

第十四届全国青少年信息学奥林匹克联赛初赛试题 2008

( 普及组 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 20 题, 每题 1.5 分, 共计 30 分。每题有且仅有一个正确答案.)。

1. 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. 获取外部信息      D. 存放程序和数据
2. 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是 ( )。  
A.  $(A \wedge B) \vee (C \wedge D \vee \neg A)$       B.  $((\neg A \wedge B) \vee C) \wedge \neg D$   
C.  $(B \vee C \vee D) \wedge D \wedge A$       D.  $A \wedge (D \vee \neg C) \wedge B$
3. 在下列关于图灵奖的说法中, 不正确的是 ( )。  
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
4. 计算机在工作过程中, 若突然停电, ( ) 中的信息不会丢失。  
A. ROM 和 RAM      B. CPU      C. ROM      D. RAM
5. 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.  $N$       C.  $2*N$       D.  $2^{N-1}$
6. 在以下各项中, ( ) 不是操作系统软件。  
A. Solaris      B. Linux      C. Windows Vista      D. Sybase
7. 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, f, e, c, a, 则栈 S 的容量至少应该是 ( )。  
A. 6      B. 5      C. 4      D. 3
8. 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
9. 设字符串  $S = \text{"Olympic"}$ , S 的非空子串的数目是 ( )。  
A. 28      B. 29      C. 16      D. 17
10. 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

11. 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. 栈

12.  $(2008)_{10} + (5B)_{16}$  的结果是（ ）。

- A.  $(833)_{16}$                       B.  $(2089)_{10}$                       C.  $(4163)_8$                       D.  $(100001100011)_2$

13. 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），中根遍历是 2 4 1 5 7 3 6，则该二叉树的后根遍历是（ ）。

- A. 4 2 5 7 6 3 1                      B. 4 2 7 5 6 3 1  
C. 7 4 2 5 6 3 1                      D. 4 2 7 6 5 3 1

14. 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

15. 对有序数组 {5, 13, 19, 21, 37, 56, 64, 75, 88, 92, 100} 进行二分查找，成功查找元素 19 的查找长度（比较次数）是（ ）。

- A. 1                      B. 2                      C. 3                      D. 4

16. 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，不正确的是（ ）。

- 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。

17. 在  $32 \times 32$  点阵的“字库”中，汉字“北”与“京”的字模占用字节数之和是（ ）。

- A. 512                      B. 256                      C. 384                      D. 128

18. 设 T 是一棵有 n 个顶点的树，下列说法不正确的是（ ）。

- A. T 有 n 条边                      B. T 是连通的  
C. T 是无环的                      D. T 有 n-1 条边

19. 下列不属于 NOIP 竞赛推荐使用的语言环境的是（ ）。

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

Lazarus

20. 在 C++ 程序中，表达式  $200|10$  的值是 (     )

- A. 20                      B. 1                      C. 220                      D. 202

## 二. 问题求解 (共 2 题，每题 5 分，共计 10 分)

1. 书架上有 4 本不同的书 A、B、C、D。其中 A 和 B 是红皮的，C 和 D 是黑皮的。把这 4 本书摆在书架上，满足所有黑皮的都排在一起的摆法有\_\_\_\_\_种。满足 A 必须比 C 靠左，所有红皮的都要摆放在一起，所有黑皮的都要摆放在一起，共有\_\_\_\_\_种摆法。

2. 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

## 三. 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: 3 1 2

输出: \_\_\_\_\_

```

3. #include <iostream>
using namespace std;

void func(int ary[], int n )
{
    int i=0, j, x;
    j=n-1;
    while(i<j)

```

```

    {
        while (i<j&&ary[i]>0) i++;
        while (i<j&&ary[j]<0) j--;
        if (i<j){
            x=ary[i];
            ary[i++]=ary[j];
            ary[j--]=x;
        }
    }
}

```

```

int main()
{

    int a[20], i, m;
    m=10;
    for(i=0; i<m; i++)
    {
        cin>>a[i];
    }
    func(a, m);
    for (i=0; i<m; i++)
        cout<<a[i]<<" ";
    cout<< endl;
    return 0;
}

```

输入: 5 4 -6 -11 6 -59 22 -6 1 10

输出: \_\_\_\_\_

```

4. #include<iostream>
    #include<cstring>
using namespace std;

```

```

#define MAX 100

```

```

void solve(char first[], int spos_f, int epos_f, char mid[], int spos_m,
int epos_m)
{
    int i, root_m;
    if(spos_f > epos_f)
        return;
    for(i = spos_m; i <= epos_m; i++)
        if(first[spos_f] == mid[i])
        {
            root_m = i;
            break;
        }
    solve(first, spos_f + 1, spos_f + (root_m - spos_m), mid, spos_m,
root_m - 1);
    solve(first, spos_f + (root_m - spos_m) + 1, epos_f, mid, root_m + 1,
epos_m);
    cout << first[spos_f];
}

int main()
{
    char first[MAX], mid[MAX];
    int len;
    cin >> len;
    cin >> first >> mid;
    solve(first, 0, len - 1, mid, 0, len - 1);
    cout << endl;
    return 0;
}

```

输入： 7

ABDCEGF

BDAGECF

输出： \_\_\_\_\_

#### 四. 完善程序（前 4 空，每空 2.5 分，后 6 空，每空 3 分，共 28 分）

1. （字符串替换）给定一个字符串 S（S 仅包含大小写字母），下面的程序将 S 中的每个字母用规定的字母替换，并输出 S 经过替换后的结果。程序的输入是两个字符串，第一个字符串是给定的字符串 S，第二个字符串 S' 由 26 个字母组成，它是 a-z 的任一排列，大小写不定，S' 规定了每个字母对应的替换字母：S' 中的第一个字母是字母 A 和 a 的替换字母，即 S 中的 A 用该字母的大写替换，S 中的 a 用该字母的小写替换；S' 中的第二个字母是字母 B 和 b 的替换字母，即 S 中的 B 用该字母的大写替换，S 中的 b 用该字母的小写替换；…… 以此类推。

```
#include <iostream>
#include <string.h>
char change[26], str[5000];
using namespace std;

void CheckChangeRule()
{
    int i;
    for (i = 0; i < 26; i++)
    {
        if ( _____ ① )
            change[i] -= 'A' - 'a';
    }
}

void ChangeString()
{
    int i;
    for (i = 0; i < strlen(str); i++)
    {
        if ( _____ ② )
            str[i] = change[str[i] - 'A'] - 'a' + 'A';
        else
            _____ ③
    }
}
```

```

}

int main()
{
    int i;
    cin >> str ;
    cin >> change;
    CheckChangeRule();
    _____④_____
    cout << str << endl;
    return 0;
}

```

2. **（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 ( $1 \leq n \leq 1000000$ )，然后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;
}

```

## 参考答案

### 一. 选择题

ABCCB DCDAB DABBB ABABD

### 二. 问题解答

1. 12 4

2. 7

### 三. 阅读程序

1. 23

2. 2.,3,1

3. 5 4 10 1 6 22 -59 -6 -11 -6

4. DBGEFCA

### 四. 完善程序

1.

(1) `change[i] >= 'A' && change[i] <= 'Z'`(只写 `change[i] <= 'Z'`也对)

(2) `str[i] >= 'A' && str[i] <= 'Z'` (只写 `str[i] <= 'Z'`也对)

(3) `str[i] = change[str[i] - 'a'];`

(4) `ChangeString();`

2.

(1) `a[left];`

(2) `a[j] < value`

(3) `a[i] > value`

(4) `a[i] = value;`

(5) `i + 1, right, n`

(6) `FindKth(left, i - 1, n);`