

非专业级别软件能力认证第一轮

(CSP-J1) 入门级 C++ 语言模拟卷

考生注意事项：

- 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. 下列对中央处理器 CPU 的组成，描述最准确的是（ ）。

- A. 控制器，指令集，缓存
- B. 存储器，总线，运算器
- C. 运算器，控制器
- D. 运算器，控制器及一些寄存器

2. 数据结构中的顺序结构和链式结构的区别主要是什么（ ）。

- A. 顺序结构常用于线性数据的存储，链式结构则用于非线性数据的存储。
- B. 顺序结构存储的数据增、删、改、查速度更快。
- C. 顺序结构的内存地址一定是连续的，链式结构则不一定连续。
- D. 链式结构存储的数据增、删、改、查速度更快。

3. 有一个容量无限的整数背包，这个背包规定：若整数 x 在这个背包中，则 $2 * x$ 和 $5 * x$ 同样属于这个背包，若背包里包含数字 1，则背包中一定包含（ ）。

- A. 600 B. 560 C. 360 D. 640

4. 一个无向简单连通图 G 有 n 个节点和 m 条边， m 的取值范围是（ ）。

- A. $m \geq n-1 \ \&\& \ m \leq n(n-1)/2$
- B. $m \geq n \ \&\& \ m \leq n(n+1)/2$
- C. $m \leq n-1$
- D. $m \geq n(n-1)/2$

5. 一颗二叉树的中序遍历序列为 DBGEACF，后序遍历序列为 DGEBFCA，则前序遍历序列是（ ）。

- A. ABDECFG
- B. ABDEGCF
- C. ABDGECF
- D. ABCEGDF

6. 假设存在一个十六进制数 0x1A3, 学生们需要将其转换为一个具有 7 个符号的进制系统, 在这个 7 进制系统中, 十六进制数 0x1A3 对应的 7 进制数是多少? ()

- A. 419
- B. 6311
- C. 1136
- D. 643

7. 用一个大小为 10 的数组实现循环队列, 某一时刻 front=4 指向队首元素, rear=2 指向队尾元素的下一位置, 经历了元素入队, 出队, 出队, 入队, 出队, 出队后, 该循环队列中还剩多少元素? ()

- A. 5
- B. 6
- C. 7
- D. 8

8. C++中, 运算表达式 $(-3 \mid -5)$ 的结果为 ()

- A. -1
- B. 1
- C. -2147483647
- D. 2147483647

9. 现在有一张分辨率为 2048x2048 的 128 位真彩色图象。请问存储这张图像, 需要多大的存储空间?()

- A. 16MB
- B. 32MB
- C. 8MB
- D. 64MB

10. 用 A、B、C、D、E、F 这 6 个字母组成一个长度为 4 的字符串, 要求字符串中至少包含一个 A 和一个 C, 字母可以重复使用, 可以组成的不同的字符串有 () 种。

- A. 900
- B. 302
- C. 648
- D. 324

11. 假设栈 s 的初始状态为空, 存在 e1~e6 六个互不相同的数据, 若这六个元素以

e2,e4,e6,e1,e3,e5 的顺序入栈 s, 同时不允许进行连续 3 次及以上的出栈操作, 那么下列合法的出栈序列为()。

- A. e1,e6,e3,e4,e2,e5
- B. e1,e6,e4,e2,e3,e5
- C. e2,e6,e4,e3,e1,e5
- D. e2,e6,e3,e4,e1,e5

12. 前缀表达式 $+ / 3 2 - 1 4$ 的值为 ()

- A. -2.4
- B. -1.5
- C. 4.5
- D. 3.6

13. 有以下结构体说明和变量定义, 如图所示, 指针 p、q、r 分别指向一个链表中的三个连续结点。

```
struct Node {  
    int data;  
    node *next;  
} *p, *q, *r;
```

现要将 q 和 r 所指结点的先后位置交换, 同时要保持链表的连续, 以下程序段中错误的是 ()。

- A. q->next = r->next; p->next = r; r->next = q;
- B. p->next = r; q->next = r->next; r->next = q;
- C. q->next = r->next; r->next = q; p->next = r;
- D. r->next = q; q->next = r->next; p->next = r;

14. 下面 C++ 代码用于计算整数 N 的位数, 比如输入 123 则输出: 123 是 3 位整数, 但代码中可能存在 bug。下面有关描述正确的是 ()

```
1  int N, N0, rc=0;  
2  cout << "请输入整数: ";  
3  cin >> N;  
4  
5  N0 = N;  
6  while(N){  
7      rc++;  
8      N /= 10;  
9  }  
10  
11 printf("%d 是%d 位整数\n", N, rc); //L11
```

- A. 变量 N0 占用额外空间, 可以去掉
- B. 代码对所有整数都能计算出正确位数
- C. L11 标记的代码行简单修改后可以对正整数给出正确输出

D. L11 标记的代码行的输出格式有误

15. 明明得到 4 个格雷码: 1110, 1010, 1011, 1001。请问这些格雷码对应下列哪个选项的四个数字 ()。

- A. 8 9 10 11
- B. 9 10 11 12
- C. 10 11 12 13
- D. 11 12 13 14

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

(1)

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cmath>
4 using namespace std;
5 int main()
6 {
7     const double pi = 3.14159;
8     double s = 0, a = 0, b = 0, c = 0;
9     cin >> s;
10    a = round(pi * s);
11    c = pi * a;
12    swap(a,c);
13    b = c + ceil(a);
14    printf("%.4lf",b);
15    return 0;
16 }
```

程序输入的 s 为大于 0 且小于 100 的实数。

判断题

- 16. 将第 7 行的 $\text{pi} = 3.14159$ 改为 $\text{pi} = 3.14$ 程序会报语法错误。()
- 17. 最终输出的数据, 小数点后面一定是 4 个 0。()
- 18. 第 14 行语法错误, 想要输出变量 b 的值应写为 $\text{printf}("%.4lf", \&b)$;()
- 19. 将第 12 行去掉, 程序输出结果不变。()

单选题

20. (2 分) 当输入为 2.31, 输出为 ()
- A. 28.0000
 - B. 28.9911
 - C. 29.0000
 - D. 29.9911

(2)

```
1 #include<iostream>
2 #include<cstdio>
3 using namespace std;
4 int f1(int n,int m){
5     int res = 1;
6     for (int i = n;i >= n - m + 1;i--){
7         res *= i;
8     }
9     return res;
10 }
11 int f2(int n,int m){
12     int res = 1;
13     for (int i = n,j = 1;j <= m;i--,j++){
14         res *= i;
15         res /= j;
16     }
17     return res;
18 }
19 int main(){
20     int n = 0,m = 0;
22     cin >> n >> m;
22     cout << f1(n,m) - f2(n,m) << endl;
23     return 0;
24 }
```

输入数据均为正整数 且 $1 \leq m \leq n \leq 10$ 。

判断题

- 21. 该程序的输出一定是正整数。()
- 22. 第 15 行会出现 res 不能被 j 整除的情况。()
- 23. f1 函数在算 n 的阶乘 ()
- 24. 当输入的 n 和 m 一样时, f1 和 f2 函数返回的结果是一样的 ()

单选题

- 25. (2 分) 下列哪一组数据输出的答案最大 ()
A. 8 5 B. 9 3 C. 10 10 D. 10 5
- 26. (2 分) 若输入 7 6, 输出为 ()
A. 5033 B. 714 C. 2499 D. 114514

(3)

```
01 #include<iostream>
02 #include<vector>
03 #include<cstring>
04 #include<algorithm>
05 using namespace std;
06 int n;
07 string s;
08 int work(string s)
09 {
10     int len = s.size();
11     if (!len) return 0;
12     vector<vector<int>> dp(len,vector<int>(len,0));
13     for (int i = 0;i < len;i++) dp[i][i] = 1;
14     for (int i = len - 1;i >= 0;i--)
15     {
16         for (int j = i + 1;j < len;j++)
17         {
18             if (s[i] == s[j]) dp[i][j]=dp[i + 1][j - 1] + 2;
19             else dp[i][j]=max(dp[i + 1][j],dp[i][j - 1]);
20         }
21     }
22     return dp[0][len - 1];
23 }
24 int main()
25 {
26     cin >> s;
27     s = s.substr(1);
28     cout << work(s);
29     return 0;
30 }
```

输入的字符串长度大于 1，且最多不超过 500。

判断题

27. 该程序的输出最小的情况下是 1 ()。
28. 该程序在求解一个字符串的最长回文子串的长度 ()。
29. 将第 14 行改为: for(int i = 0;i < len;i++)程序行为不变 ()。
30. 若将 work 函数中的形参删去, 28 行调用的实参删去, 程序不会发生语法错误, 但最终输出的结果可能会发生改变 ()。

单选题

31. 输入为 abba 时, 输出 ()。
- A. 4 B. 3 C. 2 D. 1
32. 若输入 azcshrfgczaabbcbvzasadbglj, 输出为 ()。
- A. 1 B. 10 C. 4 D. 7

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

(1) (十六进制转八进制)

给定 n 个十六进制正整数, 输出它们对应的八进制数。(十六进制数中的字母均为大写)
十六进制转换八进制: 将十六进制数转换为二进制数后从右至左分成三位一组, 然后将每组转换成对应的八进制数。

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cmath>
4 using namespace std;
5 string jz[16] =
6 {"0000","0001","0010","0011","0100","0101","0110","0111","1000","1001",
7  "1010","1011","1100","1101","1110","1111"};
8 string s;
9 int n;
10 int main()
11 {
12     cin >> n;
13     while (n--)
14     {
15         cin >> s;
16         string temp = "";
17         for (int i = 0; i < s.size(); i++)
18         {
19             if (s[i] >= 'A' && s[i] <= 'Z')
20             {
21                 temp += jz[ ① ];
22             }
23             else
24             {
25                 temp += jz[s[i] - '0'];
26             }
27         }
28         while( ② )
```

```

28     {
29         temp = "0" + temp;
30     }
31     string res = "";
32     for (int i = 0; i < temp.size(); ③ )
33     {
34         int ans = (temp[i] - '0') * 4 + (temp[i + 1] - '0') * 2 + (temp[i + 2] - '0') * 1;
35         res += ④ ;
36     }
37     int k = 0;
38     while ( ⑤ && k < res.size() - 1) k++;
39     for (int i = k; i < res.size(); i++) cout << res[i];
40     cout << endl;
41 }
42 return 0;
43 }

```

单选题

33. ①处应填 ()。
- A. s[i] + 10 B. s[i] - 'a' + 10 C. s[i] - 'A' + 10 D. s[i]
34. ②处应填 ()。
- A. temp.size() % 4 != 0 B. temp.size() % 3 != 0
C. (temp.size() - 1) % 3 != 0 D. (temp.size() - 1) % 4 != 0
35. ③处应填 ()。
- A. i += temp.size() B. i += 4
C. i++ D. i += 3
36. ④处应填 ()。
- A. ans * 8 + '0' B. temp[i+2] + temp[i + 1] + temp[i]
C. ans D. ans + '0'
37. ⑤处应填 ()。
- A. res[k + 1] = '0' B. res[k] == '0'
C. res[0] = '0' D. res[res.size() - 1] = '0'

(2) (购票问题)

儿童节到了，公园的售票处排起了长龙。按售票处规定，每位购票者限购一张门票，每张门票售价 50 元。现在有 n 个人手持 50 元排队，有 n 个人手持 100 元排队。假设售票处在开始售票的时候没有零钱，试问这 $2n$ 个人有多少种排队方式可以使售票处不会出现找不开钱的尴尬局面。

例如当 $n = 2$ 时，可以有 50 100 50 100，50 50 100 100 这两种排队方式。试补全模拟程序。


```

1 #include <iostream>
2 using namespace std;
3 typedef long long LL;
4 LL k,sum,n;
5 void dfs(int dep)
6 {
7     if (    ①    )
8     {
9         if (    ②    ) sum++;
10        return;
11    }
12    if (    ③    )
13    {
14        ④
15        dfs(dep + 1);
16        ⑤
17    }
18    k++;
19    dfs(dep + 1);
20    k--;
21 }
22 int main()
23 {
24     scanf("%lld",&n);
25     k = 0; //售票处 50 元的张数
26     sum = 0;
27     dfs(0);
28     printf("%lld",sum);
29     return 0;
30 }

```

单选题

38. ①处应填 ()。
- A. $dep == n$ B. $dep == 2 * n$
C. $dep > 2 * n$ D. $dep > n$
39. ②处应填 ()。
- A. $!k$ B. $dep == 2 * n$ C. $k != 0$ D. $sum == 0$
40. ③处应填 ()。
- A. $sum > 0$ B. $k > 0$ C. $k >= 0$ D. $dep >= 0$

41. ④处应填 ()。

A. k++; B. k--;

C. sum++; D. dfs(dep - 1);

42. ⑤处应填 ()。

A. k++; B. k--;

C. sum--; D. return;