# 单项选择题

1 选C 前三步操作相同，主要看第四步，先分析前三步做了什么，一二的操作是修改p的前后节点，而p是需要插入的节点，所以还要修改的是q和原本q的后面一个节点

2 选D，队列为空的情况

3 选A，排除带括号的，括号外的为x

4 选D，8层的完全二叉树，第7层为2^6个节点，64个节点，非叶节点为54个，第8层为108个节点，前7层为2^7-1个，127个，共127+108=235，一个有几种情况？3种，7层的完全二叉树，一种234，快速方法，不用计算，因为肯定多于70多，又必是奇数

5 选B，送分题

6 选D，连通可能有回路，一般无向图不讨论拓扑排序

7 选A 0 99中进行查找49，24，11，5，2，0算出了六次，一共七次，用满二叉树来计算，一个6层的满二叉树为63个元素，7层的是127个元素

8 选C

9 选D，插入排序是不是？

10 选A，倒数第二个找到之后，最后一个数不需要进行排序

# 二简答题

1 见笔记本

2 递归调用自身的，堆栈

3 初始点，遍历方法，边的顺序

4 不细讲，算一下，选择排序是10000 + ···+9991，快速排序差不多为n\*Logn，即10000\*log10000，堆是log10000

# 三综合题

1 ABCFED | ABFCED从A开始，没有指向A的边，把图画出来即可

2 先从层序开始，根节点为A，在中序中C为A的左子树序列，DFBE是右子树序列。再看A的左子树DFBE在层序中B为第一个，所以B为子树的根节点，再回到中序，DF在B的前面，所以为左子树，E为右子树。再分析子树DF，DF在层序中D为第一个，所以D为根节点，而中序中F在D的右边所以F为D的右孩子

3 17填在位置5，27填在位置0

4 求数组的最大值，可以写得详细点

# 四、算法设计题

利用堆栈结构，我写是四不像，没有先后顺序，可以进行修改，改成前序

typedef int ElementType;//多余的一行，但能使程序可以编译

typedef struct BinaryNode

{

struct BinaryNode\*lchild;

ElementType data;

struct BinaryNode\*rchild;

}Node,\*BinaryTree;

Node\*searchBroNode(BinaryTree T,Node\*q)

{

Node\*Stack[100];

int top = -1;

Stack[++top] = T;

while(top != -1)

{

Node\*current = Stack[top--];

if(current->lchild == q)

return current->rchild;

else

if(current->lchild)

Stack[++top] = current->lchild;

if(current->rchild != q)

return current->lchild;

else

if(current->rchild)

Stack[++top] = current->rchild;

}

return NULL;

}

# 五、单项选择题

1.选B

2 选C，由于i会导致全部都变为double

3 选A

4 选C，A必须有输入项，B规定位数没用，输入项必须是变量，或者说地址

5 选D，可以这么想，循环继续进行下去的条件是s不等于100且k小于3，则结束条件就是s等于100或者k大于等于3

6 选A，j++是先执行完这条语句然后再j++，所以循环结束j大于等于4，j++就是5

7 选B，D还有什么反例？return

8 选B

9 选D

10 选C，“123456”可以看成const char \*

11 选C，A是比较地址，B是判断两个是否相等

12 选B，第一个都是参数的数量，与参数名无关，第二个参数是字符串数组，标准形式是char\*argv[ ]，等同于char\*\*

13 选A，所有变量都是传递值，但是传地址是传什么？传地址也是传地址的值，所以说传地址实际上也是传值

14 选D

15 选C，这是一个指针数组，数组存放的是int类型的指针，需要区别的是D，D是数组指针，所以选C，原因在于[ ]的优先级高于\*

16 选D，a三个元素分别的值是{1001，20}，{1002，19}，{1003，21}，只有第2个结构体的age是1002，所以是B，D，B是1001，D是正确答案

17 选C，强制类型转换需要带括号

18 选D，了解宏定义是什么，宏定义再预处理阶段做处理

19 选A，C最完善，但正常不需要这样

20 选B，指向结尾所以是a，同时是可读写所以是a+

# 六、填空题

1 1，1跟结合性有关，b=1，a=0，运算顺序，b--，重点a+b其中b已经减1了，然后是<=，!=，最后是||

2 5 来看一下f是在，干什么，大致跟长度有关，最后s指向字符串开头，p指向’\0’，用一个字符串长度为1的字符串作例子，则返回1，所以返回字符串长度

3 11 7 i=（2\*M）=>> i=(2\*N+1) =>> i=(2\*5+1)，所以i=11，

j=(1+1\*M) =>> j=(1+1\*5+1) =>> j=7

4 123456 这道题的疑问就是有没有空格

# 七、程序设计题

#include <stdio.h>

int main(int argc,char \*argv[])

{

int a,n;

scanf("%d %d",&a,&n);

int sum = 0;

int temp = 0;

int i;

for(i = 0;i < n;i++)

{

temp = temp \* 10 + a;

sum += temp;

}

printf("%d\n",sum);

return 0;

}

# 八、程序设计题