

【睿爸信奥】CCF 非专业级别软件能力认证第一轮 (CSP-J) 入门级 C++语言模拟题

考生注意事项:

- 试题纸共有 9 页, 答题纸共有 1 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 15 题, 每题 2 分, 共计 30 分; 每题有且仅有一个正确选项)

1. 下列属于面向对象的程序设计语言是 ()。
A. Basic B. Pascal C. C D. C++
2. 与十进制数 28.5625 相等的四进制数是 ()。
A. 123.21 B. 131.22 C. 130.21 D. 130.20
3. $a[1]=1, a[2]=3, a[3]=2, a[4]=4$ 那么表达式 $a[a[3]+a[a[1]]]$ 的值为 ()
A. 1 B. 4 C. 2 D. 3
4. 一次数学考试试题由两部分组成, 结果全班有 15 人得满分, 第一部分做对的有 31 人, 第二部分做错的有 19 人, 那么两部分都做错的有 ()。
A. 3 B. 6 C. 4 D. 12
5. 今天是中秋节, 徐老师给各位同学发月饼, 假设做了 400 个月饼 (期待每位同学都能 AK 复赛), 要把它们装到 18 个盒子里面, 那么数量最多的一箱至少装 () 个月饼?
A. 16 B. 17 C. 23 D. 15
6. 已知字符 'a' 的 ASCII 码是 97, 写出下面程序的输出结果: ()

```
1 char c='a'+4;  
2 cout<<c<<" "<<(int)c+3<<endl;
```

- A. e, h B. e, 104 C. 104, e D. 101, h

7. 写出下面程序的输出结果: ()

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int x=0x35,y=027,z;
5     x^=y, y^=x, x^=y;
6     cout<<x<<" "<<y<<" ";
7     z=(x&y)+((x^y)>>1);
8     cout<<z<<endl;
9     return 0;
10 }

```

A. 23 53 38 B. 35 27 32 C. 53 23 38 D. 27 53 39

8. 设一组初始记录关键字序列为{50, 40, 95, 20, 15, 70, 60, 45}, 进行排序, 每一次操作将未排序部分最小值和该部分的首元素对调, 四轮交换后, 前 4 个元素的值为 ()

- A. 15, 20, 50, 40 B. 15, 40, 60, 20
C. 15, 20, 40, 50 D. 15, 20, 40, 45

9. 具有 n 个顶点, e 条边的图采用邻接表存储, 进行深度优先遍历和广度优先遍历运算的时间复杂度分别为 ()

- A. $O(n+e)$ 和 $O(n+e)$ B. $O(e^2)$ 和 $O(n+e)$
C. $O(ne)$ 和 $O(ne)$ D. $O(ne)$ 和 $O(n+e)$

10. 一棵二叉树前序遍历为 ABDECFGH, 后序遍历为 EDBGFHCA, 以下不可能的中序遍历的是 ()

- A. DEBAFGCH B. BEDAFGCH C. EDBAGFCH D. DBEAFGCH

11. 给出以下邻接矩阵, 其表示的图是 DAG (有向无环图) 的是 ()。

- A. $\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ B. $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ C. $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ D. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

12. 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. $2*N-1$ D. N

13. 2024 年徐老师的信奥班进复赛的同学有 6 人, 徐老师让他们站成一圈分发讲评, 请问有多少种排法 ()。

- A. 240 B. 120 C. 60 D. 180

14. 字符串 abcab 不同的连续子串 (包含空串) 个数 ()。

- A. 13 B. 14 C. 15 D. 16

15. 给定如下定义的一个循环单向链表结构，假设你现在有一个指向链表中某个节点 p 的指针，链表中的节点结构和操作如下图，当从节点 p 开始，沿着链表的 $next$ 指针遍历时，什么时候将再次回到节点 p ()

```

1 struct Node {
2     int data;
3     struct Node* next;
4 };
5 struct Node* p = ... // p 指向某个节点
6 struct Node* current = p;
7 do {
8     printf("%d ", current->data);
9     current = current->next;
10 } while (current != p);
11

```

- A. 当遍历了链表中所有节点的数量时
- B. 当遍历了链表中的节点数加 1 的时候
- C. 当遍历到链表的尾节点时
- D. 当链表中的下一个节点为 NULL 时

二、阅读程序（程序输入不超过数组或字符串定义的范围；注意：判断题正确填 T，错误填 F；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

阅读下面程序，完成第 16~20 题。

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 unsigned int num;
4 bool bit[32];
5 int main() {
6     cin>>num;
7     for (int i=31; i>=0&&num; i--) {
8         bit[i]=num%2;
9         num/=2;
10    }
11    for (int i=0; i<16; i++) swap(bit[i], bit[16+i]);
12    for (int i=0; i<32; i++)
13        if (bit[i]) num += 1<<(31-i);
14    cout<<num;
15    return 0;
16 }
17

```

16. 该程序能够将一个无符号整数的二进制位进行前后交换 ()。

17. 如果输入的整数是 $2^{32}-1$ (提示: $0xFFFFFFFF$, 即 32 位全为 1), 则程序输出应为 $0x00000000$ 。 ()
18. 程序中的 swap 操作可以替换为位操作来提高效率。 ()
19. 程序中的 bit 数组的作用是 ():
- A. 存储整数的二进制表示
 - B. 存储整数的十六进制表示
 - C. 存储整数的八进制表示
 - D. 存储整数的十进制表示
20. 代码中的 `swap(bit[i], bit[16+i]);` 这一行的作用是什么?
- A. 将 bit 数组的前 16 位和后 16 位进行交换
 - B. 将 bit 数组的奇数位和偶数位进行交换
 - C. 将 bit 数组的第 1 位和第 32 位进行交换
 - D. 将 bit 数组的第 16 位和第 17 位进行交换

阅读下面程序, 完成第 21~26 题。

```
1 # include <bits/stdc++.h>
2 using namespace std;
3 int L,ans;
4 char a[2002][2002];
5 int cross(int x,int y){
6     int length=1;
7     if(x<=1 || x>=L) return 1;
8     for(int i=1; ;i++){
9         if(x-i<=0 || x+i>=L+1) return length;
10        else if(a[x-i][y]!=a[x+i][y]) return length;
11        else length+=2;
12    }
13 }
14
15 int down(int x, int y) {
16     int length=1;
17     if(y<=1 || y>=L) return 1;
18     for (int i=1; ;i++) {
19         if (y-i<=0 || y+i>=L+1) return length;
20         else if(a[x][y-i]!=a[x][y+i]) return length;
21         else length+=2;
22     }
23 }
24 int MAXN ( int a, int b) {
25     if(a>=b) return a;
26     else return b;
27 }
28
29 int main() {
30     cin >> L;
31     for (int i=1; i<=L;i++)
32         for (int j=1; j<=L;j++) cin>>a[i][j];
33     int x, y;
34     cin>>x>>y;
35     ans=MAXN(cross(x,y),down(x,y));
36     cout<<ans;
37     return 0;
38 }
39
```

21. 第 35 行 `ans=MAXN(cross(x,y),down(x,y))`;若改成 `MAXN(down(x,y),cross(x,y))`, 运行结果不变。()

22. 第 34 行 `cin>>x>>y`;输入值包含 0 时,程序可能会产生 Runtime Error()

23. 程序输出的 ans 可能等于 0。 ()
24. 当第 34 行输入值 $x > y$ 时, $\text{cross}(x, y)$ 返回值**必然**大于 $\text{down}(x, y)$ 返回值。
()
25. 对于输入的 $L \times L$ 的字符矩阵, ans 值最大是()。
A. $2L$ B. L C. $L \times L$ D. $L-1$
26. 输入以下值, 则输出是()
5
abcba
bcdcb
cdedc
bcdcb
abcba
3 3
A. 3 B. 5 C. 10 D. 25

阅读下面程序, 完成第 27~32 题。

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int N = 1000003;
4  int m,d[N];
5  int gcd(int a, int b){
6      if (b == 0) return a;
7      return gcd(b, a % b);
8  }
9  int main(){
10     cin>>m;
11     while(m--){
12         int A,B;
13         cin>>A>>B;
14         int n=A*B/gcd(A,B);
15         int cnt=0;
16         for (int i=2;i*i<=n;i++){
17             while (n%i==0){
18                 d[++cnt]=i;
19                 n/=i;
20             }
21         }
22         if(n!=1) d[++cnt]=n;
23         for (int i=1; i<=cnt; i++)
24             cout<<d[i]<<endl;
25     }
26     return 0;
27 }

```

27. 上述代码是在对 A 和 B 的最小公倍数做质因数分解。 ()

28. 将程序会输出 n 行。 ()

29. d 数组中的元素按照不下降顺序排列。 ()

30. 若 $A=1 \times 10^7$, $B=2 \times 10^6$, 程序能正常执行 ()。

31. 若输入为 1 12 10, 则输出为 ()

- | | | | |
|------|------|------|------|
| A. 2 | B. 2 | C. 2 | D. 2 |
| 3 | 2 | 3 | 2 |
| 5 | 3 | 5 | 15 |
| | 5 | 2 | |

32. 如果删除第 22 行的 if(n!=1) 则 ()。

- A. 程序的正确性一定会改变

- B. d 数组中的元素一定不会按照不下降顺序排列
- C. 如果 AB 不相同, cnt 一定会改变
- D. 对于所有的 $d[i], d[i] \leq \lfloor \sqrt{n} \rfloor$ 都成立

三、完善程序（单选题，每小题 3 分，共计 30 分）

阅读下面题目，完成第 33~42 题。

(1) (区间元素个数) 给定一个数组中的 N 个元素，统计位于给定区间 [L, R] 的元素个数。

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  const int N=10010;
5  int getRight(int arr[], int left, int right, int target) {
6      if (left >= right) return left;
7      int mid = (right+left)/2;
8      if (arr[mid] <= target) {
9          return getRight(arr, _____ 1 _____, target);
10     } else {
11         return getRight(arr, _____ 2 _____, target);
12     }
13 }
14
15 int main() {
16     int arr[N], n, m;
17     cin >> n;
18     for (int i = 0; i < n; i++) cin >> arr[i];
19     sort(arr, arr + n);
20     cin >> m;
21     while(m--){
22         int L, R;
23         cin >> L >> R;
24         int left = _____ 3 _____;
25         int right = _____ 4 _____;
26         int count = _____ 5 _____;
27         cout << count << endl;
28     }
29     return 0;
30 }
31

```

33. (1) 处应填 ()
 A. mid, right+1 B. mid, right C. mid+1, right D. mid, right-1
34. (2) 处应填 ()
 A. left+1, mid B. left, mid C. left, mid+1 D. left, mid-1

35. (3) 处应填 ()
- A. `lower_bound(arr, arr+n, L)` B. `lower_bound(arr, arr+n, L)-arr`
C. `lower_bound(0, n, L)-arr` D. `lower_bound(arr, arr, R)-arr`
36. (4) 处应填 ()
- A. `getRight(arr, 1, n+1, R)` B. `getRight(arr[], 0, n, R)`
C. `getRight(0, arr, n, R)` D. `getRight(arr, 0, n, R)`
37. (5) 处应填 ()
- A. `right-left` B. `right-left+1`
C. `right-left-1` D. `left+(right-left)/2`

(2) (子串查找) 给定一个字符串 S , 有 q 组询问, 每次给定一个字符串 T , 求字符串 T 是否是 S 中的一个子序列。其中 $1 \leq S \leq T \leq 1e^6$, 所有字符串仅包含小写字母。提示: `Pos` 数组的主要作用是为了快速确定在字符串 S 中从任何一个给定位置开始, 各个字母 (从 'a' 到 'z') 下一次出现的位置。这使得能够迅速判断一个字符串 T 是否是 S 的子序列。

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  const int ALPHABET_SIZE = 26;
5
6  string S, T;
7  vector<vector<int> > Pos;
8  int main() {
9      cin >> S;
10     int n = ____ 1 ____;
11     Pos.resize(n + 1, vector<int>( ____ 2 ____ ));
12     for (int i = n - 1; i >= 0; --i) {
13         ____ 3 ____;
14         Pos[i][S[i]-'a'] = i;
15     }
16     int q;
17     cin >> q;
18     while (q--) {
19         cin >> T;
20         int len=T.size(), now=0;
21         for (int i=0; i<len && now!=-1; ++i) {
22             now = ____ 4 ____;
23         }
24         cout << ( ____ 5 ____ ? "YES\n" : "NO\n");
25     }
26     return 0;
27 }

```

38. (1) 处应填 ()
 A. S.size() B. S.size()-1 C. ALPHABET_SIZE-1 D. ALPHABET_SIZE
39. (2) 处应填 ()
 A. ALPHABET_SIZE, 0 B. ALPHABET_SIZE, -1
 C. S.size(), -1 D. S.size(), 0
40. (3) 处应填 ()
 A. Pos[i] = Pos[i+1]; B. Pos[i] = Pos[i-1];
 C. Pos[i+1] = Pos[i]; D. Pos[i-1] = Pos[i];
41. (4) 处应填 ()
 A. Pos[now][T[i-1]-'a'] B. Pos[now][T[i]]
 C. Pos[now][T[i]-'a'] D. Pos[now-1][T[i]-'a']



42. (5) 处应填 ())
A. now!=0 B. now==-1 C. now==0 D. now!=-1

睿爸信奥预祝各位考生在 2024 年 CSP 竞赛中取得好成绩!



微信公众号



QQ 交流群

复赛集训刷题 OJ 地址: <https://www.csp-j.com/>