***注：请不要在本试题上作答，答案填写在另外的答题纸上。***

**分布式数据库研发中心 C++ 试题**

***注意事项：***

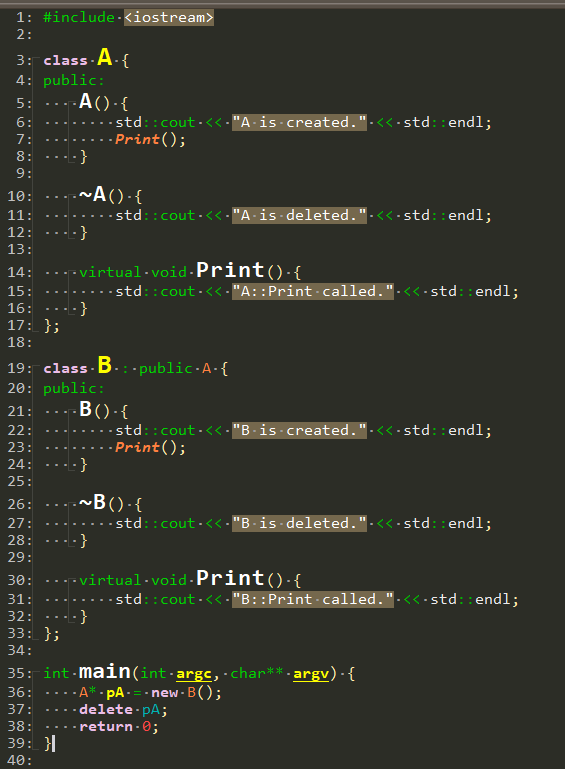
1. ***本试题为闭卷测试，请勿使用互联网或书籍。***
2. ***答题者请基于32位元操作系统的特性回答所有题目***
3. ***答题者碰到题目的答案是输出部分结果后崩溃,必须回答该部分结果的内容***
4. ***答题者若认为答案跟编译器设计有相依性,请注明是基于哪个编译器作答***
5. **在 C++中，struct 和 class 在定义或操作特性上有什么不同？**

答：struct定义成员默认是公有属性，class是私有的。

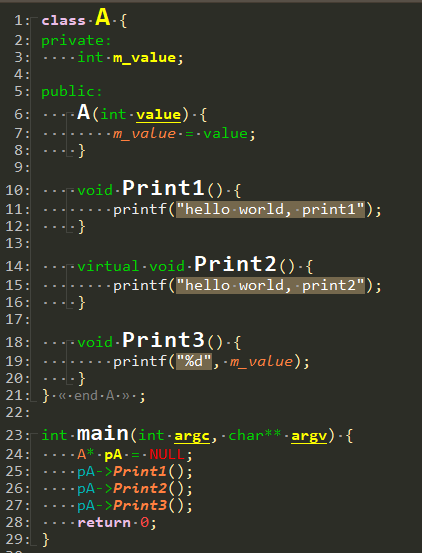
1. **当一个类A 中没有声命任何成员变量与成员函数,这时sizeof(A)的值是多少？如果不是零，请试著解释为什么编译器没有让它为零？**

**答：sizeof(A)的值是1。**

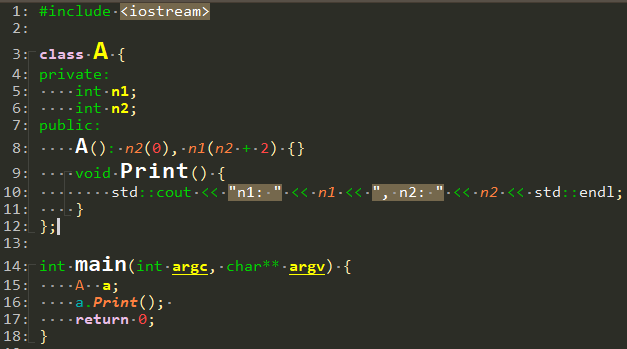
1. **运行下图中代码，输出的结果是什么？这段代码有什么问题？**



1. **下面的C++代码，能否正常编译执行？如果能，运行结果是什么？如果不能，请说明原因。**

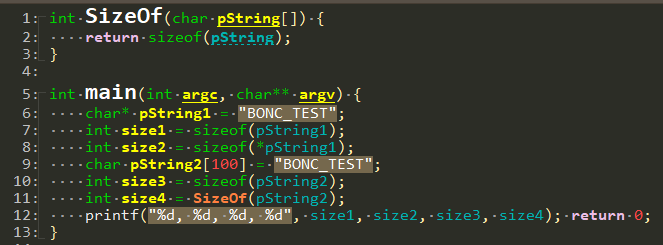


1. **如下的C 代码，能否正常运行？如果能，输出是什么？如果不能，请说明原因。**

****

**答，可以输出。结果是n1: 2 n2: 0**

1. **如下的C 代码，能否正常运行？如果能，输出是什么？如果不能，请说明原因。**

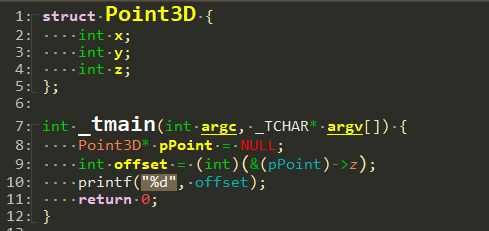
****

答：可以正常运行。输出是 4, 9 4 4。

1. **说明一下线程的基本概念，多线程同步和互斥有何异同以及在什么情况下分别使用他们？举例说明。**

答：线程是调度的基本单位。线程同步用于多个线程并发修改某个变量时，结果总是保持正确。线程互斥是对于多个线程共享的变量，一次只能有一个线程可以修改该变量。

1. **下列C++代码，能否正常编译运行？如果能，结果是什么？如果不能，为什么？**

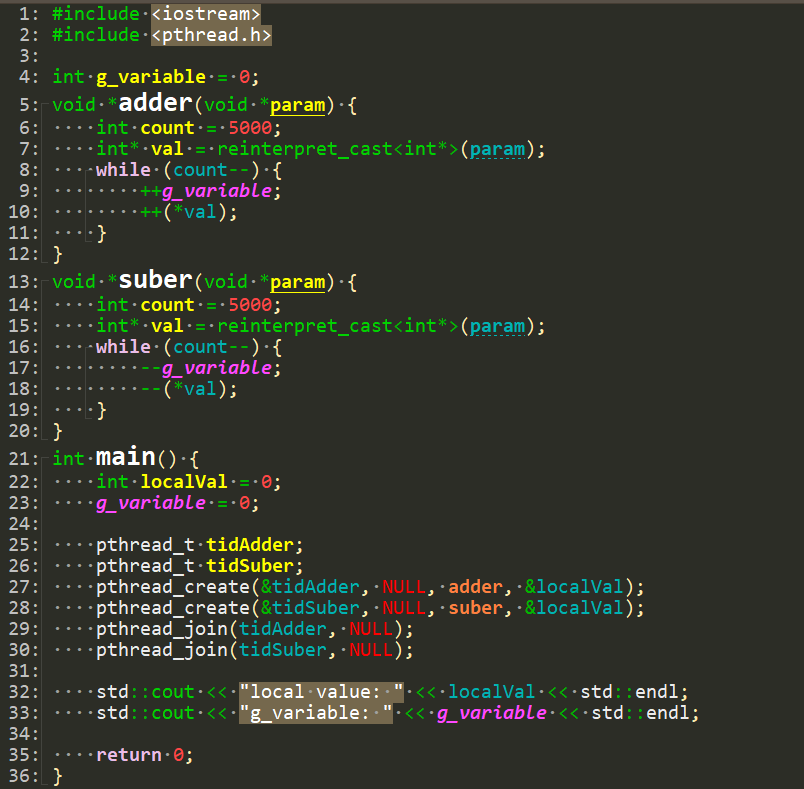
****

1. **以下代码能否正常编译执行？如果能，请描述其功能，并给出示例；如果不能，请说明原因。**

****

**答：**

1. 以下代码能否正常编译运行？如果能，请给出结果；如果不能，请说明原因。

****

1. **实现一个二分查找函数**

**(1)函数原型定义**

**int bin\_search(int \*array, int array\_size, int key)**

**(2)函数入参说明**

**参数 array 是按从小到大排序的数组**

**参数 array\_size 是array的有效长度**

**参数 key 是要查找的数值**

**(3)函数返回值说明：**

**在array中找出与key相等的值，并返回它的下标**

**int bin\_search(int \*array, int array\_size, int key) {**

**if(array\_size<1) {**

**return -1;**

**}**

**int l=0,r=array\_size-1;**

**while(l<r) {**

**int mid=(l+r)>>1;**

**if(array[mid]>key) {**

**r=mid-1;**

**}else if(array[mid]<key) {**

**l=mid+1;**

**}else{**

**return mid;**

**}**

**}**

**return -2;**

**}**

1. **集合操作算法**(给出主要算法和数据结构的思路）

**数据源定义**

**S1 : { a, b, c, d, d, a, b }**

**S2 : { b, d, e, f, b }**

**…**

**…**

**Sn : { … … }**

**EnterS 函数**

**Talking a set as input and storing the information in the data structure that you are going to define.**

**PrintS 函数**

**Taking one of { UA, U, I } as input, visiting the data structure that you defined and …**

**-UA ( Union All 全并 ) : printing out all elements in all sets that have been entered.**

**-U ( Union 并集 ) : printing out all unique elements in all sets that have been entered.**

**-I ( Intersect 交集 ) : printing out the intersection of the sets.**

**For example**

**EnterS( S1 ) -> enter the info into the data structure that you defined.**

**EnterS( S2 ) -> enter the info into the data structure that you defined.**

**PrintS( UA ) : { a, b, c, d, d, a, b, b, d, e, f, b }**

**PrintS( U ) : { a, b, c, d, e, f }**

**PrintS( I ) : { b, d }**

**Note :**

**a)a, b, c, d, e, f 等是集合的元素，不代表英文字母 ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’，它们可能代表任意的数据结构。**

**b)集合的操作无须考虑顺序，打印不需依赖顺序。**

**试题：**

**试定义出 EnterS 及 PrintS 所需要的核心数据结构与操作算法，需考虑大数据量时的完整性、合理性与可执行性。**

答：可采用设计一个HashSet存放元素。HashSet采用数组存储，数组下标存储元素值，添加一个之后数组对应下标元素置为1。

取交集，定义2个hashset A、B，遍历其中一个A，去判断元素是否在B中，在的话，结果保存到C里。遍历完成后，返回C。

取并集，定义2个hashset A、B，遍历其中一个A，去判断元素是否在B中，不在的话，结果保存到B里。遍历完成后，返回B。

public class HashSet {

int [] hash;

public HashSet() {

hash = new int[1000001];

}

public add(int key) {

hash[key]=1;

}

public void remove(int key) {

hash[key]=1;

}

public boolean contains(int key) {

if(hash[key]==1) {

return true;

}else {

return false;

}

}