

## Задача 27 (алгоритмы обработки последовательностей от простого к сложному)

Начнём с самых простых задач.

Первая часть условия всех задач, рассмотренных далее, одинакова: “С клавиатуры вводится натуральное число  $N$  ( $1 \leq N \leq 10\,000$ ), а затем в следующих  $N$  строках –  $N$  целых чисел, по одному в каждой строке.”

Конкретное содержание задачи уточняется в каждом отдельном случае.

Решения будут приводиться на языке C++. Шаблон типовой программы на языке C++, выглядит так:

```
#include <iostream>
using namespace std;
int main()
{
    int ...; //объявление переменных
    cin >> n;
    // инициализация переменных
    for (int i = 1; i <= n; i++)
    {
        cin >> x;
        // обработка значения x
    }
    cout << ... << endl; // вывод результата
}
```

**Задача 1.** Требуется вывести одно неотрицательное число – количество пар с чётной суммой, образованных различными элементами последовательности.

**Решение.**

Чтобы не учитывать одну и ту же пару дважды, будем рассматривать только упорядоченные пары  $(a_i, a_j)$ , для которых  $i < j$ .

В данной задаче очередное число, прочитанное из входного потока, может образовывать пары с *любым* из предыдущих элементов последовательности. Будем говорить, что все предыдущие элементы принадлежат *множеству выбора пары* для следующего элемента.

*Подходящими* будем называть пары, которые удовлетворяют условиям задачи. В этой задаче подходящими будут те пары, для которых сумма элементов чётна.

Обозначим через  $Q_k$  количество подходящих пар для подпоследовательности, состоящей из первых  $k$  элементов полной последовательности. Если очередное полученное число  $x$  чётно, оно образует подходящие пары (с чётной суммой) только с предыдущими *чётными* элементами последовательности, а нечётное значение  $x$  – только с предыдущими *нечётными* элементами. Поэтому нам нужно хранить на каждой итерации количество предыдущих чётных чисел  $E_{k-1}$  и количество предыдущих нечётных чисел  $O_{k-1}$ .

В блоке инициализации присваиваем переменным  $Q$ ,  $O$ ,  $E$  нулевые значения. После ввода очередного значения  $x$  из входного потока нужно обновить  $Q$ , а затем обновить  $E$  или  $O$ , в зависимости от чётности  $x$ :

```
#include <iostream>
using namespace std;
int main()
{
    int n, q, e, o, x; //объявление переменных
    cin >> n;
    q = 0;
    e = 0;
    o = 0; // инициализация переменных
    for (int i = 1; i <= n; i++)
    {
        cin >> x;
        if (x % 2 == 0)
```

```

    {
        q += e;
        e += 1;
    }
    else
    {
        q += o;
        o += 1;
    }
}
cout << q << endl;
}

```

**Задача 2.** Требуется вывести одно неотрицательное число – количество пар с суммой, равной  $S = 20$ , образованных различными элементами последовательности.

**Решение.**

Применим общий подход: получив из входного потока новое значение  $x$ , определим, сколько подходящих пар оно может образовать с предыдущими элементами последовательности. Обозначим через  $Q_k$  количество подходящих пар в подпоследовательности, состоящей из первых  $k$  чисел полной последовательности.

Напомним, что по условию (см. начало этого раздела) все элементы последовательности целые и положительные. Поэтому числа, большие или равные  $S$ , не могут образовать ни одной подходящей пары.

Если число  $x$  меньше  $S$ , оно образует подходящие пары со всеми предыдущими элементами, равными  $S - x$ . Следовательно, нам нужно хранить массив значений  $D$ , где  $D[i]$  – это количество предыдущих элементов последовательности, равных  $i$  ( $1 \leq i < S$ ).

На этапе инициализации значение переменной  $Q$  и все элементы массива  $D$  обнуляются. Они обновляются на очередной итерации только тогда, когда из входного потока получено число  $x$ , которое меньше  $S$ :

```

#include <iostream>
using namespace std;
int main()
{
    int n, s=20, q, x;
    int d[s]; //объявление переменных
    cin >> n;
    q = 0;
    for (int i = 0; i < n; i++)
        d[i] = 0; // инициализация переменных
    for (int i = 1; i <= n; i++)
    {
        cin >> x;
        if (x < s)
        {
            q += d[s - x];
            d[x] += 1;
        }
    }
    cout << q << endl;
}

```

**Задача 3.** Требуется вывести одно неотрицательное число – количество пар с суммой, меньшей или равной  $S = 20$ , образованных различными элементами последовательности.

**Задача 4.** Требуется вывести одно неотрицательное число – количество пар с суммой, большей, чем  $S = 20$ , образованных различными элементами последовательности.

**Задача 5.** Рассматриваются все пары различных элементов последовательности, в которых хотя бы одно число кратно  $F = 17$ . Из всех таких пар нужно выбрать пару с максимальной суммой и вывести на экран оба элемента, образующие эту пару. Если таких пар несколько, то нужно вывести любую из них. Если таких пар нет, то нужно вывести два нуля.

**Задача 6. (Статград апрель)** Рассматриваются все пары различных элементов последовательности, разность которых делится на  $R = 60$  и при этом хотя бы один элемент из пары больше  $B = 80$ . Требуется вывести одно неотрицательное число – количество пар, удовлетворяющих указанным условиям.

**Задача 7. (досрочный вариант ЕГЭ-2020)** Рассматриваются все пары различных элементов последовательности, разность которых чётна и при этом хотя бы одно число из пары кратно  $F = 17$ . Из всех таких пар нужно выбрать пару с максимальной суммой и вывести на экран оба элемента, образующие эту пару. Если таких пар несколько, то нужно вывести любую из них. Если таких пар нет, то нужно вывести два нуля.

**Задача 8.** Рассматриваются все пары различных элементов последовательности, произведение которых кратно  $F = 39$ . Из всех таких пар нужно выбрать пару с максимальной суммой и вывести на экран оба элемента, образующие эту пару. Если таких пар несколько, то нужно вывести любую из них. Если таких пар нет, то нужно вывести два нуля.

**Задача 9.** Рассматриваются все пары различных элементов последовательности, номера которых в последовательности отличаются не менее, чем на  $L = 4$ , а произведение кратно  $F = 39$ . Из всех таких пар нужно выбрать пару с максимальной суммой и вывести на экран оба элемента, образующие эту пару. Если таких пар несколько, то нужно вывести любую из них. Если таких пар нет, то нужно вывести два нуля. Предполагается, что  $N > L$ .