

Всероссийская олимпиада школьников по информатике  
Вологодская область  
II (муниципальный) этап  
2017-2018 учебный год  
7 - 8 классы

**Методические рекомендации по разбору задач**

**Задача 1. Числовая последовательность (100 баллов)**

Несложно заметить, что искомый номер можно найти как сумму арифметической прогрессии, и ответом будет формула

$$N * (N + 1) / 2$$

Важно то, что операция деления стоит именно в конце, так как деление целочисленное. Формулы вида  $N / 2 * (N + 1)$  или  $(N + 1) / 2 * N$  будут давать неверные ответы при нечётном  $N$  или нечётном  $(N+1)$  соответственно.

**Задача 2. Черепашка (100 баллов)**

Ответы на первые 4 вопроса несложно получить, просто взяв ручку и бумагу и повторив действия исполнителя Черепашка. В пятом вопросе также несложно сообразить, что после поворота на 90 градусов нужно повернуть ещё на 30, то есть всего повернуть на 120 градусов. Решение:

4 60 3 2 120

**Задача 3. Игра (100 баллов)**

Так как надо найти наибольший выигрыш, значит, хитрая Баба-Яга должна взять карточки с самыми большими числами, а Змею Горынычу достанутся все остальные. Для решения этой задачи можно воспользоваться одним из алгоритмов сортировки. После того как числа в массиве станут упорядочены по убыванию, достаточно из суммы чисел, расположенных в первой половине массива, вычесть сумму чисел, расположенных во второй половине.

Чтобы решить эту задачу на полный балл, необходимо использовать эффективный алгоритм сортировки, работающий быстрее, чем за квадрат. Простейший вариант - использовать готовую функцию сортировки из стандартной библиотеки языка программирования.

**Пример решения на C++:**

```

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

int main() {
    int n;
    scanf("%d", &n);
    std::vector<int> a(n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    std::sort(a.begin(), a.end());
    int sum1 = 0, sum2 = 0;
    for (int i = 0; i < n / 2; i++)
        sum1 += a[i];
    for (int i = n / 2; i < n; i++)
        sum2 += a[i];
    printf("%d\n", sum2 - sum1);
}

```

**Задача 4. Отношение углов (100 баллов)**

Внутренние углы треугольника относятся как  $x : y : z$ , то есть  $x \cdot k + y \cdot k + z \cdot k = 180$  (где  $k$  - некий коэффициент). Выразим коэффициент  $k$  - он равен  $180/(x+y+z)$ . Тогда внутренние углы равны  $180 \cdot x/(x+y+z)$ ,  $180 \cdot y/(x+y+z)$ ,  $180 \cdot z/(x+y+z)$ , а внешние углы равны

$$180 - 180 \cdot x/(x+y+z) = 180 \cdot (y+z)/(x+y+z)$$

$$180 - 180 \cdot y/(x+y+z) = 180 \cdot (x+z)/(x+y+z)$$

$$180 - 180 \cdot z/(x+y+z) = 180 \cdot (x+y)/(x+y+z)$$

Отношение внешних углов (после сокращения) получается таким:  $(y + z) : (x + z) : (x + y)$ .

Разделим эти числа на наибольший общий делитель (его можно найти алгоритмом Евклида) и отсортируем по возрастанию.

**Пример решения на C++:**

```

#include <bits/stdc++.h>

int gcd(int a, int b) {
    while (b > 0) {
        int c = a % b;
        a = b;
        b = c;
    }
}

```

```

    return a;
}

int main() {
    int a, b, c;
    scanf("%d : %d : %d", &a, &b, &c);
    int x[3] = {b + c, a + c, a + b};
    int g = gcd(x[0], gcd(x[1], x[2]));
    x[0] /= g; x[1] /= g; x[2] /= g;
    std::sort(x, x + 3);
    printf("%d:%d:%d\n", x[0], x[1], x[2]);
}

```

### Задача 5. Экономный садовод (100 баллов)

Простейший способ решения данной задачи - использование алгоритма Флойда-Уоршалла. Некоторую сложность может вызвать то, что в задаче имеются два критерия - стоимость маршрута и количество пересадок. На самом деле, их можно рассматривать как один составной критерий: в первую очередь сравниваются стоимости, а если они одинаковы, то число пересадок.

Вычислительная сложность алгоритма составляет  $O(N^3)$ .

#### Пример решения на C++:

```

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

const int INF = 1000000000;

int main() {
    int n;
    scanf("%d", &n);
    std::vector<std::vector<int>> > a(n, std::vector<int>(n));
    std::vector<std::vector<int>> > c(n, std::vector<int>(n, 1));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
            if (a[i][j] == 0) {
                a[i][j] = INF;
            }
        }
    }
    for (int k = 0; k < n; k++) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (a[i][k] + a[k][j] < a[i][j]) {
                    a[i][j] = a[i][k] + a[k][j];
                    c[i][j] = c[i][k] + c[k][j];
                } else if (a[i][k] + a[k][j] == a[i][j])

```

```

        && c[i][k] + c[k][j] < c[i][j]) {
            c[i][j] = c[i][k] + c[k][j];
        }
    }
}
printf("%d %d", a[0][n - 1], c[0][n - 1] - 1);
}

```

В материалах олимпиады можно найти также примеры и других решений задач на языках C++, Java, Pascal.

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

| Задача                         | Тестов из условия | Основных тестов | Баллов за один основной тест | Проверяющая программа |
|--------------------------------|-------------------|-----------------|------------------------------|-----------------------|
| 1. Числовая последовательность | -                 | 10              | 10                           | <b>check.exe</b>      |
| 2. Черепашка                   | -                 | 5               | 20                           | <b>check.exe</b>      |
| 3. Игра                        | 1                 | 20              | 5                            | <b>check.exe</b>      |
| 4. Отношение углов             | 1                 | 10              | 10                           | <b>check.exe</b>      |
| 5. Экономный садовод           | 1                 | 10              | 10                           | <b>check.exe</b>      |

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