# 完美世界面试题答案

1. 计算题

1.

sizeof(C\_1)＝1 sizeof(C\_2)＝4 sizeof(C\_3)＝16

2.

4 占一个指针的大小

3.

0x2F = 47;0xBC = 188;0xF8 = 248; 0xFF = 255;

0x02 ^ 0x03 = 1;(按位异或) 0xF2 & 0x0F = 2;

0x07 + 0x01 = 8;

1. 问答题

1.

a.

base b;声明一个base的结构体变量b；

b.init();调用b的方法对其进行初始化赋值；

b.show();调用b的打印函数。

b.

const void show();

2.

禁止在堆上开辟空间：将new operator和delete operator重载设置成private；

禁止在栈上开辟空间：将构造和析构函数私有化

3.

reinterpret\_cast<type-id> (expression)

type-id 必须是一个指针、引用、算术类型、函数指针或者成员指针。它可以把一个指针转换成一个整数，也可以把一个整数转换成一个指针（先把一个指针转换成一个整数，再把该整数转换成原类型的指针，还可以得到原先的指针值）。

static\_cast

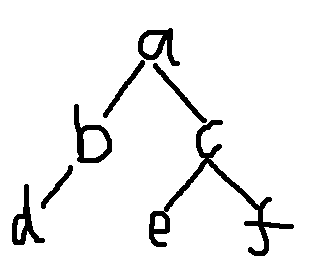
该运算符把expression转换为type-id类型，但没有运行时类型检查来保证转换的安全性。

dynamic\_cast

dynamic\_cast运算符可以在执行期决定真正的类型。如果 downcast 是安全的（也就说，如果基类指针或者引用确实指向一个派生类对象）这个运算符会传回适当转型过的指针。如果 downcast 不安全，这个运算符会传回空指针（也就是说，基类指针或者引用没有指向一个派生类对象）。

三、

1.

a.后序遍历是：dbefca

b.

可以根据任意两个写出另一个，因为两个可确定一棵树

c.

typedef struct node

{

char ch;

struct node \*left,\*right;

}node; // 定义节点的结构

node \* creat(char \*pre,char \*in,int len);

void print(node \*head);

int main()

{

int i,j,k,m,n,len;

char pre[30],in[30]; // 存储先序和中序遍历的序列

node \*head;

head=(node\*)malloc(sizeof(node));

while(scanf("%s%s",pre,in)!=EOF)

{ len=strlen(pre);

head=creat(pre,in,len);

print(head);

printf("\n");

}

return 0;

}

node \* creat(char \*pre,char \*in,int len) // 创建后序遍历的函数

{ int k;

if(len<=0) return NULL;

node \*head=(node\*)malloc(sizeof(node));

head->ch=\*pre;

char \*p;

for(p=in;p!=NULL;p++)

if(\*p==\*pre) break; // 在中序遍历的序列中得到与先序相同的节点

k=p-in;

head->left=creat(pre+1,in,k); //递归调用得到左子树

head->right=creat(pre+k+1,p+1,len-k-1);//得到右子树

return head;

}

void print(node \*head) // 打印后序遍历序列

{

if(head==NULL) return ;

print(head->left);

print(head->right);

printf("%c",head->ch);

}

2.

void func(map<int, string>\* \_map,int n){

for (map<int, string>::iterator i=\_map->begin(); i!=\_map->end(); i++)

{

if (i->first < n)

{

\_map->erase(i);

}

}

}

3.

int func(int n){

if (n>1) {

return n\*func(n-1);

}

else

return 1;

}

4.

#define max(a,b) (((a)>(b))?(a):(b))