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

_)A-:	—	沣	⇉	
)比·	±١	/-	慐	٠
~~	J	ᅩ	NEW.	٠

A. 0.8

•	试题纸共有9页, 试题纸上的一律; 不得使用任何电- 料。	无效。					
一、 项)	单项选择题(共	20 題	返,每题 1.5 分,	共	计 30 分;每题有	有且 个	仅有一个正确选
Α	以下不是微软公 . Powerpoint . Excel	司出占	品的软件是() B. D.	Word Acrobat Reader		
	如果 256 种颜色) . 6	用二j B.		, <u> </u>		位。 D.	9
	以下不属于无线; . 蓝牙		技术的是() WiFi	° C.	GPRS	D.	以太网
	以下不是 CPU 生 . Intel		商的是() AMD	°C.	Microsoft	D.	IBM
	以下不是存储设 [。] . 光盘		是()。磁盘	C.	固态硬盘	D.	鼠标
6.	如果开始时计算 字母键 A、字母链 CapsLock、A、S、	建S禾	印字母键 D 的顺	序循	盾环按键,即 Ca _l	psLo	ck、A、S、D、
Α	. A	В.	S	C.	D	D.	a
7.	二进制数 001011	00 和	00010101 的和	是	() 。		
Α	. 00101000	В.	01000001	C.	01000100	D.	00111000
8.	与二进制小数 0.1	相等	等的八进制数是	()。		

C. 0.2

B. 0.4

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( )。
2,2
                            C. 3.2
             B. 2,3
```

D. 3,3

14. 给定含有 n 个不同的数的数组 L= $\langle x_1, x_2, ..., x_n \rangle$ 。如果 L 中存在 $x_i(1 < i < n)$ 使得 $x_1 < x_2 < ... < x_{i-1} < x_i > x_{i+1} > ... > x_n$, 则称 L 是单峰的,并称 x_i 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。

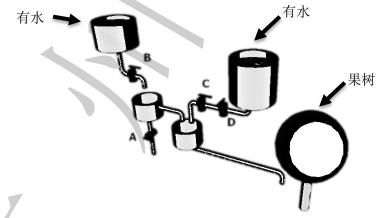
- a. Search(k+1, n)
- b. Search(1, k-1)
- c. return L[k]

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()。

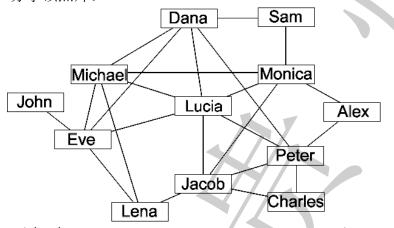
- A. c, a, b
- B. c, b, a
- C. a, b,
- D. b, a, c
- **15.** 设简单无向图 G 有 16 条边且每个顶点的度数都是 2,则图 G 有 ()个顶点。
 - A. 10
- B. 12
- C. 8
- D. 16
- 16. 有7个一模一样的苹果,放到3个一样的盘子中,一共有()种放法。
 - A. 7
- B. 8
- C. 21
- D. 3
- 17. 下图表示一个果园灌溉系统,有 A、B、C、D 四个阀门,每个阀门可以打开或关上,所有管道粗细相同,以下设置阀门的方法中,可以让果树浇上水的



是()。

- A. B 打开, 其他都关上
- C. A打开,其他都关上
- B. AB 都打开, CD 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.	<pre>#include <iostream> using namespace std;</iostream></pre>
	<pre>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)</pre>
	} cout << max << "," << min << "," << sum / count << endl; return 0;
	}
	输入: 1 2 3 4 5 6 0 7 输出:
2.	<pre>#include <iostream></iostream></pre>
	using namespace std;
	<pre>int main()/{</pre>

```
int i = 100, x = 0, y = 0;
      while (i > 0) {
          i--;
         x = i \% 8;
          if (x == 1)
             y++;
      cout << y << endl;</pre>
      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;</pre>
       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;</pre>
   else if (s1 > s2)
       cout << ">" << endl;</pre>
   else
       cout << "<" << endl;</pre>
   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:
```

123 -789 输出: 123 -789

#include <iostream>
using namespace std;

```
if (c == '-')
       negative = 1;
   else
       (2)
   c = cin.get();
   while (<u>(3)</u>) {
      (4)
      c = cin.get();
   if (negative == 1)
       (5);
   return num;
}
int main() {
   int a, b;
   a = readint();
   b = readint();
   cout << a << endl << b << endl;</pre>
   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)__)</pre>
```

```
count += C[j] - M[i];
       i++;
       j++;
   return (3);
}
void sort(int a[], int l, int r) {
   int i = 1, j = r, x = a[(1 + 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 (1 < j) sort(a, 1, j);
}
int main() {
   int i;
   cin >> n >> B >> A;
   for (i = 1; i \le n; i++)
       cin >> M[i];
   for (i = 1; i <= B; i++)
       cin >> C[i];
   sort(M, 1, n);
   sort(C, 1, B);
   1 = 0;
   r = n;
   while (1 <= r) {
       mid = (1 + r) / 2;
       if (<u>(4)</u>) {
           ans = mid;
         1 = mid + 1;
       } else
                 (5);
    cout << ans << endl;</pre>
   return 0;
}
```