点后位数没有限制，请写一个高精度算法

与大数相乘原理一样，只是调整小数点位置，即有效小数个数

算法提示：

输入 string a, string b; 计算 string c=a*b; 返回 c;

1, 纪录小数点在 a, b 中的位置 l1, l2, 则需要小数点后移动位置数为
 $l = \text{length}(a) + \text{length}(b) - l1 - l2 - 2$;

2, 去掉 a, b 中的小数点, (a, b 小数点后移, 使 a, b 变为整数)

3, 计算 $c = a * b$; (要么用 java 的 BigInteger 搞, 要么自己用 C++ 写高精度乘法, 超过百万位, 用 FFT, 我就不细说, 这都预先写过就别做了)

4, 输出 c, (注意在输出倒数第 1 个数时, 输出一个小数点。若是输出的数少于 1 个, 就补 0)

1. 哪一个声明是错误的 ()

```
char * const p = 0;
```

```
char const * p = 0;
```

```
const char * p = 0;
```

```
char * p const = 0;
```

2. 下面的结构体所占内存是(12)

```
#pragma pack(4)
```

```
struct {
```

```
    unino {
```

```
        char ch1;
```

```
        double num;
```

```
    } bbb;
```

```
    char ch2;
```

```
    }aaa;
```

```
#pragma pack()
```

3. $x = 0xffffffff$ ，则 $x = (x \& 0xffff0000) \gg 16$ 的结果是()

4. `#define TEST(x) (x*2+1)`，则 `TEST(1+1)` 的结果是()

5. 有 15 个结点的满二叉树，叶子数是()

6. 程序片段的输出结果()

```
class A {
```

```
public:
```

```
    A() { printf("A1 "); }
```

```
    ~A() { printf("A2 "); }
```

```
};
```

```
class B : public A {
```

```
public:
```

```
    B() { printf("B1 "); }
```

```
~B() { printf("B2 "); }
```

```
};
```

```
B b = new B;
```

```
b;
```

7. int Func(const char *)的重载函数是() (选项略)

8. 按顺序 A, B, C 进栈，出栈的结果是()

9. 程序片段的输出结果()

```
class A {
```

```
public:
```

```
virtual int func() { return 1; }
```

```
};
```

```
class B : public A {
```

```
public:
```

```
virtual int func() { return 2; }
```

```
};
```

```
A *a = new B;
```

```
printf("%d %d\n", a->func(), a->A::func());
```

```
a;
```

10. 程序片段的执行描述中正确的是()

```
class A {
```

```
public:
```

```
virtual void func(A& ra);
```

```
};
```

```
class B : public A {
```

```
public:
```

```
virtual void func(B& rb);
```

```
};
```

```
B b;
```

```
A &a = b;
```

```
a.func(b);
```

(有点记不清了，反正俺错了这题，以为引用和指针是不同的...>_<)

11. 关于哈希表特点错误的是() (选项略)

12. 函数功能描述正确的是()

```
int whatido(char * str)
```

```
{
```

```
return *str ? whatido(++str)+1 : 0;
```

```
}
```

笔试题-腾讯数据库笔试题:

在一个文件中有 10G 个整数，乱序排列，要求找出中位数。内存限制为 2G。只写出思路即可。

腾讯笔试题解答 (Peak Wong):

1, 把整数分成 256M 段，每段可以用 64 位整数保存该段数据个数， $256M \times 8 = 2G$ 内存，先清 0

2, 读 10G 整数，把整数映射到 256M 段中，增加相应段的记数

3, 扫描 256M 段的记数，找到中位数的段和中位数的段前面所有段的记数，可以把其他段的内存释放

4, 因中位数段的可能整数取值已经比较小 (如果是 32bit 整数，当然如果是 64bit 整数的话，可以再次分段)，对每个整数做一个记数，再读一次 10G 整数，只读取中位数段对应的整数，并设置记数。

5, 对新的记数扫描一次，即可找到中位数。

如果是 32bit 整数，读 10G 整数 2 次，扫描 256M 记数一次，后一次记数因数量很小，可以忽略不记。

解释一下：假设是 32bit 整数，按无符号整数处理

整数分成 256M 段？整数范围是 $0 - 2^{32} - 1$ 一共有 4G 种取值， $4G/256M = 16$ ，每 16 个数算一段 0-15 是 1 段，16-31 是一段，...

整数映射到 256M 段中？如果整数是 0-15，则增加第一段记数，如果整数是 16-31，则增加第二段记数，...

其实可以不用分 256M 段，可以分的段数少一些，这样在扫描记数段时会快一些，还能节省一些内存。