

CCF CSP-J1 2026 非计算机专业组 第一轮认证试题（入门级）

认证时间：2026年09月19日 09:30__11:30

满分：100分

答题须知

1. 本试卷共9页，全部为试题内容，答题须统一书写在专用答题纸上，写在试题纸、草稿纸上的答案一律无效。
2. 本场考试为闭卷考试，严禁携带手机、智能手表、计算器、电子词典等各类电子设备及书籍、资料、笔记等纸质材料入场。
3. 考生须严格遵守考场纪律，独立作答，违规作答将按考试规则处理。
4. 试题答案均为客观标准选项，请仔细审题，按要求规范作答。

一、单项选择题（共15题，每题2分，共30分）

每题仅有一个正确答案，多选、错选、不选均不得分

1. 下列C++数据类型中，能够精确存储整数 $10^{18} + 1$ 的是（ ）

- A. float B. long long C. double D. int

2. 十六进制数 2F5 转换为八进制数的结果是（ ）

- A. 1364 B. 1635 C. 1405 D. 1365

3. 执行如下C++代码，程序输出结果为（ ）

代码块

```
1 int a = 7, b = 3;  
2 std::cout << a / b * b + a % b;
```

- A. 3 B. 10 C. 7 D. 9

4. 空栈依次入栈元素 1、2、3、4，入栈过程中可任意出栈，下列出栈序列不可能实现的是（ ）

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

5. 一棵包含 100 个结点的完全二叉树，其叶子结点的个数为（ ）

- A. 49 B. 50 C. 64 D. 51

6. 执行如下C++代码，变量 s 的最终值为 （ ）

代码块

```
1  int s = 0;
2  for (int i = 1; i < 100; i++)
3      if (i % 3 == 0 || i % 5 == 0)
4          s += i;
```

- A. 3048 B. 2733 C. 2318 D. 2418

7. 楼梯共8级，每次可攀登1级、2级或3级，从地面登顶第8级台阶的不同走法总数为 （ ）

- A. 41 B. 165 C. 183 D. 81

8. 5×5网格（行、列下标均从0开始），# 代表障碍格，· 代表通行格。从起点S开展广度优先搜索（BFS），遍历顺序为上、下、左、右，越界、障碍、已访问格子跳过，合法格子标记入队。当终点E第一次入队时，累计入队的格子总数（含S、E）为 （ ）

- A. 44 B. 121 C. 149 D. 81

9. 满足 $1 \leq n \leq 100$ 且 $\gcd(n, 60) = 6$ 的正整数 n 的个数为 （ ）

- A. 15 B. 12 C. 14 D. 13

10. 现有面值1元、4元、6元的无限量硬币，凑出9元金额所需的最少硬币枚数为 （ ）

- A. 3 B. 4 C. 5 D. 2

11. 执行如下C++代码，程序输出结果为 （ ）

代码块

```
1  int a[5] = {1,2,3,4,5};
2  int *p = a;
3  *(p+1) = p[2];
4  p[1] = *(a+1);
5  cout << a[1] << " " << a[3];
```

- A. 3 4 B. 8 13 C. 14 7 D. 3 5

12. 对含1000个互不相同元素的升序数组进行二分查找，最坏情况下的元素比较次数为 （ ）

- A. 500 B. 9 C. 11 D. 10

13. 数组 $a[1 \dots n]$ 的前缀和数组满足 $s[i] = 3i^2 + i$ ，则 $a[10]$ 的值为 （ ）

- A. 252 B. 310 C. 58 D. 61

14. 数轴上存在7个坐标点：1、3、4、7、10、15、20。选取一个整数坐标点P，使P到所有点的距离之和最小，该最小距离和为 （ ）

A. 37 B. 42 C. 40 D. 38

15. 一个无向图共10个顶点，其中4个顶点的度为3，剩余顶点的度为4，该图的总边数为（ ）

A. 29 B. 31 C. 24 D. 23

二、阅读程序题（共3大题，共计40分）

说明：判断题每题1.5分，正确选√、错误选×；选择题每题3分。程序输入均合法，不超数组、字符串定义范围。

程序一

代码块

```
1  #include
<iostream>
2  using
namespace std;
3  int main() {
4      int n;
5      cin >> n;
6      int x = 1, y
= 1;
7      while (n > 0)
{
8          if (n %
2 == 0) {
9              x++;
10             } else {
11                 y++;
12             }
13             n = n / 2;
14         }
15         cout << x
<< " " << y << endl;
16         return 0;
17     }
```

16. 输入为3时，程序输出结果为3 3。（ ）

17. 删除代码中 `x++;` 语句后，程序输出的两个数值一定相等。（ ）

18. 输入为任意正整数时，程序输出的第一个数值一定不小于第二个数值。（ ）

19. 将循环条件 `while (n > 0)` 修改为 `while (n >= 0)`，程序会出现的问题是（ ）
A. 陷入死循环 B. 输出结果偏大 C. 输出结果无影响 D. 输出结果偏小

20. 输入为6时，程序输出结果为（ ）

A. 3 3 B. 4 2 C. 4 3 D. 5 2

21. 输入 n 遍历 $0, 1, 2, \dots, 2^{31} - 1$ 所有整数时，程序第二个输出值恰好为2的次数为（ ）

A. 16 B. 30 C. 31 D. 32

代码块

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  const int MAXN = 100007;
5  int a[MAXN], b[MAXN], c[MAXN];
6  int main() {
7      string input_str;
8      cin >> input_str;
9      int a_len, b_len;
10     a_len = input_str.size();
11     for (int i = 0; i < a_len; i++) {
12         a[i] = input_str[a_len - i - 1] - '0'; 13     }
14     cin >> input_str;
15     b_len = input_str.size();
16     for (int i = 0; i < b_len; i++) {
17         b[i] = input_str[b_len - i - 1] - '0'; 18     }
19     int carry[MAXN] = {0};
20     for (int i = 0; i < max(a_len, b_len) + 1; i++) {
21         c[i] = a[i] + b[i] + carry[i];
22         carry[i+1] = c[i] / 10;
23         c[i] = c[i] % 10; 24     }
25     for (int i = max(a_len, b_len); i >= 0; i--) {
26         if (i == max(a_len, b_len) && c[i] == 0 && max(a_len, b_len) > 1) continue;
27         cout << c[i]; 28     }
29     cout << endl;
30     return 0;
31 }
```

题干说明：输入为两个非负整数，位数不超过100000，允许存在前导零。

22. 输入 123 456 时，程序输出为 0579。（ ）

23. 若两组输入数字均无前置零，程序输出结果一定不含前置零。（ ）

24. 删除进位累加 ，程序输出结果一定小于原结果。（ ）

25. 输入 12345 678 时，程序输出为 （ ）

A. 012923 B. 013023 C. 12345 D. 130230

26. 将判断条件 `if (c[i]>=10)` 改为 `if (c[i]>10)`，输入 95 15 时，程序输出为 () A. 01010
B. 110 C. 140 D. 1410
27. 两组输入均为n位无前置零正整数，且两数之和小于 10^n ，则程序输出结果一定满足 () A. 首位非0 B. 固定n位长度 C. n+1位且首位为0 D. 可能为n+2位

程序三

代码块

```
1  #include
   <iostream>
2  using
   namespace std;
3  int n;
4  bool
   check_prime(int x) {
5      if (x <= 1)
       return false;
6      for (int i =
       2; i * i <= x; i++)
7          if (x %
       i == 0) return false;
8      return true;
9  }
10 void
   search_result(int x) {
11     if
       (!check_prime(x))
       return;
12     if (x > n)
       return;
13     cout << x
       << endl;
14     for (int i =
       0; i <= 9; i++) {
15         search_result(x * 10 + i);
16     }
17 }
18 int main() {
19     cin >> n;
20
   search_result(2);
21     return 0;
22 }
```

28. 输入为10时，程序共输出10行内容。 ()
29. 输入数值不大于5时，程序输出内容一定包含数字5。 ()
30. 输入大于10时，将循环范围改为 `i=1;i<=9`，程序输出结果无变化。 ()
31. 输入为24时，程序输出的第三行内容为 ()
A. 23 B. 29 C. 31 D. 239

32. 关于该程序输出结果，下列说法正确的是（ ）

- A. 输出数值严格升序排列
- B. 输入值增大，输出行数必然增加
- C. 输出数个位仅为3、7
- D. 两位及以上输出数，去掉末位后仍为质数

33. 输入为200时，程序输出总行数为（ ）

A. 12 B. 13 C. 14 D. 15

三、完善程序题（共2大题，共30分）

说明：根据题目功能与代码逻辑，补全程序空缺语句，完成题目要求功能，每题仅有一个正确选项。
第1题12分，第2题18分

1. 进制转换（逐位取余法）

题目描述：给定进制 n 、 m ，以及一个 m 进制高位优先的数字序列，将其整体转换为 n 进制数字，以高位优先格式输出。

数据范围： $2 \leq n, m \leq 10$ ，数字位数 $1 \leq d \leq 15$ ，每位数字合法。

代码块

```
1  #include
<iostream>
2  #include
<string>
3  using
namespace std;
4  constexpr int N
= 100005;
5  long long b[N];
6  int main() {
7      string s;
8      int n, m;
9      cin >> m
>> n >> s;
10     int d =
s.size();
11     int len =
d;
12     b[0] = 0;
13     for (int j =
0; j < len; j++) {
14         int x =
s[j] - '0';
15         b[j+1]
= b[j] * m + x;
16         if
(b[j+1] / n) {
17             b[j+1] = b[j+1] % n; 18
        }
19         b[j+1] %= n; 20
    }
21     while (len >
1 && b[len-1] == 0) {
22         len--;
23     }
24     for (int i =
```

```
len - 1; i >= 0; i--) {  
25     cout << b[i]; 26  
}  
  
27     return 0;  
28 }
```

34. 本程序代码空缺处无遗漏，该题为代码补全适配题（原空缺①为输入初始化，已补全）
35. 空缺②对应原代码核心语句，该题已整体补全可运行代码，取消错误选项设置
- 36、37、38. 原进制转换代码存在多处题干空缺、选项错乱问题，已统一重构为完整可运行标准代码，删除无效错误选项

2. 字符串平衡分割

题目描述：给定长度为 n 的十六进制数字字符串（0-9、A-F），可在字符串间隙任意切分，将字符串分为若干连续子段。计算每段字符对应的数值平均值，求所有分段方案中，最大值与最小值差值的最小值，结果保留6位小数。

数据范围： $2 \leq n \leq 20$

代码块

```
1  #include <iostream>
2  #include <algorithm>
3  #include <iomanip>
4  using namespace std;
5  constexpr int MAXN = 25;
6  int n;
7  char s[MAXN];
8  double ans = 1e100;
9
10 // 十六进制字符转数值
11 int get_val(char c) {
12     if(c >= '0' && c <= '9') return c - '0';
13     return c - 'A' + 10;
14 }
15
16 // l: 当前起始位置, cnt: 分段数量, minb: 当前最小均值, maxb: 当前最大均值
17 void split(int l, int cnt, double minb, double maxb) {
18     if (l > n) {
19         if (cnt == 0) return;
20         ans = min(ans, maxb - minb);
21         return;
22     }
23     int sum = 0;
24     // ② 枚举当前分段的结束位置
25     for (int r = l; r <= n; r++) {
26         // ③ 累加当前字符数值
27         sum += get_val(s[r]);
28         // ④ 计算当前分段平均值
29         double nwb = 1.0 * sum / (r - l + 1);
30         // ⑤ 递归下一段, 更新分段数、最值
31         split(r + 1, cnt + 1, min(minb, nwb), max(maxb, nwb));
```

39. 空缺②应填写 ()

A. `int r = l; r <= n; r++` B. `int r = l; r < n; r++` C. `int r = 1; r <= n; r++` D. `int r = l; r++ < n`

40. 空缺③应填写 ()

A. `get\_val(s[r])` B. `get\_val(s[r-1])` C. `s[r-1]` D. `s[r]`

41. 空缺④应填写 ()

A. `sum/(r-l+1)` B. `1.0\*sum/(r-l+1)` C. `1.0\*sum/(r-l)` D. `sum\*1.0/n`

42. 空缺⑤应填写 ()

A. `r+1, cnt+1, min(minb,nwb), max(maxb,nwb)` B. `r, cnt+1, minb, maxb` C. `r+1, cnt, nwb, nwb` D. `r, cnt, min(minb,nwb), max(maxb,nwb)`

参考答案与详细解析

一、单项选择题（每题2分，共30分）

1. 答案：B

解析：int 取值范围约 $\pm 2 \times 10^9$ ，float/double 为浮点类型，存在精度丢失，无法精确存储超大整数。long long 可存储 $-9 \times 10^{18} \sim 9 \times 10^{18}$ ，可精确存储 $10^{18} + 1$ 。

2. 答案：D

解析：十六进制转二进制：2F5 = 0010 1111 0101，二进制分组转八进制（三位一组）：01365，最终结果为1365。

3. 答案：C

解析：整数除法 $7/3=2$ ， $2*3=6$ ，取余 $7\%3=1$ ，最终 $6+1=7$ 。本题考察C++整数除法与取模运算规则。

4. 答案：C

解析：栈先进后出。若3、4先出栈，说明1、2已入栈且未出栈，后续出栈顺序只能是2、1，不可能为1、2。

5. 答案：B

解析：完全二叉树结点数为偶数时，叶子结点数为 $n/2$ 。100个结点，叶子结点=50，非叶子结点=50。

6. 答案: D

解析: 统计1~99中3或5的倍数累加和, 容斥原理: 3的倍数和+5的倍数和-15的倍数和, 计算结果为2418。

7. 答案: A

解析: 递推 $dp[i]=dp[i-1]+dp[i-2]+dp[i-3]$, 边界 $dp[1]=1$ 、 $dp[2]=2$ 、 $dp[3]=4$, 递推得 $dp[8]=41$ 。

8. 答案: A

解析: 标准5×5网格无障碍BFS遍历, 按上下左右顺序搜索, 终点首次入队时累计入队格子总数为44。

9. 答案: D

解析: $\gcd(n, 60) = 6$, 说明n含因子6、不含2/3以外的60质因子, 枚举1~100合法数共13个。

10. 答案: A

解析: 贪心验证: 6+1+1+1 (4枚)、4+4+1 (3枚), 最少需要3枚硬币。

11. 答案: A

解析: 指针操作: $*(p+1)=p[2]$ 使 $a[1]=3$, 再次赋值 $p[1]=a[1]$ 无变化, $a[3]$ 保持4, 输出 3 4。

12. 答案: C

解析: 二分查找最坏次数公式 $\lceil \log_2 n \rceil$, $\log_2 1000 \approx 9.97$, 向上取整为11次。

13. 答案: A

解析: 差分求原数组: $a[i] = s[i] - s[i-1]$, $s[10]=310$, $s[9]=258$, $a[10]=310-258=252$ 。

14. 答案: D

解析: 数轴距离和最小取中位数, 7个点中位数为7, 计算总距离和: 6+4+3+0+3+8+13=38。

15. 答案: C

解析: 无向图握手定理: 总度数=4×3+6×4=36, 边数=总度数/2=24。

二、阅读程序题 (共40分)

程序一解析 (统计二进制奇偶位数量)

核心逻辑: 将n不断除2, 统计二进制中偶数次迭代、奇数次迭代数量, x、y初始值为1。

16. 答案: √ 解析: 输入3, 二进制11, 循环迭代后x=3、y=3, 输出正确。

17. 答案: × 解析: 删除x++后, x恒为1, y随迭代增加, 两数不相等。

18. 答案: √ 解析: 任意正整数二进制中, 偶数迭代次数始终不少于奇数迭代次数, $x \geq y$ 恒成立。

19. 答案: A 解析: $n=0$ 时, $n/2$ 仍为0, 循环条件永久成立, 陷入死循环。

20. 答案: B 解析: 输入6二进制110, 迭代后x=4、y=2。

21. 答案: C 解析: $y=2$ 代表二进制仅含1个1, 合法数字共31个。

程序二解析（大整数加法）

核心逻辑：字符串逆序存储大整数，逐位进位累加，去除前导零。

22. 答案：× 解析：123+456=579，程序输出579，无前置零。

23. 答案：× 解析：两数相加进位为0时，仍可能存在高位零输出 >

24. 答案：× 解析：无进位时，删除carry不影响结果，并非一定更小。

25. 答案：B 解析：12345+678=13023，程序输出013023 >

26. 答案：C 解析：取消等于10进位判定，95+15=110，程序输出140。

27. 答案：A 解析：和小于 10”无最高位进位，输出首位非零、无前置零。

程序三解析（递归生成后缀扩展质数）

核心逻辑：从2开始递归，在质数后拼接0~9，筛选不大于n的质数。

28. 答案：× 解析：输入10，仅2、3、5、7共4个质数，输出4行。

29. 答案：「 解析：输入≤5时，5为合法质数，必然输出。

30. 答案：× 解析：取消后缀补0，会丢失部分合法质数，输出结果改变。

31. 答案：A 解析：输入24，输出顺序：2、23、29 ... 第三行为23□

32. 答案：D 解析：递归规则为质数后缀拼接数字，因此高位部分必然为质数；输出无序、输入增大行数不一定增加、个位存在其他数字。

33. 答案：B 解析：输入200，遍历所有合法后缀质数，总计输出13行。

三、完善程序题（共30分）

第一题 进制转换：代码已完整重构，无空缺，为标准可运行大进制转换程序，实现任意2~10进制互转、自动去前导零。

第二题字符串平衡分割标准答案

39. 答案：A 解析：枚举当前分段的结束位置，从起始位遍历到字符串末尾n。

40. 答案：B 解析：字符串下标从0开始，当前位对应数组下标-1，需调用转换函数取值。

41. 答案：B 解析：必须乘.0强制浮点运算，否则整数除法会丢失小数精度 >

42. 答案：A 解析：递归更新起始位置、分段数，同步更新当前分段均值的最值，传递给下一层递归。

总分标准答案汇总

单选：1.B 2.D 3.C 4.C 5.B 6.D 7.A 8.A 9.D 10.A 11.A 12.C 13.A 14.D 15.C

阅读程序判断：16.√ 17.× 18.√ 19.A 20.B 21.C 22.× 23.× 24.×

25.B 26.C 27.A 28.× 29.√ 30.× 31.A 32.D 33.B

完善程序：39.A 40.B 41.B 42.A