华东师范大学 数据科学与工程学院 2017-2018学年 第一学期 期末考试试卷

科目： 数据结构与程序设计 主讲教师： 胡卉芪 出卷人： 胡卉芪

A 卷

考试说明：考卷共 15 页, 12题，答题时间120分钟. 最后页为补充的空白页，填不下或需改正的地方可在原题中说明，填在补充页中。

考题：

## 题目1-C++基本概念 (8分)

请写出至少4个在C++出现，但C语言中不存在的关键字，并用1-2句话简单描述该关键字的功能或用法。填入下述表格2-5行 (第1行为示例，每行一分，多写不加分)。

|  |  |
| --- | --- |
| 关键字 | 功能描述 |
| class | 用class来定义类，比如class A{…}；则A为类的名称。 |
|  |  |
|  |  |
|  |  |
|  |  |

## 题目2-C++代码阅读(8分)

请阅读下述程序，按照print或cout函数，依次写出程序所有的输出。

举例：对下述a+b的程序

#include<stdio.h>

int main(){

int a, b;

a=b=5;

print(“a=%d, b=%d;\n”);

print(“a+b=%d;\n”);

return 0;

}

上例输出程序将包含两行:

a=5, b=5;

a+b=10;

阅读程序如下：

#include <iostream>

using namespace std;

void ff (int &m, int \*n)

{

m=16;

\*n=8;

}

int rec(int a,int b){

if(a&&b){

cout<<a<<" "<<b<<endl;

return rec(a/4,b/2)+rec(a/8,b/8)+1;

}

else return 0;

}

int main()

{

int m =8;

int n = 16;

ff(n, &m);

cout << n <<" "<< m << endl;

cout<<rec(n,m)<<endl;

return 0;

}

依次写出程序所有的输出：

## 题目3-C++代码修正(9分)

请指出下述C++代码中**三处**语法或实现的错误，其中语法错误是导致程序无法编译通过的错误，实现错误是指能通过编译但程序执行后会崩溃的错误。然后为每个指出的错误简述可能的解决方案, 并填入下述表格的错误1-3中(表格中错误0为举例，每个错误2分，多写不加分)。

1. #include<iostream>
2. class Test
3. {
4. public:
5. Test(int t) {
6. p=new int;
7. \*p=t;
8. }
9. Test(){}
10. ~Test(){ delete p; }
11. void A() const{
12. return;
13. }
14. void B (int \*t) const {
15. p=t;
16. }
17. private:
18. int \*p;
19. };
20. int main()
21. {
22. cout<<“Test Class”<<endl;
23. Test a();
24. a.A();
26. Test \*b=new Test(ten);
27. int ten=10;
28. b->B(&ten);
29. Test \*c=new Test(\*b);
30. delete c;
31. delete b;
32. return 0;
33. }

|  |  |  |  |
| --- | --- | --- | --- |
| 错误 | 类型 | 错误描述 | 改正方法 |
| 错误0 | 编译错误 | 未使用using namespace std; 导致24行cout与endl符号无法识别 | 在行1-2间添加代码  using namespace std; |
| 错误1 |  |  |  |
| 错误2 |  |  |  |
| 错误3 |  |  |  |

## 题目4 (8分)-栈和队列

1. 设有编号为1，2，3，4的四辆列车，依次进入一个队列式结构的车站，具体写出这四辆列车开出车站的顺序 (3分)。
2. 设有编号为1，2，3，4的四辆列车，依次进入一个栈式结构的车站。现允许先进站的列车在后续列车进站前离开车站。比如当1号车在入站后，且先于2、3、4号车进站前出站，那么列车最终出站的顺序为1, 4, 3, 2。

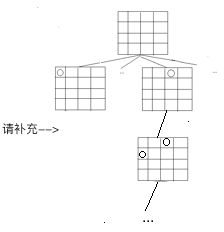
请写出这四辆列车开出车站的所有可能的顺序 (5分)。

## 题目5 (7分) -回溯法

四皇后问题。求解如何在4×4的棋盘上无冲突的摆放4个皇后棋子。在国际象棋中，皇后的移动方式为横竖交叉的，因此在任意一个皇后所在位置的水平、竖直、以及45度斜线上都不能出现皇后的棋子。

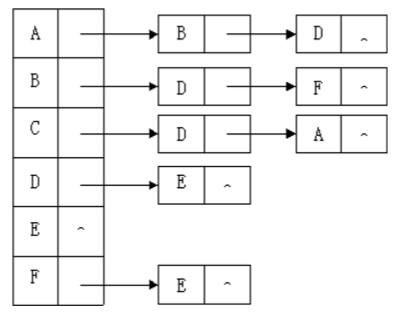
采用回溯法求解四皇后问题。算法首先尝试在第1行第1列放置一颗棋子，然后尝试在第2行无冲突的列(第3列)放置第二枚棋子，然后依次在第三行、第四行无冲突的列放置棋子。若在第四行能放置无冲突的棋子，那么算法寻找到了一个解。对于其中的任意一行，若无法进一步放置无冲突的棋子，则算法会回退一行，尝试在其他无冲突列放置棋子。算法直到尝试过所有可能性后结束。

1. 回溯法本质上是产生一颗解空间树，通过探索这棵解空间树，可以得到问题的一种或几种解。下图是该问题的解空间树，根节点包含四颗子树，分别表示第1行棋子放在1，2，3，4列的子空间，请参照第1棵子树的部分作图示例（无需画有冲突的尝试）, 补充完全第4棵子树 (即当棋盘的第1行第3列放置了一颗棋子，下一步开始要在第2行尝试放置棋子时算法尝试，直到尝试完放置第四行棋子) 的搜索过程（5分）。



1. 画出四皇后问题的一个解 (2分)

**题目6 (10分)-DFS**



上图是用邻接表存储的一个有向图：

1. 画出此图（3分）
2. 写出从C点开始按深度优先遍历该图的结果 (3分)
3. 用栈实现第（2）小题深度优先遍历时，画出栈中元素的变化过程 （4分)

## 题目7 (9分)-AVL树

给出下述的一颗AVL树，分别画出下述插入/删除完成之后的AVL树。 *注：每个插入/删除操作均为分开的对下面给出的AVL树直接进行操作，并不是5个连续操作。*

给出的AVL树及操作如下：



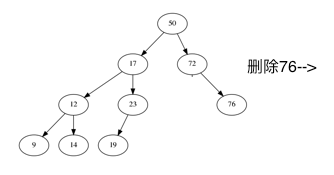
(2)



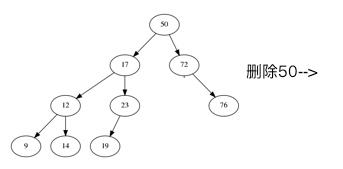
(3)



(4)



(5)



## 题目8 (9分)-B树

给出下述的5颗B树，分别画出下述插入/删除完成之后的B树。 其中删除操作中，B树将借调左右兄弟节点，优先借调右兄弟的节点。*注：每个插入/删除操作均为分开的对下面给出的B树直接进行操作，并不是5个连续操作。*

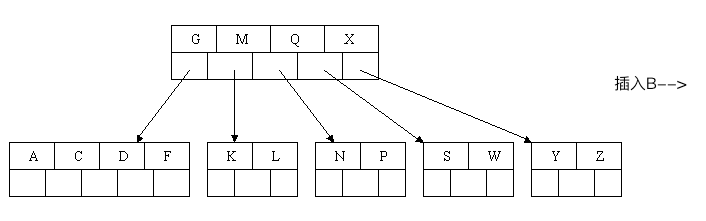
给出的B树及操作如下，每颗树中每个节点的键值个数不超过4。



(2)



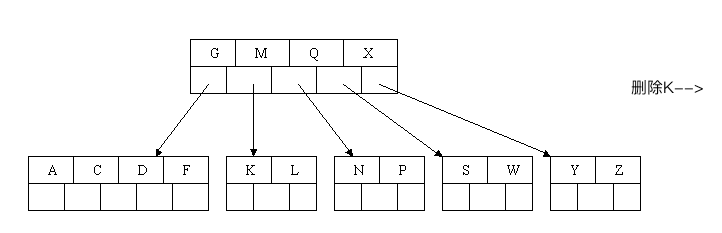
(3)



(4)



(5)



## 题目9 (8分)-拓扑排序

请对下面的有向无环图进行拓扑排序。



1. 请写出一种拓扑排序的结果(5分)
2. 现在利用BFS实现上图的拓扑排序, 其步骤如下： 1) 算法首先统计每个节点的入度，如0号节点的入度为1, 6号节点的入度为2。 2) 算法将所有入度为0的节点，按照节点序号从小到大依次加入一个队列Q中 (节点2, 8, 这时候队列中有2个节点)。3) 算法一次从Q中弹出一个节点，然后将该点所有出边指向的节点的入度分别减1， 比如队列中的第一个节点为2,弹出后节点0与3的入度将分别减一。4）算法将所有减一后节点入度为0的点按照节点序号从小到大加入到Q。5) 重复以上3）、4）两个步骤直到所有节点均已在Q中得到过处理。

根据上图及BFS算法，求在整个算法执行过程中，队列Q中最多的时候有多少个节点，分别是哪几个节点 (3分)？

## 题目10 (6分)-索引设计

有N (N=3)个文档，每个文档包括1个id和若干个 (1-3个)单词。另有一个查询单词q (q=”cat”)。现允许对文档进行预处理及提前构建索引，请设计相应数据结构及查询算法，使得在*不遍历所有文档*的情况下尽快地返回包含该查询单词 (即文档具备完全匹配的子字符串q)的所有文档id。

对下面给出文档及q=”cat”, 相应的结果为：

* id= 2，3

|  |  |
| --- | --- |
| Id | 文档 |
| 1 | rat |
| 2 | tom cat |
| 3 | tomcat in java |

请为上述文档画出相应的索引结构，并以q=”cat”简述查询过程。

## 题目11 (9分)-应用题1

Michael喜欢登山，爬的区域必须向上倾斜，而且当爬到坡顶，不得不再次从另一个坡底出发。Michael想知道在一个区域中最长的爬坡。区域由一个二维数组给出。数组的每个数字代表点的高度。下面是一个例子:

1 2 3 4 5

16 17 18 19 6

15 23 23 20 7

14 22 22 21 8

13 12 11 10 9

1. 一个人可以从某个点爬向上下左右相邻四个点之一，当且仅当高度增加。在上面的例子中，一条可行的坡道为1-2-17-23。当然1-2-3-4-5-6…-20-21-22-23这条更长。事实上，这是最长的一条。请写出相应的伪代码，求出这条最长坡道的长度，即坡道包含的点的数量 (6分)。
2. 现在Michael对爬坡的要求做了一点改变，允许爬高度增加或者高度不变的坡。因此，上面的例子中，1-2-17-23-23是一条可行的坡道，另外1-2-3-4-5-6…-20-21-22-22-

23-23这条是最长的可行坡道。另外假定区域中至多只有两个点拥有相同的高度，如图中至多只有两个22，两个23。 请写出相应的伪代码，求出最长坡道的长度，即坡道包含的点的数量 (3分)。

## 题目12 (10分)-应用题2

1. 遗产继承:当前婚育制度下一个家庭最多可以有两个子女，整个家庭的族谱构成了一颗二叉树，树的节点为1到n (下面伪代码中用number表示)中的值。 给定族谱的前序遍历，请写出建立族谱的**伪代码**（3分）；

说明：给定的前序遍历：1 2 4 # # 5 7 # # 8 9 # # # 3 6 # # #，则构建的树为



伪代码:

#include <iostream>

using namespace std;

struct BinaryTreeNode

{

int m\_nValue;

BinaryTreeNode\* m\_pLeft;

BinaryTreeNode\* m\_pRight;

BinaryTreeNode(){

m\_pLeft = NULL;

m\_pRight = NULL;

}

};

int number;

//前序构建二叉树的函数

int main() {

cin>>number;

BinaryTreeNode\* pRoot = NULL;

//构建二叉树的函数

return 0;

}

1. 已知任何一个孩子节点的遗产继承自其父亲节点，如果父亲节点只有一个孩子那么继承100%，如果父亲节点有两个孩子那么继承50%（两个孩子平分），整个家族的遗产决定于家族创始人（root节点）的总资产。现在给定了家族创始人的资产值S（double）和要计算的孩子节点标记C（m\_nValue=C）,请写出遗产计算的**伪代码**（2分）



例如:资产值**S=32**,孩子节点**2和3**平分遗产而获得**16**，同理节点4和5都获得平分节点2所得的8，而节点7和8则获得平分节点5所得的4；由于节点3仅有一个孩子节点，故节点6获得节点3的所有遗产16，节点8同样只有一个孩子节点，节点9获得节点8的所有遗产4。

在第一问的基础上：

//计算遗产

Calc\_Heritage (BinaryTreeNode\* pRoot, int S ,int C )

{

}

int main() {

cin>>number;

BinaryTreeNode\* pRoot = NULL;

//构建二叉树

int S,C;

cin>>S>>C;

Calc\_Heritage(pRoot,S,C );

return 0;

}