

# YDSP Junior 组模拟考试

Copyright ©云斗学院 ©北斗学友教育科技有限公司

2023 年 9 月 10 日

本次考试的答题时间为两个小时，请各位选手自行掌握时间。

提交请访问 [yundouxueyuan.com](http://yundouxueyuan.com) 参加对应的比赛，将最终答案以程序的形式提交至对应题目。

更进一步的提交细节，请参考对应题目里的说明。

最后，本次模拟考试为纯公益目的，不得用于任何未经授权的盈利活动。

\* 祝考试顺利 \*

# 2023 云斗学院软件能力认证第一轮 (YDSP - Junior) 入门级 C++ 语言试题

---

考生注意事项:

- 试题纸一共 14 页，满分 100 分。作答后请记录答案，并按要求提交至对应比赛处。
- 考试过程中不得使用任何电子设备或查阅任何书籍资料。

## 1 单选题（每题 2 分，共 30 分）

### 1.1 第 1 题

CCF 的全称是 ( )。

- A. Coin Collecting Federation
- B. China Cheating Federation
- C. China Computer Federation
- D. Chinese Computer Foundation

### 1.2 第 2 题

集合  $[1, 100] \cup [32, 44] \cup [111, 144] \cup [199, 211] \cup [100, 112]$  中有 ( ) 个整数。

- A. 157
- B. 156
- C. 173
- D. 211

### 1.3 第 3 题

$(114)_7$  和  $(514)_{16}$  的最大公约数的十进制表示是 ( )。

- A. 2
- B. 4
- C. 6
- D. 1300

### 1.4 第 4 题

下列 ( ) 不是计算机的存储设备。

- A. RAM
- B. U 盘
- C. 硬盘
- D. 光盘驱动器

### 1.5 第 5 题

以序列 ABCDEFG，对一个初始为空的栈依次进行入栈、入栈、出栈、入栈、入栈、入栈、出栈、入栈、出栈、出栈、入栈的操作，最终栈中的元素从栈顶到栈底依次为 ( )。

- A. ACDG
- B. ACG
- C. EFG
- D. CDG

### 1.6 第 6 题

语句 `freopen("input","r",stdin)` 的含义解释正确的是 ( )。

- A. 从 `input.txt` 中读入文件
- B. 输出到 `input.in`
- C. 从 `input` 读入文件
- D. 输出到 `input`

### 1.7 第 7 题

对于两个布尔变量  $x, y$ ，前缀表达式 `&& || x y ! && x y` 等价于中缀表达式 ( )。

- A. `x and y`
- B. `x or y`
- C. `x xor y`
- D. `x == y`

### 1.8 第 8 题

大浮点数加上小浮点数会产生很大的精度损失。在计算正浮点数总和时，为了减小精度误差，一种算法是每次取最小的两个，并用和替代这两个浮点数。这个过程接近于 ( )。

- A. 求图上两点间最短路径
- B. 求一定背包容量能装下物品价值和最大值
- C. 画一棵完全二叉树
- D. 求一堆数字的哈夫曼编码

### 1.9 第 9 题

前序遍历为 IHEABCDGF，后序遍历为 ABEGFDCHI 的二叉树，中序遍历不可能是(cannot be) ( )。

- A. AEBHGDCFI
- B. AEBHCGDFI
- C. IAEBHGDFC
- D. IAEBHCGDF

### 1.10 第 10 题

如果  $a, b, c$  均为 `int` 类型且绝对值不超过  $10^9$ ，那么 C++ 中下列一定成立的是 ( )。

- A.  $\max(a, b) + c == \max(a, b + c)$
- B.  $\max(a, b) * 111 * c == \max(a * 111 * c, b * 111 * c)$
- C.  $\max(a * 2, b * 2) + c == \max(a * 2 + c, b * 2 + c)$
- D.  $2 * \min(a, b) == a + b - \text{abs}(a - b)$

### 1.11 第 11 题

现有 24 个一样的苹果，要分给三个小朋友，且每个小朋友至少获得 2 个，方案数有 ( ) 种。

- A. 105
- B. 190
- C. 210
- D. 380

### 1.12 第 12 题

现有数列  $a$ ，规定  $S(l, r) = a_l + a_{l+1} + \dots + a_r$ ，且  $S(1, 12) = 31$ ， $S(7, 12) = 17$ ， $S(10, 20) = 16$ ， $S(7, 20) = 24$ ，那么  $S(1, 9)$  为 ( )。

- A. 22
- B. 23
- C. 24
- D. 25

### 1.13 第 13 题

一款音乐游戏的计分方式是，假如某局游戏有  $n$  个音符，那么每个音符的基础分  $m$  为  $\frac{10^7}{n}$ ；对于每个音符，可能有 Max Pure, Pure, Far, Lost 四种判定，分别可以拿  $m + 1, m, 0.5m, 0$  分。

称一次游玩“PM”，当且仅当所有音符都是 Max Pure 或 Pure，不难发现 PM 得分一定不小于  $10^7$ ；称一次游玩“伪 PM”，当且仅当这次游玩分数不小于  $10^7$ ，且存在 Far 或 Lost。

这款游戏中有三首曲目名称以 T 开头的歌曲，分别有 2221, 1540, 1392 个音符。这三首歌曲中，有 ( ) 首可能产生伪 PM。

- A. 0
- B. 1
- C. 2
- D. 3

### 1.14 第 14 题

下列程序的输出是 ( )。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main(){
4     string s="yUMmyadoRAbLE";
```

```

5     for(int i=1;i<s.size();i++)
6         if( 'A'<s[i] && s[i]<='Z')
7             swap(s[i],s[s.size()-i]);
8     cout<<s;
9     return 0;
10 }

```

- A. yUMmyadoRAbLE
- B. yUMmyRdoaAbLE
- C. yUMmARdoaybLE
- D. EUMmRadoyAbLy

### 1.15 第 15 题

$G$  有 6 个结点 1,2,3,4,5,6 以及 7 条边 (1,2)(1,3)(2,4)(3,4)(4,5)(4,6)(5,6)。若以 ( ) 为起点, 对这张图进行深度优先遍历, 得到的遍历序可能性最多。

- A. 1
- B. 2 或 3
- C. 4
- D. 5 或 6

## 2 阅读程序 (无特殊说明时判断 1.5 分, 选择 3 分, 3 题共 40 分)

### 2.1 第 1 题 (12 分)

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 union yun{
4     int a[3];
5     long long b;
6 }u,v;
7 int main(){
8     #define S(x) x*x
9     cout<<S(3<<1)<<'\n';
10    u.a[0]=1;
11    u.a[1]=2;
12    u.a[2]=0;
13    cout<<sizeof sizeof v<<endl;
14    cin>>v.b;
15    switch(v.b){
16    case 1:
17        putchar((0110 ^ 101)-35);
18    case 2:
19        cout<<u.a[v.b[u.a][u.a]];

```

```

20     return 0;
21 case0:
22     cout<<v.b+5;
23 default:
24     memset(&v,v.b,sizeof v);
25     int i=2;
26     while(i --> 0)
27         cout<<hex<<max(v.a[i],1);
28 }
29 return 0;
30 }

```

### 2.1.1 判断题

1. 输出的第二行是 8。
2. 如果输入 400，那么程序会出错。
3. 在第 13 行执行后输出 u.b 是未定义行为。
4. 第 27 行换成 `cout<<hex<<(v.a[i]>100?v.a[i]:1);`，效果不变。

### 2.1.2 选择题

5. 输出的第一行是 ▲。
  - A. 36
  - B. 48
  - C. 192
  - D. 前三个选项都不正确
6. 在输出的第三行中过滤非数字字符后，可能得到下列哪个数字？
  - A. 1
  - B. 511
  - C. 20202022020202
  - D. 前三个选项都不正确

## 2.2 第 2 题 (15 分)

阅读下面的程序，回答问题。

```

1 #include <cstdio>
2
3 #define base 2

```

```

4 #define Y 1799
5 using namespace std;
6 int d_y[3010], d_m[13];
7 int sum_y[3010], sum_m[13];
8 int check(int i){
9     return !(i % 400) || i % 100 && !(i % 4);
10 }
11 void init(){
12     for(int i = Y + 1 ; i <= 3000 ; i++){
13         d_y[i] = 365 + check(i);
14         sum_y[i] = sum_y[i - 1] + d_y[i];
15     }
16     for(int i = 1 ; i <= 12 ; i++){
17         if((i < 8 && i & 1) || (i > 7 && !(i & 1))) d_m[i] = 31;
18         else if(i == 2) d_m[i] = 28;
19         else d_m[i] = 30;
20         sum_m[i] = sum_m[i - 1] + d_m[i];
21     }
22 }
23 int main(){
24     init();
25     int y, m, d, ans;
26     scanf("%d%d%d", &y, &m, &d);
27     ans = sum_y[y - 1] + sum_m[m] - d_m[m] + d + base;
28     if(check(y) && m > 2 || (m == 2 && d == 29)) ans++;
29     printf("%d\n", ans % 7);
30     return 0;
31 }

```

程序满足输入  $y \geq 1800$ ,  $m, d$  均合法。

### 2.2.1 (一) 判断题

1. 第 28 行中的判断  $(m == 2 \ \&\& \ d == 29)$  多余。
2. 当  $y = 4 \times 10^3$  时, 程序仍能输出正确结果。
3. 将第 17 行的所有不等号后添加  $=$  号, 对答案没有影响。
4. 应将  $\text{sum}_y$  数组的  $\text{int}$  类型改为  $\text{long long}$  类型。

### 2.2.2 (二) 选择题

5. 将  $Y$  与  $\text{base}$  分别替换为 ▲ 时, 对程序输出没有影响。

A. 1798, 3

B. 1798, 1

C. 1799, 2 \* (int)cos(0)

D. 1800, 3

6. 输入为 2019 3 1 时, 程序输出为     ▲    。

A. 3

B. 4

C. 5

D. 6

7. 下列做法中, 对程序结果有影响的是     ▲     (除了该做法外不进行任何其他操作)。

A. 对于数组  $d_m$ , 不使用数组而仅用一个变量对数组  $sum_m$  进行累加。

B. 将  $sum_y$  数组中所有数初始化为  $365 \wedge 1$ 。

C. 在第 24 行不调用 `init()` 而在第 26 行后调用 `init()`。

D. 交换第 27, 28 两行。

### 2.3 第 3 题 (13 分)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int f(int &n,int m)
4 {
5     if(n==m) return n;
6     int A=0;
7     if(n>m) A=f(n--m,m);
8     else A=f(n,m--n);
9     return A+n;
10 }
11 int main()
12 {
13     int n,m;
14     cin>>n>>m;
15     cout<<f(n,m);
16     return 0;
17 }
```

若无其它限制, 本题默认输入的  $n, m$  是  $[1, 10^9]$  内的整数。



### 2.3.1 判断题

1. 将第 8 行的 `m==n` 改为 `m-n`，程序可以编译且运行结果不变。
2. 程序的执行过程中可能会出现有符号整数溢出。
3. 将第 15 行改为 `cin>>m>>n`，程序运行结果不变。
4. 如果输入中  $n = 0$  或  $m = 0$ ，程序可能会进入无限循环。

### 2.3.2 选择题

5. 程序的最坏时间复杂度为     ▲    。  
A.  $O(\log nm)$   
B.  $O(\log n \cdot \log m)$   
C.  $O(n + m)$   
D.  $O(nm)$
6. (本题 4 分) 如果输入 10 12，输出为     ▲    。  
A. 9  
B. 12  
C. 15  
D. 18

## 3 完善程序 (2 题, 每空 3 分, 共 30 分)

### 3.1 冒泡排序

将要排序的数全部放进双向链表，然后从小到大排序并输出。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 struct node{
4     int val=0;
5     node *prv=NULLPTR,*nxt=NULLPTR;
6 };
7 int main(){
8     int n;
9     scanf("%d",&n);
10    node *head=new node,*tail= /* Blank 1 */ ;
11    for(int i=1;i<=n;i++){
12        int x;
13        scanf("%d",&x);
```

```

14     node *tmp=head;
15     head=new node;
16     head->val=x;
17     head->nxt=tmp;
18     tmp->prv=head;
19 }
20 tail->nxt=head;
21 head->prv=tail;
22 for(node *i=head;i!=tail;i=i->nxt)
23     for(node *j=/* Blank 2 */){
24         node *p=j->prv;
25         if(p->val > j->val){
26             #define Symm j->prv->nxt=j; p->nxt->prv=p;
27             #define Swap j->prv=p->prv; p->nxt=j->nxt;
28             #define Reve j->nxt=p;      p->prv=j;
29                 /* Blank 3 */
30             j=p;
31             if(p==i){
32                 /* Blank 4 */
33             }
34         }
35     }
36     node *it = /* Blank 5 */;
37 for(int i=1;i<=n;i++){
38     it=it->nxt;
39     printf("%d ",it->val);
40 }
41 return 0;
42 }

```

1. 第 /\* Blank 1 \*/ 空应该填 ▲

- A. nullptr
- B. new node
- C. head
- D. &head

2. 第 /\* Blank 2 \*/ 空应该填 ▲

- A. i->nxt;j!=tail;j=j->nxt
- B. tail;j!=i->prv;j=j->prv
- C. tail;j!=i;j=j->prv

D. `tail->prv;j!=i;j=j->prv`

3. 第 /\* Blank 3 \*/ 空应该填 ▲

A. `Swap Symm Reve`

B. `Reve Symm Swap`

C. `Symm Reve Swap`

D. `Symm Swap Reve`

4. 第 /\* Blank 4 \*/ 空应该填 ▲

A. `i=i->prv;break;`

B. `j=j->prv;`

C. `j=j->nxt;`

D. `i=p;break;`

5. 第 /\* Blank 5 \*/ 空应该填 ▲

A. `head`

B. `head->prv`

C. `tail`

D. `tail->prv`

### 3.2 数独变换（第三个空 3 分，其余的 2 分）

初始给定一个有  $n \times n$  个宫，每个宫中有  $n \times n$  个元素，且早已全部正确填好的  $n$  阶数独。之后，会将一些宫向左或者向右转 90 度/180 度。例如，若一个宫初始为

```
1 087
2 654
3 321
```

那么它向左旋转 90 度后会变成：

```
1 741
2 852
3 063
```

现在你需要对数独进行恢复。在恢复数独时，也只能将一些宫向左转 90 度，一次旋转算作一步。求把数独重新恢复成合法的数独，最少需要多少步。如果一开始给出的数独局面不可以通过任意次、任意位置的左旋得到，则输出  $-1$ 。

数据保证当存在解时，最优解方案唯一。 $2 \leq n \leq 4$ 。

提示：

1.  $n$  阶数独合法的条件：每一行、每一列、每一个粗线宫 ( $n \times n$ ) 内的数字均含  $0 \sim n^2 - 1$ ，且不重复。

需要注意的是，本题内对于 4 阶数独的表示方式中  $> 9$  的数字采用了十六进制表示法。准确来说， $A = 10, B = 11, C = 12, D = 13, E = 14, F = 15$ 。

2. 输出格式如下：

第 1 行输出一个整数，表示最小步数  $s$ 。

第 2  $\sim s + 1$  行，每行输出两个整数  $x_i, y_i$ 。表示对行号列号为  $x_i, y_i$  的宫向左旋转了 90 度。

若不存在合法方案，请输出  $-1$ 。

3. 解决思路：深度优先搜索，但是在此基础上有两个优化：

- 程序会定期检查当前行和全部列是否已经正确填写，否则不会继续进行。 - 保证 dfs 运行到最后时的答案单调不升。

试补全程序。

```
1 #include <bits/stdc++.h>
2
3 using namespace std ;
4
5 const int N = 110 ;
6 const int Inf = 998244353 ;
7
8 int n ;
9 int ans ;
10
11 char a[N][N] ;
12
13 int buc[N] ;
14 int tmp[N][N] ;
15 int res[N][N] ;
16 int base[N][N] ;
17
18 bool chkRow(int x){
19     x = /* Blank 2 */ ;
20     for (int i = x ; i < x + n ; ++ i){
21         memset(buc, 0, sizeof(buc)) ;
22         for (int j = 1 ; j <= n * n ; ++ j){
23             if ( /* Blank 3(1)*/ ) return 0 ;
24             /* Blank 3(2)*/
25         }
26     }
27     return 1 ;
28 }
29 bool chkColumn(){
30     for (int i = 1 ; i <= n * n ; ++ i){
31         memset(buc, 0, sizeof(buc)) ;
32         for (int j = 1 ; j <= n * n ; ++ j){
```

```

33     if (buc[base[j][i]]) return 0 ;
34     buc[base[j][i]] = 1 ;
35 }
36 }
37 return 1 ;
38 }
39 void rot(int x, int y){
40     ++ tmp[x][y] ;
41     x = n * (x - 1) + 1 ;
42     y = n * (y - 1) + 1 ;
43     if (/* Blank 1 */){
44         swap(base[x + 1][y], base[x][y + 1]) ;
45         swap(base[x + 2][y], base[x][y + 2]) ;
46         swap(base[x + 3][y], base[x][y + 3]) ;
47         swap(base[x + 2][y + 1], base[x + 1][y + 2]) ;
48         swap(base[x + 3][y + 1], base[x + 1][y + 3]) ;
49         swap(base[x + 3][y + 2], base[x + 2][y + 3]) ;
50
51         swap(base[x + 1][y], base[x + 2][y]) ;
52         swap(base[x + 1][y + 1], base[x + 2][y + 1]) ;
53         swap(base[x + 1][y + 2], base[x + 2][y + 2]) ;
54         swap(base[x + 1][y + 3], base[x + 2][y + 3]) ;
55         swap(base[x][y], base[x + 3][y]) ;
56         swap(base[x][y + 1], base[x + 3][y + 1]) ;
57         swap(base[x][y + 2], base[x + 3][y + 2]) ;
58         swap(base[x][y + 3], base[x + 3][y + 3]) ;
59     }
60     else {
61         swap(base[x + 1][y], base[x][y + 1]) ;
62         swap(base[x + 2][y], base[x][y + 2]) ;
63         swap(base[x + 2][y + 1], base[x + 1][y + 2]) ;
64
65         swap(base[x][y], base[x + 2][y]) ;
66         swap(base[x][y + 1], base[x + 2][y + 1]) ;
67         swap(base[x][y + 2], base[x + 2][y + 2]) ;
68     }
69 }
70 int cnt;
71 void dfs(int c, int r, int s){
72     ++ cnt ;
73     if (s > ans) return ;
74     if (c == n + 1){

```

```

75     if (/* Blank 4 */) return ;
76     if (/* Blank 5 */){
77         if (ans > s){
78             ans = s ;
79             for (int i = 1 ; i <= n ; ++ i)
80                 for (int j = 1 ; j <= n ; ++ j)
81                     res[i][j] = tmp[i][j] ;
82         }
83         return ;
84     }
85     if (r == n) return ;
86     return dfs(1, r + 1, s) ;
87 }
88 dfs(c + 1, r, s) ;
89 rot(r, c) ;
90     dfs(c + 1, r, s + 1) ;
91 rot(r, c) ;
92     dfs(c + 1, r, s + 2) ;
93 rot(r, c) ;
94     dfs(c + 1, r, s + 3) ;
95 rot(r, c) ;
96     /* Blank 6 */
97 }
98 int main(){
99     cin >> n ; ans = Inf ;
100     memset(res, 0, sizeof(res)) ;
101     memset(tmp, 0, sizeof(tmp)) ;
102     for (int i = 1 ; i <= n * n ; ++ i)
103         for (int j = 1 ; j <= n * n ; ++ j){
104             cin >> a[i][j] ;
105             if (isdigit(a[i][j]))
106                 base[i][j] = a[i][j] - '0' ;
107             else base[i][j] = a[i][j] - 'A' + 10 ;
108         }
109     dfs(1, 1, 0) ;
110     cout << (ans == Inf ? -1 : ans) << endl ;
111     for (int i = 1 ; i <= n ; ++ i)
112         for (int j = 1 ; j <= n ; ++ j)
113             for (int o = 1 ; o <= /* Blank 7 */ ; ++ o)
114                 printf("%d %d\n", i, j) ;
115     return 0 ;
116 }

```

1. 第 /\* Blank 1 \*/ 空应该填 ▲

- A. `n == 3`
- B. `n == 4`
- C. `n >= 3`
- D. `n <= 3`

2. 第 /\* Blank 2 \*/ 空应该填 ▲

- A. `n * x - n + 1`
- B. `n * x`
- C. `n * x - n`
- D. `n * x - n - 1`

3. 第 /\* Blank 3(1) \*/ 空和第 /\* Blank 3(2) \*/ 空分别应该填 ▲ 和 ▲

- A. (1) `buc[base[j][i]]` (2) `buc[base[j][i]] ++` ;
- B. (1) `buc[base[i][j]]` (2) `buc[base[i][j]] ++` ;
- C. (1) `buc[base[i][j]] > 1` (2) `buc[base[i][j]] ++` ;
- D. (1) `buc[base[j][i]] > 1` (2) `buc[base[j][i]] ++` ;

4. 第 /\* Blank 4 \*/ 空应该填 ▲

- A. `chkColumn()`
- B. `!chkColumn()`
- C. `chkRow(n)`
- D. `!chkRow(n)`

5. 第 /\* Blank 5 \*/ 空应该填 ▲

- A. `chkRow(n) and r == n`
- B. `chkColumn() and r == n`
- C. `r == n`
- D. `chkRow(1) and r == n`

6. 第 /\* Blank 6 \*/ 空应该填 ▲

- A. `tmp[r][c] --` ;
- B. `res[r][c] = 0;`
- C. `tmp[r][c] -= 4` ;

D. `c -- ;`

7. 第 `/* Blank 7 */` 空应该填 ▲

A. `res[i][j]`

B. `base[i][j]`

C. `tmp[i][j]`

D. `res[i][j] * tmp[i][j]`