

CSP-S 2022

第一轮真题 · 试卷与答案

作答时间: 120 分钟 姓名: _____

满分 100 分, 共 43 个作答点。判断题填写 A 或 B; 参考答案附于卷末。

一、单项选择 (共 30 分)

(共 15 题, 每题 2 分, 共计 30 分; 每题有且仅有一个正确选项)

第1题 (2 分)

在 Linux 系统终端中, 用于切换工作目录的命令为 ()。

- A. ls
- B. cd
- C. cp
- D. all

第2题 (2 分)

你同时用 time 命令和秒表为某个程序在单核 CPU 的运行计时。假如 time 命令的输出如下:

```
1 | real    0m30.721s
2 | user    0m24.579s
3 | sys     0m6.123s
```

以下最接近秒表计时的时长为 ()。

- A. 30s
- B. 24s
- C. 18s
- D. 6s

第3题 (2 分)

若元素 a、b、c、d、e、f 依次进栈, 允许进栈、退栈操作交替进行, 但不允许连续三次退栈操作, 则不可能得到的出栈序列是 ()。

- A. dcebfa
- B. cbdaef
- C. bcaefd
- D. afedcb

第4题 (2 分)

考虑对 n 个数进行排序, 以下最坏时间复杂度低于 $O(n^2)$ 的排序方法是 ()。

- A. 插入排序
- B. 冒泡排序
- C. 归并排序
- D. 快速排序

第5题 (2 分)

假设在基数排序过程中，受宇宙射线的影响，某项数据异变为一个完全不同的值。请问排序算法结束后，可能出现的最坏情况是（ ）。

- A. 移除受影响的数据后，最终序列是有序序列
 - B. 移除受影响的数据后，最终序列是前后两个有序的子序列
 - C. 移除受影响的数据后，最终序列是一个有序的子序列和一个基本无序的子序列
 - D. 移除受影响的数据后，最终序列基本无序
-

第6题 (2 分)

计算机系统用小端（Little Endian）和大端（Big Endian）来描述多字节数据的存储地址顺序模式，其中小端表示将低位字节数据存储在低地址的模式、大端表示将高位字节数据存储在低地址的模式。在小端模式的系统和大端模式的系统分别编译和运行以下 C++ 代码段表示的程序，将分别输出什么结果？（ ）

```
1 unsigned x = 0xDEADBEEF;  
2 unsigned char *p = (unsigned char *)&x;  
3 printf("%X", *p);
```

- | | |
|----------|----------|
| A. EF、EF | B. EF、DE |
| C. DE、EF | D. DE、DE |
-

第7题 (2 分)

一个深度为 5（根结点深度为 1）的完全 3 叉树，按前序遍历的顺序给结点从 1 开始编号，则第 100 号结点的父结点是第（ ）号。

- | | |
|-------|-------|
| A. 95 | B. 96 |
| C. 97 | D. 98 |
-

第8题 (2 分)

强连通图的性质不包括（ ）：

- | | |
|-------------------|------------------|
| A. 每个顶点的度数至少为 1 | B. 任意两个顶点之间都有边相连 |
| C. 任意两个顶点之间都有路径相连 | D. 每个顶点至少都连有一条边 |
-

第9题 (2 分)

每个顶点度数均为 2 的无向图称为“2 正规图”。由编号为从 1 到 n 的顶点构成的所有 2 正规图，其中包含欧拉回路的不同 2 正规图的数量为（ ）。

- | | |
|-----------|---------------|
| A. $n!$ | B. $(n-1)!$ |
| C. $n!/2$ | D. $(n-1)!/2$ |
-

第10题 (2 分)

共有 8 人选修了程序设计课程，期末大作业要求由 2 人组成的团队完成。假设不区分每个团队内 2 人的角色和作用，请问共有多少种可能的组队方案。（ ）。

- A. 28
 - B. 32
 - C. 56
 - D. 64
-

第11题 (2 分)

小明希望选到形如 “省 A·LLDDD ” 的车牌号。车牌号在 “.” 之前的内容固定的 5 位号码中，前 2 位必须是大写英文字母，后 3 位必须是阿拉伯数字（L 代表 A 至 Z，D 表示 0 至 9，两个 L 和三个 D 之间可能相同也可能不同）。请问总共有多少个可供选择的车牌号。（ ）

- A. 20280
 - B. 52000
 - C. 676000
 - D. 1757600
-

第12题 (2 分)

给定地址区间为 0~9 的哈希表，哈希函数为 $h(x) = x \% 10$ ，采用线性探查的冲突解决策略（对于出现冲突情况，会往后探查第一个空的地址存储；若地址 9 冲突了则从地址 0 重新开始探查）。哈希表初始为空表，依次存储(71, 23, 73, 99, 44, 79, 89)后，请问 89 存储在哈希表哪个地址中。（ ）

- A. 9
 - B. 0
 - C. 1
 - D. 2
-

第13题 (2 分)

对于给定的 n ，分析以下代码段对应的时间复杂度，其中最为准确的时间复杂度为（ ）。

```
1 | int i, j, k = 0;  
2 | for (i = 0; i < n; i++) {  
3 |     for (j = 1; j < n; j*=2) {  
4 |         k = k + n / 2;  
5 |     }  
6 | }
```

- A. $O(n)$
 - B. $O(n \log n)$
 - C. $O(n\sqrt{n})$
 - D. $O(n^2)$
-

第14题 (2 分)

以比较为基本运算，在 n 个数的数组中找最大的数，在最坏情况下至少要做（ ）次运算。

- A. $n/2$
 - B. $n-1$
 - C. n
 - D. $n+1$
-

第15题 (2 分)

ack 函数在输入参数 "(2,2)" 时的返回值为 () 。

```
1 unsigned ack(unsigned m, unsigned n) {  
2     if (m == 0) return n + 1;  
3     if (n == 0) return ack(m - 1, 1);  
4     return ack(m - 1, ack(m, n - 1));  
5 }
```

A. 5

B. 7

C. 9

D. 13

二、阅读程序 (共 40 分)

(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)

第16题—第21题

```
1 1 #include <iostream>
2 2 #include <string>
3 3 #include <vector>
4 4
5 5 using namespace std;
6 6
7 7 int f(const string &s, const string &t)
8 8 {
9 9     int n = s.length(), m = t.length();
10 10
11 11     vector<int> shift(128, m + 1);
12 12
13 13     int i, j;
14 14
15 15     for (j = 0; j < m; j++)
16 16         shift[t[j]] = m - j;
17 17
18 18     for (i = 0; i <= n - m; i += shift[s[i + m]]){
19 19         j = 0;
20 20         while(j < m && s[i + j] == t[j]) j++;
21 21         if (j == m) return i;
22 22     }
23 23
24 24     return -1;
25 25 }
26 26
27 27 int main()
28 28 {
29 29     string a ,b;
30 30     cin >> a >> b;
31 31     cout << f(a, b) << endl;
32 32     return 0;
33 33 }
```

假设输入字符串由 ASCII 可见字符组成, 完成下面的判断题和单选题:

判断题

第16题 (1 分) (1 分) 当输入为 “abcde fg” 时, 输出为-1。

- A. 正确 B. 错误

第17题 (1.5 分) 当输入为 “abbababbbab abab” 时, 输出为 4。

- A. 正确 B. 错误

第18题 (1.5 分) 当输入为 “GoodLuckCsp2022 22” 时, 第 20 行的 “j++” 语句执行次数为 2。

单选题

A. 正确

B. 错误

第19题 (3 分) 该算法最坏情况下的时间复杂度为 ()。

A. $O(n + m)$

B. $O(n \log m)$

C. $O(m \log n)$

D. $O(nm)$

第20题 (3 分) $f(a, b)$ 与下列 () 语句的功能最类似。

A. $a.find(b)$

B. $a.rfind(b)$

C. $a.substr(b)$

D. $a.compare(b)$

第21题 (3 分) 当输入为 “baaabaaabaaabaaaa aaaa” , 第 20 行的 “j++” 语句执行次数为 ()。

A. 9

B. 10

C. 11

D. 12

第22题—第27题

```
1 1 #include <iostream>
2 2
3 3 using namespace std;
4 4
5 5 const int MAXN = 105;
6 6
7 7 int n, m, k, val[MAXN];
8 8 int temp[MAXN], cnt[MAXN];
9 9
10 10 void init()
11 11 {
12 12     cin >> n >> k;
13 13     for (int i = 0; i < n; i++) cin >> val[i];
14 14     int maximum = val[0];
15 15     for (int i = 1; i < n; i++)
16 16         if (val[i] > maximum) maximum = val[i];
17 17     m = 1;
18 18     while (maximum >= k) {
19 19         maximum /= k;
20 20         m++;
21 21     }
22 22 }
23 23
24 24 void solve()
25 25 {
26 26     int base = 1;
27 27     for (int i = 0; i < m; i++) {
28 28         for (int j = 0; j < k; j++) cnt[j] = 0;
29 29         for (int j = 0; j < n; j++) cnt[val[j] / base % k]++;
30 30         for (int j = 1; j < k; j++) cnt[j] += cnt[j - 1];
31 31         for (int j = n - 1; j >= 0; j--) {
32 32             temp[cnt[val[j] / base % k] - 1] = val[j];
33 33             cnt[val[j] / base % k]--;
34 34         }
35 35         for (int j = 0; j < n; j++) val[j] = temp[j];
36 36         base *= k;
37 37     }
38 38 }
39 39
40 40 int main()
41 41 {
42 42     init();
43 43     solve();
44 44     for (int i = 0; i < n; i++) cout << val[i] << ;
45 45     cout << endl;
46 46     return 0;
47 47 }
```

假设输入的 n 为不大于 100 的正整数, k 为不小于 2 且不大于 100 的正整数, $val[i]$ 在 `int` 表示范围内, 完成下面的判断题和单选题:

判断题

第22题 (1.5 分) 这是一个不稳定的排序算法。 ()

A. 正确

B. 错误

第23题 (1.5 分) 该算法的空间复杂度仅与 n 有关。 ()

A. 正确

B. 错误

第24题 (1.5 分) 该算法的时间复杂度为 $O(m(n+k))$ 。 ()

单选题

A. 正确

B. 错误

第25题 (3 分) 当输入为 “5 3 98 26 91 37 46” 时, 程序第一次执行到第 36 行, `val[]` 数组的内容依次为 ()。

A. 91 26 46 37 98

B. 91 46 37 26 98

C. 98 26 46 91 37

D. 91 37 46 98 26

第26题 (3 分) 若 `val[i]` 的最大值为 100, k 取 () 时算法运算次数最少。

A. 2

B. 3

C. 10

D. 不确定

第27题 (3 分) 当输入的 k 比 `val[i]` 的最大值还大时, 该算法退化为 () 算法。

A. 选择排序

B. 冒泡排序

C. 计数排序

D. 桶排序

第28题—第33题

```
1 1 #include <iostream>
2 2 #include <algorithm>
3 3
4 4 using namespace std;
5 5
6 6 const int MAXL = 1000;
7 7
8 8 int n, k, ans[MAXL];
9 9
10 10 int main(void)
11 11 {
12 12     cin >> n >> k;
13 13     if (!n) cout << 0 << endl;
14 14     else
15 15     {
16 16         int m = 0;
17 17         while (n)
18 18         {
19 19             ans[m++] = (n % (-k) + k) % k;
20 20             n = (ans[m - 1] - n) / k;
21 21         }
22 22         for (int i = m - 1; i >= 0; i--)
23 23             cout << char(ans[i] >= 10 ?
24 24                 ans[i] + 'A' - 10 :
25 25                 ans[i] + '0');
26 26         cout << endl;
27 27     }
28 28     return 0;
29 29 }
```

假设输入的 n 在 `int` 范围内, k 为不小于 2 且不大于 36 的正整数, 完成下面的判断题和单选题:

判断题

第28题 (1.5 分) 该算法的时间复杂度为 $O(\log_k n)$ 。

- A. 正确 B. 错误

第29题 (1.5 分) 删除第 23 行的强制类型转换, 程序的行为不变。

- A. 正确 B. 错误

第30题 (1.5 分) 除非输入的 n 为 0, 否则程序输出的字符数为 $\lfloor \log_k |n| \rfloor + 1$ 。

单选题

- A. 正确 B. 错误

第31题 (3 分) 当输入为 “100 7” 时, 输出为 ()。

- A. 202 B. 1515
C. 244 D. 1754

第32题 (3 分) 当输入为 “-255 8” 时, 输出为 ()。

A. 1400

B. 1401

C. 417

D. 400

第33题 (3 分) 当输入为 “1000000 19” 时, 输出为 ()。

A. BG939

B. 87GIB

C. 1CD428

D. 7CF1B

三、完善程序 (共 30 分)

第34题—第38题

三 完善程序 (单选题, 每小题 3 分, 共计 30 分) **(1) (归并第 k 小)** 已知两个长度均为 n 的有序数组 $a1$ 和 $a2$ (均为递增序, 但不保证严格单调递增), 并且给定正整数 k ($1 \leq k \leq 2n$), 求数组 $a1$ 和 $a2$ 归并排序后的数组里第 k 小的数值。

试补全程序。

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int solve(int *a1, int *a2, int n, int k) {
5      int left1 = 0, right1 = n - 1;
6      int left2 = 0, right2 = n - 1;
7      while (left1 <= right1 && left2 <= right2) {
8          int m1 = (left1 + right1) >> 1;
9          int m2 = (left2 + right2) >> 1;
10         int cnt = ①;
11         if (②) {
12             if (cnt < k) left1 = m1 + 1;
13             else right2 = m2 - 1;
14         } else {
15             if (cnt < k) left2 = m2 + 1;
16             else right1 = m1 - 1;
17         }
18     }
19     if (③) {
20         if (left1 == 0) {
21             return a2[k - 1];
22         } else {
23             int x = a1[left1 - 1], ④;
24             return std::max(x, y);
25         }
26     } else {
27         if (left2 == 0) {
28             return a1[k - 1];
29         } else {
30             int x = a2[left2 - 1], ⑤;
31             return std::max(x, y);
32         }
33     }
34 }
```

第34题 (3 分) ①处应填 ()

- | | |
|--------------------|--------------------------|
| A. $(m1 + m2) * 2$ | B. $(m1 - 1) + (m2 - 1)$ |
| C. $m1 + m2$ | D. $(m1 + 1) + (m2 + 1)$ |

第35题 (3 分) ②处应填 ()

- | | |
|-----------------------|-----------------------|
| A. $a1[m1] == a2[m2]$ | B. $a1[m1] <= a2[m2]$ |
| C. $a1[m1] >= a2[m2]$ | D. $a1[m1] != a2[m2]$ |
-

第36题 (3 分) ③处应填 ()

- | | |
|----------------------|----------------------|
| A. $left1 == right1$ | B. $left1 < right1$ |
| C. $left1 > right1$ | D. $left1 != right1$ |
-

第37题 (3 分) ④处应填 ()

- | | |
|----------------------------|------------------------|
| A. $y = a1[k - left2 - 1]$ | B. $y = a1[k - left2]$ |
| C. $y = a2[k - left1 - 1]$ | D. $y = a2[k - left1]$ |
-

第38题 (3 分) ⑤处应填 ()

- | | |
|----------------------------|------------------------|
| A. $y = a1[k - left2 - 1]$ | B. $y = a1[k - left2]$ |
| C. $y = a2[k - left1 - 1]$ | D. $y = a2[k - left1]$ |
-

第39题—第43题

(2) (容器分水) 有两个容器，容器 1 的容量为 a 升，容器 2 的容量为 b 升；同时允许下列的三种操作，分别为：

1. FILL(i)：用水龙头将容器 i ($i \in 1, 2$) 灌满水；
2. DROP(i)：将容器 i 的水倒进下水道；
3. POUR(i, j)：将容器 i 的水倒进容器 j (完成此操作后，要么容器 j 被灌满，要么容器 i 被清空)。

求只使用上述的两个容器和三种操作，获得恰好 c 升水的最少操作数和操作序列。上述 a 、 b 、 c 均为不超过 100 的正整数，且 $c \leq \max\{a, b\}$ 。

试补全程序。

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  const int N = 110;
4
5  int f[N][N];
6  int ans;
7  int a, b, c;
8  int init;
9
10 int dfs(int x, int y) {
11     if (f[x][y] != init)
12         return f[x][y];
13     if (x == c || y == c)
14         return f[x][y] = 0;
15     f[x][y] = init - 1;
16     f[x][y] = min(f[x][y], dfs(a, y) + 1);
17     f[x][y] = min(f[x][y], dfs(x, b) + 1);
18     f[x][y] = min(f[x][y], dfs(0, y) + 1);
19     f[x][y] = min(f[x][y], dfs(x, 0) + 1);
20     int t = min(a - x, y);
21     f[x][y] = min(f[x][y], ①);
22     t = min(x, b - y);
23     f[x][y] = min(f[x][y], ②);
24     return f[x][y];
25 }
26
27 void go(int x, int y) {
28     if (③)
29         return;
30     if (f[x][y] == dfs(a, y) + 1) {
31         cout << "FILL(1)" << endl;
32         go(a, y);
33     } else if (f[x][y] == dfs(x, b) + 1) {
34         cout << "FILL(2)" << endl;
35         go(x, b);
36     } else if (f[x][y] == dfs(0, y) + 1) {
37         cout << "DROP(1)" << endl;
38         go(0, y);
39     } else if (f[x][y] == dfs(x, 0) + 1) {
40         cout << "DROP(2)" << endl;
41         go(x, 0);
42     } else {
43         int t = min(a - x, y);
```

```

44         if(f[x][y] == ②) {
45             cout << "POUR(2,1)" << endl;
46             go(x + t, y - t);
47         } else {
48             t = min(x, b - y);
49             if (f[x][y] == ③) {
50                 cout << "POUR(1,2)" << endl;
51                 go(x - t, y + t);
52             } else
53                 assert(0);
54         }
55     }
56 }
57
58 int main() {
59     cin >> a >> b >> c;
60     ans = 1 << 30;
61     memset(f, 127, sizeof f);
62     init = **f;
63     if ((ans = dfs (0, 0)) == init - 1)
64         cout << "impossible";
65     else {
66         cout << ans << endl;
67         go (0, 0);
68     }
69 }

```

第39题 (3分) ①处应填 ()

- | | |
|-----------------------------------|-----------------------------------|
| A. $\text{dfs}(x + t, y - t) + 1$ | B. $\text{dfs}(x + t, y - t) - 1$ |
| C. $\text{dfs}(x - t, y + t) + 1$ | D. $\text{dfs}(x - t, y + t) - 1$ |

第40题 (3分) ②处应填 ()

- | | |
|-----------------------------------|-----------------------------------|
| A. $\text{dfs}(x + t, y - t) + 1$ | B. $\text{dfs}(x + t, y - t) - 1$ |
| C. $\text{dfs}(x - t, y + t) + 1$ | D. $\text{dfs}(x - t, y + t) - 1$ |

第41题 (3分) ③处应填 ()

- | | |
|------------------------------|-----------------------------|
| A. $x == c \parallel y == c$ | B. $x == c \ \&\& \ y == c$ |
| C. $x >= c \parallel y >= c$ | D. $x >= c \ \&\& \ y >= c$ |

第42题 (3分) ④处应填 ()

- | | |
|-----------------------------------|-----------------------------------|
| A. $\text{dfs}(x + t, y - t) + 1$ | B. $\text{dfs}(x + t, y - t) - 1$ |
| C. $\text{dfs}(x - t, y + t) + 1$ | D. $\text{dfs}(x - t, y + t) - 1$ |

第43题 (3 分) ⑤处应填 ()

A. $\text{dfs}(x + t, y - t) + 1$

B. $\text{dfs}(x + t, y - t) - 1$

C. $\text{dfs}(x - t, y + t) + 1$

D. $\text{dfs}(x - t, y + t) - 1$

参考答案

满分 100 分。判断题 A 表示正确，B 表示错误。

题号	分值	答案	题号	分值	答案
第1题	2	B	第2题	2	A
第3题	2	D	第4题	2	C
第5题	2	A	第6题	2	B
第7题	2	C	第8题	2	B
第9题	2	D	第10题	2	A
第11题	2	C	第12题	2	D
第13题	2	B	第14题	2	B
第15题	2	B	第16题	1	A
第17题	1.5	B	第18题	1.5	A
第19题	3	D	第20题	3	A
第21题	3	B	第22题	1.5	B
第23题	1.5	B	第24题	1.5	A
第25题	3	D	第26题	3	D
第27题	3	C	第28题	1.5	A
第29题	1.5	B	第30题	1.5	A
第31题	3	A	第32题	3	B
第33题	3	B	第34题	3	C
第35题	3	B	第36题	3	C
第37题	3	C	第38题	3	A
第39题	3	A	第40题	3	C
第41题	3	A	第42题	3	A
第43题	3	C			