

17. Игра в пьяницу

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт — проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза"). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды). Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

Формат ввода

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Формат вывода

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 10^6 ходов игра не заканчивается, программа должна вывести слово botva.

Пример 1

Ввод <input type="text"/>	Вывод <input type="text"/>
1 3 5 7 9 2 4 6 8 0	second 5

Пример 2

Ввод <input type="text"/>	Вывод <input type="text"/>
2 4 6 8 0 1 3 5 7 9	first 5

Пример 3

Ввод <input type="text"/>	Вывод <input type="text"/>
1 7 3 9 4 5 8 0 2 6	second 23

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <queue>
3
4 int main() {
5     std::queue<int> first, second;
6     int n1, n2;
7     for (int i = 0; i < 5; i++) {
8         n1 = rand() % 10;
9         n2 = rand() % 10;
10        first.push(n1);
11        second.push(n2);
12    }
13    int steps = 0;
14    while (true) {
15        int c1 = first.front(); first.pop();
16        int c2 = second.front(); second.pop();
17        if (c1 > c2) first.push(c1); first.push(c2);
18        else if (c1 < c2) second.push(c1); second.push(c2);
19        else if (c1 == c2) {
20            first.push(c1);
21            second.push(c1);
22        }
23        steps++;
24        if (steps > 1000000) {
25            cout << "botva\n";
26            return 0;
27        }
28    }
29    if (first.empty()) {
30        cout << "second\n";
31    } else if (second.empty()) {
32        cout << "first\n";
33    }
34    return 0;
35 }
```

```
6     for (int i = 0, tmp, i < 5, ++i) {
7         std::cin >> tmp, first.push(tmp);
8     }
9     for (int i = 0, tmp; i < 5; ++i) {
10        std::cin >> tmp, second.push(tmp);
11    }
12    int i = 0;
13    for (int f_c, s_c; i < 1000000 && !first.empty() && !second.empty(); ++i) {
14        f_c = first.front(), s_c = second.front();
15        first.pop();
16        second.pop();
17        if (f_c == 0 && s_c == 9) {
18            first.push(f_c);
19            first.push(s_c);
20        } else if (f_c == 9 && s_c == 0) {
21            second.push(f_c);
22            second.push(s_c);
23        } else if (f_c < s_c) {
24            second.push(f_c);
25            second.push(s_c);
26        } else {
27            first.push(f_c);
28            first.push(s_c);
29        }
30    }
31    if (first.empty()) {
32        std::cout << "second" << " " << i;
33    } else if (second.empty()) {
34        std::cout << "first" << " " << i;
35    } else {
36        std::cout << "botva";
37    }
38    return 0;
```

[Отправить](#)[Предыдущая](#)[Следующая](#)