



Вариант № 6

Оценка выполнения олимпиадной работы (заполняется проверяющим)											
№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл	0	10	85	60	25	19	-	-	-	-	68
I проверка	Фамилия И.О. проверяющего	СЫТНИК А.С.		Подпись			Σ баллов прописью	Шестидесят восемь			
	Фамилия И.О. проверяющего	АНДРЕЯНОВ Н.В.		Подпись			Σ баллов прописью	Шестидесят восемь			
II проверка	Фамилия И.О. проверяющего	Кисорев О.А.		Подпись			Σ баллов прописью	Пятьдесят девять			
	Фамилия И.О. проверяющего	Жатучьё В.В.		Подпись			Σ баллов прописью	Пятьдесят девять			

59
88

3. Запрещённые комбинации:

1. 3. 5. 7.
 2. 4. 6. 8.

- горизонтальная связь
 - вертикальная связь

Обозначим связи светофора переменными X_1, X_2, X_3, X_4 как показано на рис. 1. Переменная принимает значение истинно (1) если связь горит, иначе - значение ложно (0).

X_1	X_2	X_3	X_4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

$f(X_1, X_2, X_3, X_4) = 1$ если светофор работает корректно, иначе $f = 0$.

ЕСЛИ ~~...~~ $X_1 = 0$ и $X_2 = 0$ и $X_3 = 0 \rightarrow$

$f(X_1, X_2, X_3, X_4) = \overline{X_4} \text{ ИЛИ } (\overline{X_1} \text{ ИЛИ } \overline{X_2} \text{ ИЛИ } \overline{X_3})$

$\overline{X_1} \text{ ИЛИ } \overline{X_2} \text{ ИЛИ } \overline{X_3} = \overline{(X_1 \text{ ИЛИ } X_2 \text{ ИЛИ } X_3)}$

$(\overline{X_4}) \text{ ИЛИ } \overline{(X_1 \text{ ИЛИ } X_2 \text{ ИЛИ } X_3)} = \overline{X_4 \text{ И } (X_1 \text{ ИЛИ } X_2 \text{ ИЛИ } X_3)}$

$f(X_1, X_2, X_3, X_4) = \overline{X_4 \text{ И } (X_1 \text{ ИЛИ } X_2 \text{ ИЛИ } X_3)}$

$f(X_1, X_2, X_3, X_4) = X_4 \& (X_1 \text{ ИЛИ } X_2 \text{ ИЛИ } X_3)$, где & - лог. И, | - лог. ИЛИ

$f(X_1, X_2, X_3, X_4) = X_4 \wedge (X_1 \vee X_2 \vee X_3)$, где \wedge - лог. И, \vee - лог. ИЛИ

$f(X_1, X_2, X_3, X_4) = X_4 \wedge X_1 \vee X_4 \wedge X_2 \vee X_4 \wedge X_3$ - лог. ИЕ

1) Зелёная: $(9^2 - 6 \cdot 3) / 6 = 10,5 = 11$
 Красная: $44 + 15189 = 44$
 Жёлтая: $33 / 42 \wedge 2 = 0$

~~Задача 6.2~~
~~В отделе КРАСНОЙ~~
~~ВВС = произведение функций~~
~~или если f не иссяках~~



```

5. #include <vector> #include <string> #include <string.h> // БИБЛИОТЕКА ВВОДА/ВЫВОДА И Т.П
using namespace std; // АЛГОРИТМ НАПИСАН НА C++ (СТАНДАРТ > C++17)
int main() { // для хранения кода детей используем готовую строку (string)
    vector<string> list1, list2; // списки list1 - детей посещавших шахматную
    // секцию, list2 - детей посещавших теннисную
    // списки уже отсортированы для использования.
    vector<string> res; // список - ответ
    int r = 0; // указательная для хранения правых границы или пос-
    // смотренного префикса списка list2.
    int n = list1.size(); // размер списка list1
    int m = list2.size(); // размер списка list2
    for (int i = 0; i < n; ++i) { // просматриваем все элементы list1
        while (r < m && list1[i] >= list2[r]) ++r; // просматриваем элемент меньше
        // чем list1[i]
        if (r < m && !list1[i].compare(list2[r])) ++r; // добавлен
        // если в list2 нет list1[i]
        res.push_back(list1[i]); // добавлен
    }
    for (auto & elem : res) cout << elem << " "; // выводим ответ на задачу.
    return 0;
}
    
```

Данный алгоритм работает за $O(n+m)$, где n - размер списка детей - шахматистов, m - размер списка детей - теннисистов, x - длина строки, списки в порядке номер ребенка (считаем, что все номера имеют равную длину, иначе заполнили нули). Перед запуском алгоритма короткая строка лидирует, чтобы все номера имели 1 цифру.

Функция которой можно описать сравнение строк одинаковой длины

```

bool check_equal(string& a, string& b) {
    bool res = 1;
    for (int i = 0; i < a.size(); ++i) res = res & (a[i] == b[i]);
    return res; // 1 если a равно b, иначе 0
}
    
```

```

bool check_smaller(string& a, string& b) {
    for (int i = 0; i < a.size(); ++i) {
        if (a[i] < b[i]) return 1;
        if (a[i] > b[i]) return 0;
    }
    return 0;
}
    
```

```

bool check_bigger(string& a, string& b) {
    for (int i = 0; i < a.size(); ++i) {
        if (a[i] < b[i]) return 0;
        if (a[i] > b[i]) return 1;
    }
    return 0;
}
    
```

1 если a больше b, иначе 0.

Пример: list1 = {"a1", "a2", "b9"}; list2 = {"a1", "b9", "d5"};
 Ответ: res = {"a2"}.

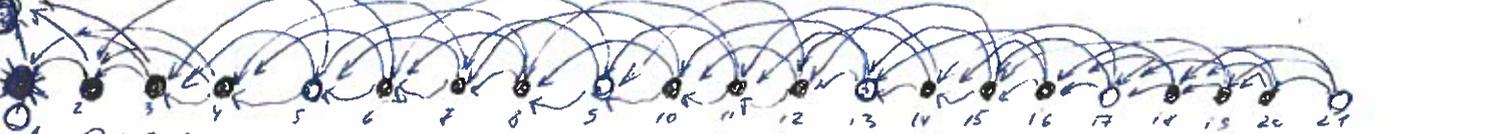
Данные функции можно использовать вместо знаков =, >, < для сравнения строк.
 В коде алгоритма используем знаки ==, что эквивалентно check_equal, >, что эквивалентно check_bigger.



6. Обозначим ширину и длину участка за n, m . В примере $n = 36, m = 14$. Разобьем участок на равные квадраты, из которого потом вылишем окружность, образованное вокруг колбика. Ответ на задачу кол-во квадратов. Очевидно, сторона квадрата = $\text{gcd}(n, m)$ - наибольший общий делитель \Rightarrow ответ на задачу $(n / \text{gcd}(n, m)) * (m / \text{gcd}(n, m))$

```
#include <bits/stdc++.h>
using namespace std;
int gcd(int x, int y) { return (y ? gcd(y, x % y) : x); } // gcd, ИД алгоритм Евклида
int main {
    int n, m; cin >> n >> m; int x = gcd(n, m);
    cout << (n/x) * (m/x);
    return 0;
}
```

2. Представим игру в виде ориентированного графа, такого что в нем существует ребро из u в v если из положения u существует переход в v (т.е. можно из u клинкой вынести 1, 2, 3 камня так что получить v) Иначе мы считаем ситуацию выигрышной если из нее $u = 0$, или существует переход в проигрышную ситуацию. Получим:



- 1 - позиции в которых независимо от действий первого игрока второй игрок выигрывает
- 2 - позиции в которых при оптимальной игре первого игрока второй игрок проигрывает

Алгоритм: Если первый игрок взял 1 или 3 камня с кучки, то взять 1 камень, иначе взять 2 камня. (Алгоритм получен из графа игры) Алгоритм работает при кол-ве камней с кучки $n \geq 5$, $n = 1$, $n \bmod 2 = 1$

У.р.	Адрес	В. размер	Размер	Д. масса	Д. адресов	Ш. адрес
Г.к.		500	25	255 255 255 1	152.168.70.1	152.168.70.1
И.к.		500	125	255 255 255 ..	152.168.70.1	-
УЦ 2	152.168.71.0	500	25	255 255 255 127	152.168.71.127	192.168.71.127
Б.		500	125	255 255 255 191	152.168.71.191	152.168.71.191
.О.		500	125	255 255 255 224	152.168.71.224	155.2+152.168.71.224
Т.ст.		500	125	255 255 255 254	152.168.71.254	152.168.71.254

6. Если $n = 36, m = 14 \Rightarrow 126 = 36 / 2 * 14 / 2 = 126$

2. Оптимальность показана стратегией доказательства тем, что если не считать кол-во камней с кучки на сторону игроку, следует не менять четность кол-ва в кучке с кучкой, т.к. выигрышные позиции для него - это четные числа \Rightarrow кол-во камней, которая была соперник - x , кол-во камней, которые взял игрок - y должно $(x+y) \bmod 2 = 0 \Rightarrow$ если x - четн., то y - четн. (2 и 2), если x - нечетн., то y - нечетн. (1 и 1, или 3 и 1)

