

第十四届全国青少年信息学奥林匹克联赛初赛试题

(提高组 C++ 语言 二小时完成)

●● 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1.5分，共计15分。每题有且仅有一个正确答案)。

1. 在以下各项中，()不是操作系统软件。

- A. Solaris B. Linux C. Sybase D. Windows Vista E. Symbian

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. 设字符串S="Olympic"，S的非空子串的数目是()。

- A. 29 B. 28 C. 16 D. 17 E. 7

4. 完全二叉树共有 $2^N - 1$ 个结点，则它的叶节点数是()。

- A. $N - 1$ B. 2^N C. N D. $2^N - 1$ E. $N/2$

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。

- A. 4 B. 5 C. 6 D. 7 E. 8

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()。

- A. 6 B. 5 C. 4 D. 3 E. 2

7. 与十进制数28.5625相等的四进制数是()。

- A. 123.21 B. 131.22 C. 130.22 D. 130.21 E. 130.20

8. 递归过程或函数调用时，处理参数和返回地址，通常使用一种称为()的数据结构。

- A. 队列 B. 多维数组 C. 线性表 D. 链表 E. 栈

9. TCP/IP是一组构成互联网基础的网络协议，字面上包括两组协议：传输控制协议(TCP)和网络协议(IP)。TCP/IP协议把Internet网络系统描述成具有四个层次功能的网络模型，其中提供源节点和目的节点之间的信息传输服务，包括寻址和路由器选择等功能的是()。

- A. 链路层 B. 网络层 C. 传输层 D. 应用层 E. 会话层

话层

10. 对有序数组{5, 13, 19, 21, 37, 56, 64, 75, 88, 92, 100}进行二分查找, 等概率的情况下查找成功的平均查找长度(平均比较次数)是()。

- A. 35/11 B. 34/11 C. 33/11 D. 32/11 E. 34/10

二、不定项选择题 (共10题, 每题1.5分, 共计15分。每题正确答案的个数大于或等于1。多选或少选均不得分)。

11. 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

12. 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

13. 设A=true, B=false, C=true, D=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. $A \wedge (D \vee \neg C) \wedge B$

14. Web2.0是近年来互联网的热门概念之一, 其核心思想是互动与分享。下列网站中, () 是典型的Web2.0应用。

- A. Sina B. Flickr C. Yahoo D. Google

15. $(2008)_{10} + (5B)_{16}$ 的结果是()。

- A. $(8313)_6$ B. $(2099)_9$ C. $(40B3)_8$ D. $(100001100011)_2$

16. 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

17. 面向对象程序设计(Object-Oriented Programming)是一种程序设计的方法论, 它将对象作为程序的基本单元, 将数据和程序封装在对象中, 以提高软件的重用性、灵活性和扩展性。下面关于面向对象程序设计的说法中, 正确的是()。

- A. 面向对象程序设计通常采用自顶向下设计方法进行设计。
B. 面向对象程序设计方法具有继承性(inheritance)、封装性(encapsulation)、多态性(polymorphism)等几大特点。
C. 支持面向对象特性的语言称为面向对象的编程语言, 目前较为流行的有C++、JAVA、C#等。
D. 面向对象的程序设计的雏形来自于Simula语言, 后来在SmallTalk语言的完善和标准化的过程中得到更多的扩展和对以前思想的重新注解。至今, SmallTalk语言仍然被视为面向对象语言的

基础。

18. 设T是一棵有n个顶点的树, 下列说法正确的是 ()。

- A. T是连通的、无环的
- B. T是连通的, 有n-1条边
- C. T是无环的, 有n-1条边
- D. 以上都不对

19. NOIP竞赛推荐使用的语言环境有 ()。

- A. Dev-C++
- B. Visual C++
- C. free pascal
- D. Lazarus

20. 在下列防火墙 (firewall) 的说法中, 正确的有 ()。

- A. 防火墙是一项协助确保信息安全的设备, 其会依照特定的规则, 允许或是限制数据通过
- B. 防火墙可能是一台专属的硬件或是安装在一般硬件上的一套软件
- C. 网络层防火墙可以视为一种 IP 数据包过滤器, 只允许符合特定规则的数据包通过, 其余的一概禁止穿越防火墙
- D. 应用层防火墙是在 TCP/IP的“应用层”上工作, 可以拦截进出某应用程序的所有数据包

三. 问题求解 (共2题, 每题5分, 共计10分)

1. 有6个城市, 任何两个城市之间都有一条道路连接, 6个城市两两之间的距离如下表所示, 则城市1到城市6的最短距离为_____。

	城市1	城市2	城市3	城市4	城市5	城市6
城市1	0	2	3	1	12	15
城市2	2	0	2	5	3	12
城市3	3	2	0	3	6	5
城市4	1	5	3	0	7	9
城市5	12	3	6	7	0	2
城市6	15	12	5	9	2	0

2. 书架上有21本书, 编号从1到21, 从其中选4本, 其中每两本的编号都不相邻的选法一共有_____种。

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;
int main()
{
    int i, a, b, c, d, f[4];
    for(i = 0; i < 4; i++) cin >> f[i];
```

```

    a = f[0] + f[1] + f[2] + f[3];
    a = a / f[0];
    b = f[0] + f[2] + f[3];
    b = b / a;
    c = (b * f[1] + a) / f[2];
    d = f[(b / c) % 4];
    if(f[(a + b + c + d) % 4] > f[2])
        cout << a + b<< endl;
    else cout << c + d << endl;
    return 0;
}

```

输入 : 9 19 29 39

输出 : _____

```

2. #include<iostream>
using namespace std;
void foo(int a, int b, int c)
{
    if(a > b)
        foo(c, a, b);
    else
        cout<<a<<','<<b<<','<<c<<endl;
}
int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    foo(a, b, c);
    return 0;
}

```

输入 : 2 1 3

输出 : _____

```

3. #include<iostream>
using namespace std;
void f(int a, int b, int c)
{
    cout << a << b << c << ' '//';
}

```

```

    if(a == 3 && b == 2 && c == 1)
        return;
    if(b < c)
        f(a, c, b);
    else if(a < b)
    {
        if(a < c)
            f(c, a, b);
        else
            f(b, c, a);
    }
}

```

```

int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    f(a, b, c);
    cout << endl;
    return 0;
}

```

输入： 1 3 2

输出： _____

4. #include <iostream>

```

#include <cstring>
using namespace std;
int i,j,len;
char s[50];

```

```

int main()
{
    cin >> s;
    len = strlen(s);
    for (i = 0; i < len; ++i)
    {
        if (s[i] >= 'A' && s[i] <= 'Z') s[i] -= 'A' - 'a';
    }
}

```

```

}
for (i = 0;i < len; ++i)
{
    if (s[i] < 'x') s[i] += 3; else s[i] += -23;
}
cout << s << '/';
for (j = 1;j < 4;j ++)
{
    for (i = 0;i < len-j; i = i + j)
    {
        s[i] = s[i + j] ;
    }
}
cout << s << endl;
return 0;
}

```

输入：ABCDEFGGuvvwxyz

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．（找第k大的数） 给定一个长度为1,000,000的无序正整数序列，以及另一个数n(1<=n<=1000000)，接下来以类似快速排序的方法找到序列中第n大的数（关于第n大的数：例如序列{1, 2, 3, 4, 5, 6}中第3大的数是4）。

```

#include <iostream>
using namespace std;

int a[1000001],n,ans = -1;
void swap(int &a,int &b)
{
    int c;
    c = a; a = b; b = c;
}

int FindKth(int left, int right, int n)
{
    int tmp,value,i,j;

```

```

if (left == right) return left;
tmp = rand()%(right - left) + left;
swap(a[tmp],a[left]);
value = _____①
i = left;
j = right;
while (i < j)
{
    while (i < j && _____②) j --;
    if (i < j) {a[i] = a[j]; i ++;} else break;
    while (i < j && _____③) i ++;
    if (i < j) {a[j] = a[i]; j --;} else break;
}
_____④
if (i < n) return FindKth(_____⑤);
if (i > n) return _____⑥;
return i;
}

int main()
{
    int i;
    int m = 1000000;
    for (i = 1;i <= m;i ++)
        cin >> a[i];
    cin >> n;
    ans = FindKth(1,m,n);
    cout << a[ans];
    return 0;
}

```

2. (矩阵中的数字) 有一个 $n \times n$ ($1 \leq n \leq 100$) 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;
}

```