

2020 CSP-S1 模拟赛1

一：单项选择题(共15题,每题1.5分,共计22.5分;每题有且仅有一个正确选项)

1: CSP非专业级别的认证是从哪一年开始的?

A: 2016 B:2017 C:2018 D:2019

2: 32位有符号整数一共可以表示多少个数

A: 4294967295 B:4294967296 C:4294967297 D:4294967298

3:2G内存可以开多少个int变量, 假设一个int是32位

A: 1073741824 B:536870912 C:134217728 D:268435456

4:在一台16位机器上以下代码会输出什么

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     cout << sizeof(int*);
6     return 0;
7 }
```

A: 1 B:2 C:4 D:8

5: 深度为20的完全二叉树最少有几个点

A:524288 B:1048576 C:524289 D:1048577

6: 若某算法的计算时间表示为递推关系式:

$$T(N) = 2T(N/4) + \text{sqrt}(n)$$

则该算法的时间复杂度为:

A: $O(n * \sqrt{n})$ B: $O(n * \log n * \sqrt{n})$ C: $O(n \log n)$ D: $O(\sqrt{n} * \log(n))$

7: $(a+b)*(c+d)$ 的逆波兰表达式为

A: $abcd++*$ B: $a+bc+d*$

C: $a+bc+d$ D: $ab+cd+*$

8: 4个点的完全图一共有多少个生成树

A: 15 B: 14 C: 16 D: 13

9: 7个不同的球放入4个不同的盒子里, 盒子不能为空, 一共有多少方案

A: 8300 B: 8400 C: 8600 D: 8800

10: 一个人一开始由一块钱, 每天产生 $1/n$ 的利息, 新产生的利息又会变成本金产生利息, 随着 n 的增大, n 天后的总钱数会接近于

A:2.6666666667 B:2.3333333333 C:2.7182818285
D:2.7173643212

11: 请问以下代码是什么排序

```
1 void secret_sort(vector<int> &a) {
2     for (int i = 0; i < (int) a.size(); i++) {
3         int tmp = a[i];
4         int id = i;
5         for (int j = i - 1; j >= 0; j--) {
6             if (a[j] > tmp) {
7                 a[j+1] = a[j];
8             } else {
9                 break;
10            }
11            id = j;
12        }
13        a[id] = tmp;
14    }
15 }
```

A: 冒泡 B:插入 C:选择 D:希尔排序

12: 反转单链表的代码中空缺的地方应该填什么

A: pTemp->m_pNext = pReversedHead; B: pReversedHead->m_pNext = pTemp;

C: pCurrent->m_pNext = pTemp D: pCurrent->m_pNext = pReversedHead

```
1  ListNode * ReverseList(ListNode * pHead)
2  {
3      if(pHead == NULL || pHead->m_pNext == NULL)
4          return pHead;
5      ListNode * pReversedHead = NULL;
6      ListNode * pCurrent = pHead;
7      while(pCurrent != NULL) {
8          ListNode * pTemp = pCurrent;
9          pCurrent = pCurrent->m_pNext;
10         _____;
11         pReversedHead = pTemp;
12     }
13     return pReversedHead;
14 }
```

13:TCP/IP是下列哪个选项的缩写

A:Transmission Communication Protocol/ Internet Protocol

B:Transportation Communication Protocol/ Internet Protocol

C:Transmission Control Protocol/ Internet Protocol

D:Transportation Control Protocol/ Internet Protocol

14:有一个鸽笼, 里面有5个白鸽,小明每分钟从桶中随机取出一只涂成红色,再放回,问小明将桶中格子全部涂红的期望时间是多少

A:11.41667分钟 B:11.42667分钟 C:11.40667分钟 D:11.43667分钟

15: 5个点的二叉树一共有几种不同的形状

二, 不定项选择题 (共5题, 每题1.5分, 共计7.5分;每题有一个或多个正确选项,多选或少选均不得分)

16: 下列有关平衡二叉树的说法错误的是

A:平衡二叉树的高度始终都是保持在 $\log n$ 的,否则无法实现平衡的目的

B:C++的SET用到了AVL树

C:平衡二叉树维护平衡的方法都需要旋转节点

D:splay树的发明者是Daniel Sleator与 Robert Tarjan

17: 下面关于C/C++中指针与引用的说法正确的是

A:一个变量可以被多个变量引用

B:sizeof指针对象和引用对象的意义不一样.sizeof 引用得到的是所指向的变量的大小,而sizeof指针是对象地址的大小

C:引用可以为空

D: `int a = 10; int *b=&a;`

`int* &p=b;`

不会产生编译错误.

18:下列关于堆内存与栈内存的描述正确的是

A:一个c/c++编译的程序只有堆内存与栈内存两种,栈内存保存局部变量,堆内存保存全局变量

B:堆内存可以由程序员分配与释放

C:堆内存的堆类似于数据结构里的儿叉堆

D:代码里面的常量存储在栈内存里面

19: STL的sort使用的排序有

A:插入排序 B:选择排序 C:堆排序 D:快速排序

20: 下列关于 2019 CSP 非专业级别的能力认证说法正确的有

A:非专业级别教之专业级别更为简单,将分为两个级别进行,分别为 CSP-J(入门级), CSP-S(提高级)

B:CSP-J和CSP-S均分别举办两轮: CSP-J1, CSP-S1及CSP-J2、CSP-S2, 认证方式均为现场认证, 非网络认证。参加CSP-J/S第二轮, 必须先参加第一轮, 达到一定的分数者方可参加第二轮。

C:CCF在各省设立认证组织单位, 授权其安排认证点、监督认证过程以及处理与认证相关的事务

D:CSP非专业认证不设年龄、性别限制, 在校生和在职人员均可参加

三. 问题求解(共2题, 每题5分, 共计10分)

21: 现有50名学生都做物理、化学实验，如果物理实验做正确的有40人，化学实验做正确的有31人，两种实验都错的有4人，则两种实验都做对的有_人

22: 要排一张有5个独唱和3个合唱的节目表，如果合唱节目不能排在第一个，并且

合唱节目不能相邻，则不同排法的种数是

四：阅读程序写结果(共4题, 每题8分, 共计32分)

23:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  const int N = 110;
4  vector <int> e[N];
5  void dfs(int u) {
6      cout << u << " ";
7      for (auto v: e[u]) {
8          dfs(v);
9          cout << u << " ";
10     }
11 }
```

```

12 int main() {
13     int n, a, b;
14     cin >> n;
15     vector<int> in(n, 0);
16     for (int i = 1; i < n; i++) {
17         cin >> a >> b; e[a].push_back(b);
18         in[b]++;
19     }
20     for (int i = 0; i < n; i++) {
21         if (in[i] == 0) {
22             dfs(i);
23         }
24     }
25     return 0; }
26

```

输入

```

1 7
2 0 1
3 0 2
4 0 4
5 2 3
6 2 5
7 2 6

```

24:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int x;
5     cin >> x;
6     vector<int> p(x + 1);
7     for (int i = 1; i <= x; i++) {
8         p[i] = i;
9     }
10    for (int i = 2; i <= x; i++) {
11        if (p[i] == i) {
12            for (int j = i + i; j <= x; j += i) {
13                p[j] = min(p[j], i);
14            }
15        }
16    }
17    const int inf = (int) 1e9;
18    vector<int> best(x + 1, inf);
19    for (int i = 1; i <= x; i++) {
20        int tmp = i;
21        while (tmp > 1) {
22            int d = p[tmp];
23            if (i != d) {
24                best[i] = min(best[i], i - d + 1);
25            }
26            tmp /= p[tmp];
27        }
28    }
29    int res = inf;
30    {
31        int tmp = x;
```

```

32         while (tmp > 1) {
33             int d = p[tmp];
34             for (int j = x - d + 1; j <= x; j++) {
35                 res = min(res, best[j]);
36             }
37             tmp /= p[tmp];
38         }
39     }
40     cout << res << '\n';
41     return 0;
42 }
43

```

输入: 20

25:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int n;
5      long long k;
6      cin >> n >> k;
7      if (k > (long long) n * (n - 1) / 2) {
8          puts("Impossible");
9          return 0;
10     }
11     k--;

```

```

12     int cnt = 1;
13     while (k >= (long long) cnt * (cnt - 1) / 2) {
14         cnt++;
15     }
16     int extra = (long long) cnt * (cnt - 1) / 2 -
k;
17     for (int i = 0; i < cnt - extra; i++) {
18         putchar('(');
19     }
20     printf("()");
21     for (int i = cnt - extra; i < cnt - 1; i++) {
22         putchar('(');
23     }
24     for (int i = 0; i < cnt - 1; i++) {
25         putchar(')');
26     }
27     for (int i = 0; i < n - cnt; i++) {
28         printf("()");
29     }
30     printf("\n");
31     return 0;
32 }
33

```

输入: 5 8

26:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int t, x, y, p, q;
5     scanf("%d", &t);
6     while(t--) {
7         scanf("%d%d%d%d", &x, &y, &p, &q);
8         int l = 1, r = (int)1e9;
9         int best = -1;
10        while(l <= r) {
11            int m = (l + r) >> 1;
12            if (1LL * m * (q - p) >= y - x && 1LL
13                * p * m - x >= 0 && 1LL * q * m - y >= 0) {
14                best = m; r = m - 1;
15            } else {
16                l = m + 1;
17            }
18            // cout << best << endl;
19            if(best == -1)
20                cout << best << " ";
21            else cout << 1LL*best*q-y<<" ";
22        }
23        return 0;
24    }
25

```

输入

```
1  4
2  3 10 1 2
3  7 14 3 8
4  20 70 2 7
5  5 6 1 1
```

五. 完善程序(共2题, 每题14分, 共计28分)

27: 一个元素各不相同有序的数组被循环移动了0或者多个位置,现在你需要在这个新的数组中寻找一个数,请完善程序

(a,b,c,d,e的得分依次是3 2 3 3 3)

```
1 class Solution {
2     public:
3         int find(int l, int r, int target,
4 vector<int>& nums) {
5             while(l <= r) {
6                 int m = l + r >> 1;
7                 if(nums[m] == target) {
8                     return m;
9                 }
10                if(__a__) {
11                    l = m + 1;
12                } else {
13                    r = m - 1;
14                }
15            }
16            return -1;
17        }
18    };
19 }
```

```

13         }
14     }
15     return -1;
16 }
17 int search(vector<int>& nums, int target)
18 {
19     int n = (int) nums.size();
20     if(n == 0) return -1;
21     int l = 0, r = ____b____;
22     while(l <= r) {
23         int m = l + r >> 1;
24         if(____c____) {
25             l = m + 1;
26         } else {
27             r = m - 1;
28         }
29     }
30     // cout << r+1 << endl;
31     if(target >= nums[0] && target <=
32     nums[r])
33         return ____d____;
34     else
35         return ____e____;
36 }
};

```

28:大整数除法 , 输入两个整数A,B, 输出A/B的结果,向下取整

数组长度均不超过1000

(a,b,c,d,e的得分依次是3 3 3 3 2)

```
1  include <bits/stdc++.h>
2  using namespace std;
3  const int L = 1210;
4  struct bint {
5      int a[L];
6      bint() {
7          for (int i = 0; i < L; i++) a[i] = 0;
8      }
9      bint(int x) {
10         for (int i = 0; i < L; i++) a[i] = 0;
11         int len = 0;
12         while(x > 0) {
13             a[len++] = x % 10; x /= 10;
14         }
15     }
16     bint(string s) {
17         for (int i = 0; i < L; i++) {
18             a[i] = 0;
19         }
20         int m = (int)s.size();
21         for (int i = 0; i < m; i++) {
22             a[i] = s[m - 1 - i] - '0';
23         }
24     }
25     bool operator ==(const bint &A) const {
26         for (int i = 0; i < L; i++) {
27             if (a[i] != A.a[i]) {
28                 return false;
29             }
```

```

30     }
31     return true; }
32 bool operator < (const bint &A) const {
33     for (int i = L - 1; i >= 0; i--) {
34         if (a[i] != A.a[i]) {
35             return a[i] < A.a[i];
36         }
37     }
38     return false;
39 }
40 bint operator - (const bint &A) const {
41     bint R;
42     int owe = 0;
43     for (int i = 0; i < L; i++) {
44         R.a[i] = a[i] - A.a[i] - owe;
45         owe = 0;
46         if (R.a[i] < 0) {
47             R.a[i] += 10;
48             owe = 1;
49         }
50     }
51     return R;
52 }
53 bint operator * (const int &x) const {
54     bint R = *this;
55     for (int i = 0; i < L; i++) {
56         R.a[i] *= x; }
57     for (int i = 0; i < L - 1; i++) {
58         R.a[i + 1] += R.a[i] / 10; R.a[i] %=
59     10;
60     }

```

```

60         R.a[L - 1] %= 10;
61         return R;
62     }
63     bint operator / (const int &x) const {
64         bint R = *this;
65         int r = 0;
66         for (int i = L - 1; i >= 0; i--) {
67             r = (r * 10 + a[i]) ;
68             R.a[i] = r / x; r %= x;
69         }
70         return R;
71     }
72     bint operator / (const bint &A) const {
73         bint P = *this, Q = A;
74         if (P < Q) {
75             return bint(0);
76         }
77         if (P == Q) {
78             return bint(1);
79         }
80         int delta = ____a____;
81         for (int i = 0; i < delta; i++) {
82             Q = Q * 10;
83         }
84         int t = 0;
85         bint R = bint();
86         for (int i = ____b____; i >= 0; i--) {
87             while (____c____) {
88                 t++; P = P - Q;
89             }
90             ____d____; t = 0; Q = Q / 10;

```

```
91         }
92         return R; }
93     int get_len() const {
94         int ret = L;
95         while(ret >= 1 && ____e____) {
96             ret--;
97         }
98         return ret;
99     }
100 void print() {
101     int m = L - 1;
102     while(m >= 1 && a[m] == 0) m--;
103     for (int i = m; i >= 0; i--)
104         printf("%d", a[i]);
105     printf("\n");
106 }
107 };
108 int main() {
109     string A, B;
110     cin >> A >> B;
111     bint a(A), b(B);
112     bint ret = a / b;
113     ret.print();
114     return 0;
115 }
116
```