第二十二届全国青少年信息学奥林匹克联赛初赛

普及组 C++语言试题

竞赛时间：2016 年 10 月 22 日 14:30~16:30

选手注意：

 试题纸共有 9 页，答题纸共有 2 页，满分 100 分。请在答题纸上作答，写在 试题纸上的一律无效。

 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资

料。

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选 项）

**1.** 以下不是微软公司出品的软件是（ ）。

|  |  |  |  |
| --- | --- | --- | --- |
| A. | Powerpoint | B. | Word |
| C. | Excel | D. | Acrobat Reader |

**2.** 如果 256 种颜色用二进制编码来表示，至少需要（ ）位。

A. 6 B. 7 C. 8 D. 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3.** 以下不属于无线通信技术的是（ | ）。 |  | | |
| A. 蓝牙 B. WiFi | C. | GPRS | D. | 以太网 |

**4.** 以下不是 CPU 生产厂商的是（ ）。

A. Intel B. AMD C. Microsoft D. IBM

**5.** 以下不是存储设备的是（ ）。

A. 光盘 B. 磁盘 C. 固态硬盘 D. 鼠标

**6.** 如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照 CapsLock、 字母键 A、字母键 S 和字母键 D 的顺序循环按键，即 CapsLock、A、S、D、 CapsLock、A、S、D、……，屏幕上输出的第 81 个字符是字母（ ）。

A. A B. S C. D D. a

**7.** 二进制数 00101100 和 00010101 的和是（ ）。

A. 00101000 B. 01000001 C. 01000100 D. 00111000

**8.** 与二进制小数 0.1 相等的八进制数是（ ）。

A. 0.8 B. 0.4 C. 0.2 D. 0.1

**9.** 以下是 32 位机器和 64 位机器的区别的是（ ）。

|  |  |  |  |
| --- | --- | --- | --- |
| A. | 显示器不同 | B. | 硬盘大小不同 |
| C. | 寻址空间不同 | D. | 输入法不同 |

**10.** 以下关于字符串的判定语句中正确的是（ ）。

|  |  |  |  |
| --- | --- | --- | --- |
| A. | 字符串是一种特殊的线性表 | B. | 串的长度必须大于零 |
| C. | 字符串不可以用数组来表示 | D. | 空格字符组成的串就是空串 |

**11.** 一棵二叉树如右图所示，若采用顺序存储结构，即用一 维数组元素存储该二叉树中的结点（根结点的下标为 1， 若某结点的下标为 i ，则其左孩子位于下标 2i 处、右孩

子位于下标(2i+1)处），则图中所有结点的最大下标为

（ ）。

A. 6 B. 10 C. 12 D. 15

**12.** 若有如下程序段，其中 s、a、b、c 均已定义为整型变量，且 a、c 均已赋值

（c 大于 0）。

s = a;

for (b = 1; b <= c; b++)

s = s + 1;

则与上述程序段修改 s 值的功能等价的赋值语句是（ ）。

A. s = a + b; B. s = a + c; C. s = s + c; D. s = b + c;

**13.** 有以下程序：

#include <iostream>

using namespace std;

int main() {

int k = 4, n = 0;

while (n < k) {

n++;

if (n % 3 != 0)

continue;

k--;

}

cout << k << "," << n << endl;

return 0;

}

程序运行后的输出结果是（ ）。

A. 2,2 B. 2,3 C. 3,2 D. 3,3

**14.** 给定含有 n 个不同的数的数组 L=<x1, x2, ..., xn>。如果 L 中存在 x（i

1 < i < n）

使得 x1 < x2 < ... < xi-1 < xi > xi+1 > ... > xn， 则称 L 是单峰的，并称 xi 是 L 的

“峰顶”。现在已知 L 是单峰的，请把 a-c 三行代码补全到算法中使得算法正确找到 L 的峰顶。

a. Search(k+1, n) b. Search(1, k-1) c. return L[k]

Search(1, n)

1. k←⌊n/2⌋

2. if L[k] > L[k-1] and L[k] > L[k+1]

3. then

4. else if L[k] > L[k-1] and L[k] < L[k+1]

5. then

6. else

正确的填空顺序是（ ）。

A. c, a, b B. c, b, a C. a, b, c D. b, a, c

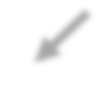
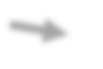
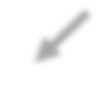
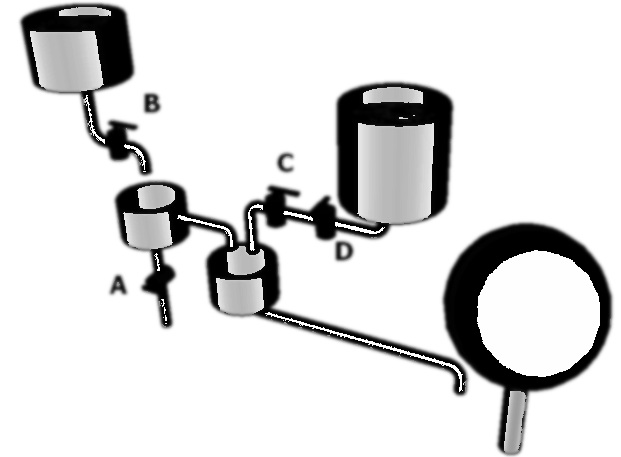
**15.** 设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（ ）个顶 点。

A. 10 B. 12 C. 8 D. 16

**16.** 有 7 个一模一样的苹果，放到 3 个一样的盘子中，一共有（ ）种放法。

A. 7 B. 8 C. 21 D. 37

**17.** 下图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开 或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的



有水 有水

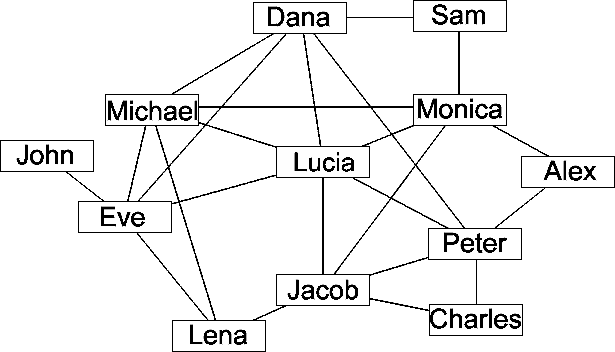
果树

是（ ）。

|  |  |  |  |
| --- | --- | --- | --- |
| A. | B 打开，其他都关上 | B. | AB 都打开，CD 都关上 |
| C. | A 打开，其他都关上 | D. | D 打开，其他都关上 |

**18.** Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他们之间的关系图，两个人之间有边相连代表这两个人是朋友，没有边相连代 表不是朋友。这个社交网站的规则是：如果某人 A 向他（她）的朋友 B 分 享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那 么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对 该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上 传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友（ ）分享该照片。

|  |  |  |  |
| --- | --- | --- | --- |
| A. | Dana, Michael, Eve | B. | Dana, Eve, Monica |
| C. | Michael, Eve, Jacob | D. | Micheal, Peter, Monica |



**19.** 周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责 切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜 10 分钟，然后切 菜 10 分钟，最后炒菜 10 分钟。那么做一道菜需要 30 分钟。注意：两道不

同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗， 也不能同时切。那么做完三道菜的最短时间需要（ ）分钟。

A. 90 B. 60 C. 50 D. 40

**20.** 参加 NOI 比赛，以下不能带入考场的是（ ）。

A. 钢笔 B. 适量的衣服 C. U 盘 D. 铅笔

二、问题求解（共 **2** 题，每题 **5** 分，共计 **10** 分；第一题全部答对得 **5** 分，没有 部分分；第二题第一空 **2** 分，第二空 **3** 分）

**1.** 从一个 4×4 的棋盘（不可旋转）中选取不在同一行也不在同一列上的两个 方格，共有 种方法。

**2.** 约定二叉树的根节点高度为 1。一棵结点数为 2016 的二叉树最少有

个叶子结点；一棵结点数为 2016 的二叉树最小的高度值是

。

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 max, min, sum, count = 0;

int tmp;

cin >> tmp;

if (tmp == 0)

return 0;

max = min = sum = tmp;

count++;

while (tmp != 0) {

cin >> tmp;

if (tmp != 0) { sum += tmp; count++;

if (tmp > max)

max = tmp; if (tmp < min) min = tmp;

}

}

cout << max << "," << min << "," << sum / count << endl;

return 0;

}

输入: 1 2 3 4 5 6 0 7

输出：

**2.** #include <iostream>

using namespace std;

int main() {

int i = 100, x = 0, y = 0;

while (i > 0) {

i--;

x = i % 8;

if (x == 1)

y++;

}

cout << y << endl;

return 0;

}

输出：

**3.** #include <iostream>

using namespace std;

int main() {

int a[6] = {1, 2, 3, 4, 5, 6};

int pi = 0; int pj = 5; int t , i;

while (pi < pj) { t = a[pi]; a[pi] = a[pj]; a[pj] = t; pi++;

pj--;

}

for (i = 0; i < 6; i++)

cout << a[i] << ",";

cout << endl;

return 0;

}

输出：

**4.** #include <iostream>

using namespace std;

int main() {

int i, length1, length2;

string s1, s2;

s1 = "I have a dream.";

s2 = "I Have A Dream."; length1 = s1.size(); length2 = s2.size();

for (i = 0; i < length1; i++)

if (s1[i] >= 'a' && s1[i] <= 'z')

s1[i] -= 'a' - 'A';

for (i = 0; i < length2; i++)

if (s2[i] >= 'a' && s2[i] <= 'z')

s2[i] -= 'a' - 'A';

if (s1 == s2)

cout << "=" << endl;

else if (s1 > s2)

cout << ">" << endl;

else

cout << "<" << endl;

return 0;

}

输出：

四、完善程序（共 **2** 题，每题 **14** 分，共计 **28** 分）

**1.** （读入整数）请完善下面的程序，使得程序能够读入两个 int 范围内的整数， 并将这两个整数分别输出，每行一个。（第一、五空 2.5 分，其余 3 分）

输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。 例如：

输入：

输出：

123 -789

123

-789

#include <iostream>

using namespace std;

int readint() {

int num = 0; // 存储读取到的整数

int negative = 0; // 负数标识

char c; // 存储当前读取到的字符

c = cin.get();

while ((c < '0' || c > '9') && c != '-')

c = (1) ;

if (c == '-')

negative = 1;

else

(2) ;

c = cin.get();

while ( (3) ) { (4) ;

c = cin.get();

}

if (negative == 1) (5) ;

return num;

}

int main() {

int a, b;

a = readint();

b = readint();

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

return 0;

}

**2.** （郊游活动）有 n 名同学参加学校组织的郊游活动，已知学校给这 n 名同学 的郊游总经费为 A 元，与此同时第 i 位同学自己携带了 Mi 元。为了方便郊 游，活动地点提供 B(≥n)辆自行车供人租用，租用第 j 辆自行车的价格为 Cj 元，每位同学可以使用自己携带的钱或者学校的郊游经费，为了方便账务管 理，每位同学只能为自己租用自行车，且不会借钱给他人，他们想知道最多 有多少位同学能够租用到自行车。（第四、五空 2.5 分，其余 3 分）

本题采用二分法。对于区间[l, r]，我们取中间点 mid 并判断租用到自行 车的人数能否达到 mid。判断的过程是利用贪心算法实现的。

#include <iostream>

using namespace std;

#define MAXN 1000000

int n, B, A, M[MAXN], C[MAXN], l, r, ans, mid;

bool check(int nn) {

int count = 0, i, j;

i = (1) ;

j = 1;

while (i <= n) {

if ( (2) )

count += C[j] - M[i];

i++;

j++;

}

return (3) ;

}

void sort(int a[], int l, int r) {

int i = l, j = r, x = a[(l + r) / 2], y;

while (i <= j) {

while (a[i] < x) i++; while (a[j] > x) j--; if (i <= j) {

y = a[i]; a[i] = a[j]; a[j] = y;

i++; j--;

}

}

if (i < r) sort(a, i, r);

if (l < j) sort(a, l, j);

}

int main() {

int i;

cin >> n >> B >> A;

for (i = 1; i <= n; i++)

cin >> M[i];

for (i = 1; i <= B; i++)

cin >> C[i]; sort(M, 1, n); sort(C, 1, B);

l = 0;

r = n;

while (l <= r) {

mid = (l + r) / 2; if ( (4) ) { ans = mid;

l = mid + 1;

} else

r = (5) ;

}

cout << ans << endl;

return 0;

}