

2022 CCF 非专业级别软件能力认证第一轮

(CSP-J)入门级 C++语言试题 模拟卷 - 1

考生注意事项:

1. 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。
2. 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 15 题, 每题 2 分, 共计 30 分。每题有且仅有一个正确答案)

1. 一个 IPv6 的地址占用多少位? ()
A. 32 B. 64 C. 128 D. 256
2. 请你计算 $(312)_8 + (1001011)_2 =$ ()
A. $(275)_{10}$ B. $(115)_{16}$ C. $(424)_8$ D. $(100010111)_2$
3. 下列哪个是邮件传输协议 ()。
A. HTTP B. TCP C. POP D. FTP
4. 使用冒泡排序时对序列 5, 4, 3, 2, 1 进行升序排序, 需要执行 () 次交换操作。
A. 0 B. 5 C. 10 D. 15
5. 下列哪个不是计算机硬件系统的必要组成部分? ()
A. CPU B. 主板 C. 操作系统 D. 内存
6. 下列哪个是数组具有的特点 ()
A. 可随机访问任一元素 B. 插入不需要移动元素
C. 不必事先估计存储空间 D. 删除不需要移动元素
7. 如果一个栈的入栈序列是 5642137, 则以下哪个不可能是出栈序列? ()
A. 4173265 B. 6513427 C. 5642137 D. 2146735
8. 在一个袋子里装有 3 个不同的红球, 3 个不同的蓝球, 3 个不同的黄球。你从中一次取出 2 个球, 抽到相同颜色的球的方案数是 ()?
A. 9 B. 3 C. 6 D. 36
9. 班里有 7 位男同学和 5 位女同学。现在班里进行抽奖活动, 总共有 4 份相同的奖品。为了保证活动的公平, 班主任准备了 12 个小纸条, 小纸条中的 4 个写着中奖。现在每位同学都发到了一个小纸条, 问最后恰好 2 男 2 女中奖的方案有多少个? ()
A. 210 B. 420 C. 72 D. 64
10. 312 和 234 的最小公倍数是 ()。
A. 936 B. 1872 C. 2808 D. 3744
11. 区间 $[1, 10000]$ 中包含连续的 62 的数字共有多少个? ()
A. 209 B. 210 C. 299 D. 310

12. 假设对 128 个排好序的元素，采用折半查找时，最大比较次数是 ()
A. 6 B. 7 C. 8 D. 9
13. 如果一棵二叉搜索树的前序遍历是 ABCDEFG，中序遍历是 BAEFDGC，则它的后序遍历是 ()
A. BFEGDCA B. BEFGDCA C. BFECGDA D. BEFGCDA
14. 如果一棵二叉树有 56 个节点，则这个二叉树最少有几层？根节点算第 1 层。()
A. 4 B. 5 C. 6 D. 7
15. 一棵完全二叉树的节点总数是 1001，其叶子节点数为 ()
A. 250 个 B. 500 个 C. 254 个 D. 501 个

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断正确填√，错误填×；除特殊说明外，判断题一题 2 分，选择题一题 2.5 分，共计 40 分）

1.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int n, a[100];
4  int main()
5  {
6      cin >> n;
7      for(int i=1; i<=n; i++)
8          cin >> a[i];
9      int ans=1;
10     for(int i=1; i<=n; i++)
11     {
12         if( i>1 && a[i]<a[i-1] )
13         {
14             ans = i;
15         }
16         while( ans<n && a[i]>=a[ans+1])
17             ++ans;
18         cout << ans << " ";
19     }
20     return 0;
21 }
```

100%的数据：int 范围内。。

1. 判断题

- (1) 第19行输出ans时，ans的值一定大于i。()
- (2) 程序输出的ans小于等于n。()
- (3) 若将13行的 "< " 改为 "!= "，程序的输出结果不会改变。()
- (4) 当程序执行到第19行时，若ans-i>2，则a[i+1]<=a[i]。()

2. 选择题

- (1) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度是()
A. $O(\log n)$ B. $O(n^2)$ C. $O(n \log n)$ D. $O(n)$
- (2) 最坏情况下，此程序的时间复杂度是()
A. $O(n^2)$ B. $O(\log n)$ C. $O(n)$ D. $O(n \log n)$

2.

```
1  #include<iostream>
2  using namespace std;
3  int a[10000], b[10000], s[10000];
4  int main() {
5      int n;
6      cin >> n;
7      for (int i = 1; i <= n; i++) {
8          cin >> a[i];
9          if (b[ a[i] ] == 0) b[ a[i] ] = i;
10         s[ b[ a[i] ] ] += a[i];
11     }
12     int sum1 = 0, sum2 = 0, sum3 = 0;
13     for(int i = 0; i < 10000; i++) {
14         if (b[i] != 0) sum1++;
15         if (s[ b[i] ] != 0) sum2 += s[ b[i] ] / a[ b[i] ];
16     }
17     for(int i = 1; i <= n; i++) {
18         if (b[ a[i] ] != i) sum3++;
19     }
20     cout << sum1 << " " << sum2 << " " << sum3 << endl;
21     int maxs = 0;
22     for (int i = 0; i < 10000; i++) {
23         if(s[i] > maxs) maxs = s[i];
24     }
25     cout << maxs << endl;
26     return 0;
27 }
28
```

约定输入数据 n 和 a_i 均为区间[1,10000]的整数。

● 判断题

- 1) 若存在 $s[i]$ 为素数, 则输入的 n 个数中一定存在某个数字只出现了1次。()
- 2) 当程序执行到第21行时, $sum1+sum3$ 等于 $sum2$ 。()
- 3) 将第13行“ $i = 0$ ”改为“ $i = 1$ ”, 程序执行的结果可能发生改变。()
- 4) 第26行的 $maxs$ 的值可能大于 n 。()

● 选择题

- 5) 当 $n=5$ 时, 输出 $sum3$ 的最大值为 ()
A.3 B.5 C.4 D.2
- 6) 当 $n=5$ 时, 输出 $maxs$ 的最小值为 ()
A.3 B.4 C.5 D.6

3.

```
1  #include<iostream>
2  using namespace std;
3  int a[10000];
4  bool p[10000];
5  int fun(int x){
6      int cnt = 0;
7      for (int i = 2; i*i <= x; i++){
8          if(x%i == 0 && !p[i]) cnt++;
9          if(x%i == 0 && x/i != i && !p[x/i]) cnt++;
10     }
11     if(cnt > 0) return cnt;
12     else return 1;
13 }
14 int main() {
15     int n, sum = 0;
16     cin >> n;
17     for (int i = 2; i < n; i++) {
18         if(p[i]) continue;
19         a[i] = 1;
20         for (int j = i + i; j < n; j += i) {
21             a[j]++;
22             p[j] = true;
23         }
24     }
25     for (int i = 2; i < n; i++)
26         sum += fun(i) - a[i];
27     cout << sum << endl;
28     return 0;
29 }
```

约定输入 n 为区间 $[1, 10000]$ 的正整数。

● 判断题

- 1) 当输入n的值为10000时，p[j]为true说明j不是素数。()
- 2) 输出值sum一定为正整数。()
- 3) 将第17行“i < n”改为“i*i<=n”不会影响程序执行的结果。()
- 4) 当输入n的值为10000时，第26行fun(i)的返回值不可能为大于1的奇数。()

● 选择题(每题3分)

- 5) 当输入n为10000时，调用函数fun(2310)的值为 ()
A.5 B.10 C.11 D.9
- 6) 当n为10时，输出sum的值为 ()
A.0 B.5 C.10 D.20

三. 完善程序 (单选题, 每小题 3 份, 共计 30 份)

1.(班级成绩计算)某不知名 CCF 中学共有 3 个班级，每人班级人数相等。如果 A 班某同学的考试分数加 B 班某同学的考试分数等于 C 班某两同学的考试分数，那么他们就可以组成一队黄金小三角。现告诉每班人数 n，以及每个班学生的成绩。(其中保证 C 班同学的分数是不下降的，即 $C[i] \leq C[i+1]$)，询问共可以找出多少组合可以使得 $A[i] + B[j] = C[x] + C[y]$ ($1 \leq i, j, x, y \leq n$ 且 $x < y$)。例如输入为

```
2
1 2
2 3
1 3
```

则输出为 2。可行的解为(1,3)(1,3)，及(2,2)(1,3)

数据范围： $n \leq 3000$, $1 \leq A[i], B[i], C[i] \leq 300$;

```
1 long long A[3000], B[3000], C[3000], cnt[1000];
2 int main() {
3     int n, ans;
4     cin >> n;
5     for (int i = 0; i < n; i++) cin >> A[i];
6     for (int i = 0; i < n; i++) cin >> B[i];
7     for (int i = 0; i < n; i++) cin >> C[i];
8     for (int i = 0; i < n; i++) {
9         for (int j = 0; j < n; j++) {
10             long long sum = ____(1)__;
11             cnt[sum]++;
12         }
13     }
14     ____(2)__;
15     for (int i = 0; i < n; i++) {
16         for (int j = ____(3)__; j < ____(4)__; j++) {
17             int sum = C[i] + C[j];
18             ans = ____(5)__;
19         }
20     }
21     cout << ans << endl;
```

	}

- 1) (1)处应填()
A. A[i]+A[j] B. B[i]+B[j] C. A[i]+B[j] D. B[i]+C[j]
- 2) (2)处应填()
A. ans=-1 B. ans=0 C. cnt[0]=0 D. cnt[0]=-1
- 3) (3)处应填()
A. 0 B. 1 C. i D. i+1
- 4) (4)处应填()
A. i+1 B. n-i C. n-i+1 D. n
- 5) (5)处应填()
A. cnt[sum] B. cnt[-sum] C. ans+cnt[sum] D. ans+cnt[-sum]

2. (循环比赛日程表) 现有 m 个选手进行循环比赛, 其中 $m=2^k$ 需要设计一个比赛日程表。每名选手要与其他 $m-1$ 名选手都赛一次, 每名选手每天比赛一次, 循环赛共进行 $m-1$ 天, 按此要求设计一张表, 该表有 n 行 (行号是 $1 \sim n$), n 列 (列号是 $0 \sim n-1$), 第 i 行 ($i \geq 1$) 第 j 列 ($j \geq 1$), 表示第 i 个人第 j 天遇到的选手, 如右图所示 ($k=3$)

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

```

1  #include<iostream>
2  using namespace std;
3  int m, table[1025][1025];
4  void solve(int row, int column, int n){
5      if( n==1 ) return ;
6      int half = __ (1) __;
7      table[row+half][__ (2) __] = table[row][column];
8      table[row][column+half]=table[__ (3) __][column]=table[row][column]+half;
9      solve(row, column, half);
10     solve(row, column+half, half);
11     solve( __ (4) __, half);
12     solve(row+half, column+half, half);
13 }
14 int main(){
15     cin >> m;
16     table[1][1] = 1;
17     solve(1,1,m);
18     for(int i=1; i<=m; i++){
19         for(int j=1; j<=m; j++){
20             cout << __ (5) __ << "\t";
21             cout << endl << endl << endl;

```

22	}
23	}

- 1) (1)处应填()
A. $m/2$ B. $n/2$ C. $m/4$ D. $n/4$
- 2) (2)处应填()
A. row B. column C. column+half D. row+half*2
- 3) (3)处应填()
A. row+half B. row C. column D. column+half
- 4) (4)处应填()
A. half, row B. row, column
C. row+half, column+half D. row+half, column
- 5) (5)处应填()
A. table[i][j] B. table[i][j+1]
C. table[i+1][j+1] D. table[j][i]