

# 练习题

全部试题答案均要求写在答卷纸上，写在试卷纸上一律无效

## 一、单项选择题（共10题，每题2分，共计20分。每题有且仅有一个正确答案）。

1. 设根节点深度为 0，一棵深度为  $h$  的满  $k$  ( $k>1$ ) 叉树，即除最后一层无任何子节点外，每一层上的所有结点都有  $k$  个子结点的树，共有（ ）个结点。
2. 微型计算机中，控制器的基本功能是（ ）。
  - A. 控制机器各个部件协调工作
  - B. 实现算术运算和逻辑运算
  - C. 存储各种控制信息
  - D. 获取外部信息
  - E. 存放程序和数据
3. 设栈S和队列Q的初始状态为空，元素 $e_1, e_2, e_3, e_4, e_5, e_6$ 依次通过栈S，一个元素出栈后即进入队列Q，若出队的顺序为 $e_2, e_4, e_3, e_6, e_5, e_1$ ，则栈S的容量至少应该为（ ）。
4. 完全二叉树共有2019个结点，则它的叶节点数是（ ）。
5. 将数组{8, 23, 4, 16, 77, -5, 53, 100}中的元素按从大到小的顺序排列，每次可以交换任意两个元素，最少需要交换（ ）次。
6. 设栈S的初始状态为空，元素a, b, c, d, e, f依次入栈S，出栈的序列为b, d, c, f, e, a，则栈S的容量至少应该是（ ）。
7. 与十进制数28.5625相等的四进制数是（ ）。
8. 小写字母“m”的十六进制的ASCII码值是（ ）。
9. 在有N个叶子节点的哈夫曼树中，其节点总数为（ ）。
10. 电线上停着两种鸟（A, B），可以看出两只相邻的鸟就将电线分为了一个线段。这些线段可分为两类：一类是两端的小鸟相同；另一类则是两端的小鸟不相同。已知：电线两个顶点上正好停着相同的小鸟，试问两端为不同小鸟的线段数目一定是（ ）。
  - A. 奇数
  - B. 偶数
  - C. 可奇可偶
  - D. 数目固定

二、不定项选择题（共5题，每题2分，共10分。正确答案个数大于等于1。多选或少选均不得分）。

1. 在下列关于图灵奖的说法中，正确的有（ ）。
  - A. 图灵奖是美国计算机协会于1966年设立的，专门奖励那些对计算机事业作出重要贡献的个人
  - B. 图灵奖有“计算机界诺贝尔奖”之称
  - C. 迄今为止，还没有华裔计算机科学家获此殊荣
  - D. 图灵奖的名称取自计算机科学的先驱、英国科学家阿兰·图灵
2. 计算机在工作过程中，若突然停电，（ ）中的信息不会丢失。
  - A. 硬盘
  - B. CPU
  - C. ROM
  - D. RAM
3. 设 $A=\text{true}$ ,  $B=\text{false}$ ,  $C=\text{true}$ ,  $D=\text{false}$ ，以下逻辑运算表达式值为真的有（ ）。
  - A.  $(A \wedge B) \vee (C \wedge D \vee \neg A)$
  - B.  $((\neg A \wedge B) \vee C) \wedge \neg D$
  - C.  $(B \vee C \vee D) \vee D \wedge A$
  - D.  $A \wedge (D \vee \neg C) \wedge B$
4. 二叉树T，已知其先根遍历是1 2 4 3 5 7 6（数字为结点的编号，以下同），后根遍历是4 2 7 5 6 3 1，则该二叉树的可能的中根遍历是（ ）。
  - A. 4 2 1 7 5 3 6
  - B. 2 4 1 7 5 3 6
  - C. 4 2 1 7 5 6 3
  - D. 2 4 1 5 7 3 6
5. 2-3 树是一种特殊的树，它满足两个条件：
  - (1) 每个内部结点有两个或三个子结点；
  - (2) 所有的叶结点到根的路径长度相同。如果一棵 2-3 树有 10 个叶结点，那么它可能有（ ）个非叶结点。
  - A. 5
  - B. 6
  - C. 7
  - D. 8

三. 问题求解（共2题，每题5分，共计10分）

1. 如果在三维直角坐标系里任取  $n$  个整点（坐标都是整数），其中一定存在两个点，它们连线的中点也是整点，那么  $n$  至少是几？ \_\_\_\_\_
2. （取石子游戏）现有 5 堆石子，石子数依次为 3, 5, 7, 19, 50，甲乙两人轮流从任一堆中任取（每次只能取自一堆，不能不取），取最后一颗石子的一方获胜。甲先取，问甲有没有获胜策略（即无论乙怎样取，甲只要不失误，都能获胜）？如果有，甲第一步应该在哪一堆里取多少？请写出你的结果： \_\_\_\_\_。

四. 阅读程序写结果 (共 4 题, 每题 8 分, 共计 32 分)

1.

```
#include<iostream>
using namespace std;
const int N=1009;
int n,q[N],x[N];
int main(){
    cin>>n;
    for(int i=1;i<=n;i++)cin>>x[i];
    for(int i=1;i<=n;i++){
        for(int j=i;j>=x[i]+1;j--)
            q[j]=q[j-1];
        q[x[i]]=i;
    }
    for(int i=1;i<=n;i++)cout<<q[i]<<" ";
    cout<<endl;
    return 0;
}
```

输入: 8 1 1 2 3 1 1 2 3

输出: \_\_\_\_\_

2.

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin>>s;
    while(1){
        int i,j;
        for(i=1;i<s.size();i++)
            if(s[i]>='a'&& s[i]<='z'&& s[i]==s[i-1])break;
        if(i==s.size())break;
        for(j=i;j<s.size();j++)
            if(s[j]!=s[j-1])break;
        string t="";
        if(j<s.size()) t=s.substr(j);
        t+=s[i]-'a'+'A';
        t+='0'+j-i+1;
    }
}
```

```

        t+=s.substr(0,i-1);
        s=t;
    }
    cout<<s<<endl;
    return 0;
}
输入:  bxttttfu
输出:  _____

```

3.

```

#include<iostream>
using namespace std;
const int V=100;
int n,m,ans,e[V][V];
bool visited[V];

void dfs(int x,int len){
    int i;
    visited[x]= true;
    if(len>ans)
        ans=len;
    for(i=1;i<=n;i++)
        if( (!visited[i]) && (e[x][i]!=-1) )
            dfs(i,len+e[x][i]);
    visited[x]=false;
}

int main(){
    int i,j,a,b,c;
    cin>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            e[i][j]=-1;
    for(i=1;i<=m;i++) {
        cin>>a>>b>>c;
        e[a][b]=c;
        e[b][a]=c;
    }
}

```

```

        for(i=1;i<=n;i++)
            visited[i]=false;
        ans=0;
        for(i=1;i<=n;i++)
            dfs(i,0);
        cout<<ans<<endl;
        return 0;
    }

```

输入：

4 6

1 2 10

2 3 20

3 4 30

4 1 40

1 3 50

2 4 60

输出： \_\_\_\_\_

4.

```

#include<iostream>
using namespace std;
typedef unsigned int ui;
ui work(ui x){
    ui s = x & (-x);
    ui r = s + x;
    ui n = r | (((x ^ r) >> 2) / s);
    return n;
}
int main(){
    ui n,k;
    cin>>n>>k;
    for(int i=1;i<=k;i++)
        n=work(n);
    cout<<n;
    return 0;
}

```

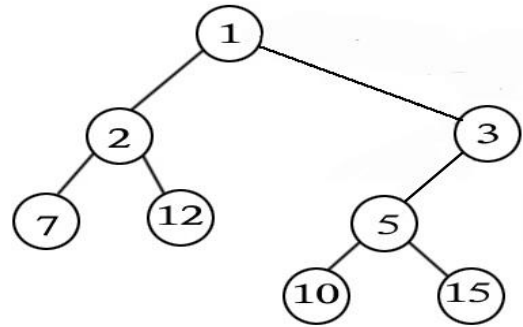
输入： 3 14

输出： \_\_\_\_\_

## 五. 完善程序 (前 6 空, 每空 3 分, 后 5 空, 每空 2 分, 共 28 分)

1. (笛卡尔树) 对于一个给定的两两不等的正整数序列, 笛卡尔树是这样的一棵二叉树: 首先, 它是一个最小堆, 即除了根结点, 每个节点的权值都大于父节点的权值; 其次, 它的中序遍历恰好就是给定的序列。例如, 对于序列 7、2、12、1、10、5、15、3, 下图就是一棵对应的笛卡尔树。现输入序列的规模  $n$  ( $1 \leq n < 100$ ) 和序列的  $n$  个元素, 试求其对应的笛卡尔树的深度  $d$  (根节点深度为 1), 以及有多少个叶子节点的深度为  $d$ 。

```
#include<iostream>
using namespace std;
const int SIZE=100+5;
const int INFINITY=1000000;
int n,a[SIZE],maxDeep,num;
void solve(int left,int right,int deep){
    int i,j,min;
    if(deep>maxDeep){
        maxDeep=deep;
        (0) _____;
    }
    else if(deep==maxDeep)
        (1) _____ ;
    min= INFINITY;
    for(i=left;i<=right;i++){
        if(min>a[i]){
            min=a[i];
            (2) _____ ;
        }
    }
    if(left<j)
        (3) _____ ;
    if(j<right)
        (4) _____ ;
}
int main(){
    int i;
    cin>>n;
    for(i=1;i<=n;i++)    cin>>a[i];
    maxDeep=0;
    _____ (5) _____
    cout<<maxDeep<<' '<<num<<endl;
    return 0;
}
```



2. (矩阵中的数字) 有一个  $n \times n$  ( $1 \leq n \leq 5000$ ) 的矩阵  $a$ , 对于  $1 \leq i < n, 1 \leq j \leq n$ ,  $a[i, j] < a[i + 1, j]$   $a[j, i] < a[j, i + 1]$ 。即矩阵中左右相邻的两个元素, 右边的元素一定比左边的大。上下相邻的两个元素, 下面的元素一定比上面的大。给定矩阵  $a$  中的一个数字  $k$ , 找出  $k$  所在的行列 (注意: 输入数据保证矩阵中的数各不相同)。

```
#include <iostream>
using namespace std;
int n,k,answerx,answery;
int a[5001][5001];

void FindKPosition(){
    int i = n,j = n;
    while (j > 0) {
        if (a[n][j] < k) break;
        j --;
    }
    ①
    while (a[i][j] != k) {
        while ( ② && i > 1) i --;
        while ( ③ && j <= n) j ++;
    }
    ④
    ⑤
}

int main(){
    int i,j;
    cin >> n;
    for (i = 1;i <= n;i ++)
        for (j = 1;j <= n;j ++)
            cin >> a[i][j];
    cin >> k;
    FindKPosition();
    cout << answerx << " " << answery << endl;
    return 0;
}
```