

Всероссийская олимпиада школьников по информатике  
Вологодская область  
II (муниципальный) этап  
2019-2020 учебный год  
7 - 8 классы

**Методические рекомендации по разбору задач**

**Задача 1. Шифровальщик (100 баллов)**

Ответы первые три вопроса можно получить, просто вручную применив алгоритм в прямую или обратную сторону. Ответы: *ctubsu*, *flower*, *bee*.

Для ответа на четвёртый вопрос заметим, что, начиная с третьего применения алгоритма, первые два символа будут чередоваться – ‘ce’, ‘df’, ‘ce’, ‘df’ и так далее. Так как 100 – это чётное число, то ответом будет *df*.

В пятом вопросе сразу заметим, что первый символ ответа равен ‘d’ (смотрите предыдущий пункт). А последний символ – это буква ‘t’, циклически сдвинутая по алфавиту на миллион раз. Но это – то же самое, что сдвиг на  $(1000000 \bmod 26) = 14$  раз, и это буква ‘h’. Ответом будет *dh*.

**Задача 2. Кораблик (100 баллов)**

Ответы на первые два вопроса легко получить прогонкой алгоритма вручную – должно получиться **7** и **E3**.

Ответы на остальные вопросы, в принципе, тоже можно найти вручную. Однако, при этом нужно быть аккуратным, чтобы не ошибиться в подсчётах.

Заметим, что для участников, умеющих работать с матрицами, может быть проще и быстрее занести схему лабиринта в файл и написать программы для вычисления ответов. Ответы: **17**, **18**, **3**.

**Задача 3. Поездка на олимпиаду (100 баллов)**

Данную задачу можно решить, используя лишь условный оператор и некоторые вычисления по формулам. Пример такого решения на языке C++:

```
#include <cstdio>  
#include <algorithm>
```

```

int main() {
    int first, last;
    scanf("%d %d", &first, &last);
    if (last <= 36) {
        printf("%d", (last + 3) / 4 - (first + 3) / 4 + 1);
    } else if (first >= 37) {
        printf("%d", (last + 1) / 2 - (first + 1) / 2 + 1);
    } else {
        printf("%d", 9 - std::min((first + 3) / 4,
            10 - (last - 35) / 2) + 1);
    }
}

```

Ещё одно решение использует циклы: переберём все отсеки, в каждом отсеке переберём все места. Если хотя бы одно место в отсеке попадает в интервал от  $X$  до  $Y$ , увеличиваем ответ:

```

#include <iostream>

int main() {
    int x, y, ans = 0;
    std::cin >> x >> y;
    int u = 1, b = 54;
    for (int i = 0; i < 9; i++) {
        int match = 0;
        for (int j = 0; j < 4; j++) {
            if (u >= x && u <= y) match = 1;
            u++;
        }
        for (int j = 0; j < 2; j++) {
            if (b >= x && b <= y) match = 1;
            b--;
        }
        ans += match;
    }
    std::cout << ans;
}

```

#### Задача 4. Библиотеки (100 баллов)

Вначале определим максимальное количество школьников, посещающих все библиотеки. Пусть оно равно  $d_{\max}$ . Уберем из дальнейшего рассмотрения этих  $d_{\max}$  школьников, тогда у нас осталось  $(N - d_{\max})$ , из которых  $(A - d_{\max})$  ходят в школьную библиотеку,  $(B - d_{\max})$  – в детскую,  $(C - d_{\max})$  – в районную. Из них никто не ходит во все три сразу. Чтобы в каждую библиотеку мог ходить хотя бы один из них, надо, чтобы:

$$A - d_{\max} + B - d_{\max} + C - d_{\max} \geq N - d_{\max}$$

Отсюда:

$$d_{\max} \leq (A + B + C - N) / 2$$

Итак,  $d_{\max} = \max(0, (A+B+C-N) \text{ div } 2)$

Теперь определим минимальное количество школьников, посещающих все библиотеки. Пусть оно равно  $d_{\min}$ . Уберем из дальнейшего рассмотрения этих  $d_{\min}$  школьников, тогда у нас осталось  $(N-d_{\min})$ , из которых  $(A-d_{\min})$  ходят в школьную библиотеку,  $(B-d_{\min})$  – в детскую,  $(C-d_{\min})$  – в районную. Из них никто не ходит во все три сразу. Для удобства введём обозначения:  $n=N-d_{\min}$ ,  $a=A-d_{\min}$ ,  $b=B-d_{\min}$ ,  $c=C-d_{\min}$ .

Вначале посмотрим, когда пересечение школьной и детской библиотек будет как можно меньше. Минимальный размер пересечения равен  $a+b-n$ , а максимальный размер непересекающихся частей тогда равен  $n-(a+b-n) = 2n-a-b$ .

Теперь нам нужно, чтобы те, кто ходит в районную библиотеку, пересекались лишь с этими  $2n-a-b$  школьниками, то есть  $c \leq 2n-a-b$ . Немного преобразуем:

$$c+a+b \leq 2n$$

$$C-d_{\min}+A-d_{\min}+B-d_{\min} \leq 2N-2d_{\min}$$

$$d_{\min} \geq A+B+C-2N$$

Итак,  $d_{\min} = \max(0, A+B+C-2N)$

**Пример решения на C++:**

```
#include<iostream>
#include<algorithm>

int main() {
    long long n, a, b, c;
    std::cin >> n >> a >> b >> c;
    std::cout << std::max(0LL, c + a + b - 2 * n) << " "
    << std::min((a + b + c - n) / 2, std::min(a, std::min(b, c)));
}
```

Для решения первой подзадачи можно не выводить конечные формулы, а, например, перебирать значение  $d$  в цикле и каждый раз проверять, существует ли подходящее разбиение школьников.

### Задача 5. Гости (100 баллов)

Для решения первой подзадачи можно перебрать тремя вложенными циклами все тройки гостей и проверить их время на отсутствие пересечений.

Полное решение выглядит так. Найдём гостя, который ушёл раньше всех, а также гостя, который пришёл позже всех. Теперь

осталось найти третьего гостя, который пришёл позже первого, а ушёл раньше второго. Если такого гостя не найдётся, то ответ -1.

### Пример решения на C++:

```
#include <stdio.h>
#include <vector>

int main() {
    int n;
    scanf("%d", &n);
    std::vector<std::pair<int, int> > t(n);
    for (int i = 0; i < n; i++) {
        int h1, m1, s1, h2, m2, s2;
        scanf("%d:%d:%d %d:%d:%d", &h1, &m1, &s1, &h2, &m2, &s2);
        t[i] = {h1 * 3600 + m1 * 60 + s1, h2 * 3600 + m2 * 60 + s2};
    }
    int left = 0, right = 0;
    for (int i = 0; i < n; i++) {
        if (t[i].second < t[left].second)
            left = i;
        if (t[i].first > t[right].first)
            right = i;
    }
    for (int i = 0; i < n; i++)
        if (t[i].first > t[left].second && t[i].second <
            t[right].first) {
            printf("%d %d %d", left + 1, i + 1, right + 1);
            return 0;
        }
    puts("-1");
}
```

### Информация о тестах для задач 7-8 классов

Задача	Тестов из условия	Основных тестов	Баллов за один основной тест	Проверяющая программа
1. Шифровальщик	-	5	20	<b>check.exe</b>
2. Кораблик	-	5	20	<b>check.exe</b>
3. Поездка на олимпиаду	1	10	10	<b>check.exe</b>
4. Библиотеки	1	20	5	<b>check.exe</b>
5. Гости	2	20	5	<b>check.exe</b>

Информация об использовании системы автоматической проверки решений приведена в Требованиях к организации и проведению муниципального этапа.