

МУНИЦИПАЛЬНЫЙ ЭТАП ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ В 2022-2023 УЧЕБНОМ ГОДУ ДЛЯ УЧАЩИХСЯ 9-11 КЛАССОВ

Вариант 1

Время проведения - 235 минут (3 часа 55 минут)

Максимальное количество баллов за задачу – 100

Максимальное количество баллов – 400

Особенности проведения - задания практического тура выполняются на компьютерах и сохраняются в отведенные папки. Не допускается использование локальной сети. Для проверки программ используется автоматическая тестирующая система Яндекс.Контест. Интернет-фильтр должен быть настроен только на использование соответствующих адресов и портов. Для авторизации должны быть подготовлены пароли и логины.

Проверяющая система:

<https://official.contest.yandex.ru/contest/XXXX/enter>
(здесь XXXX - номер контеста).

Ограничение по времени – 1 секунда

Ограничение по памяти – 64Mb

Ввод данных – через стандартный поток ввода или из файла input.txt

Вывод данных – через стандартный поток вывода или в файл output.txt

Критерии оценивания: за каждый пройденный тест к задаче начисляется балл, который вычисляется по формуле: $100 / \text{количество тестов}$

№	1	2	3	4
количество тестов	8	10	10	8
балл за тест	12	10	10	12

1. Полевые работы

«Лютики и разные другие цветочки... А еще картофель и кабачки. Опять же и пчелы...» - такие мысли кружились у главы сельской администрации Егора Ильича, покуда он разбирался с делами и заботами села Степанчиково.

И, конечно, самая важная задача на повестке дня – разобраться с участками сельчан, вокруг которых было множество споров и пересудов. А именно, Егору Ильичу срочно требовалось узнать размеры самого маленького и самого большого участка, а также суммарную площадь всех участков.

И повезло же ему, право, с этой задачей – ведь все участки в этом селе были прямоугольной формы!

Входные данные

На первой строке входного файла записано 1 целое число N , $1 \leq N \leq 10000$. Каждая из N следующих строк содержит два целых числа в диапазоне от 1 до 1000, разделенных пробелом – длину и ширину очередного участка.

Выходные данные

Необходимо вывести через пробел 3 числа: суммарную площадь участков, площади самого большого и самого маленького участков.

Пример

Входные данные	Выходные данные
3 10 10 20 20 5 3	515 400 15

Решение:

```
#include <fstream>
using namespace std;

int main()
{
    ifstream in("input.txt");
    int N, // count
        a, b; // length and width
    in >> N;

    long long S,
        Smax = 0,
        Smin = 1e+7,
        Stotal = 0;

    for (int i = 0; i < N; ++i)
    {
        in >> a >> b;
        S = (long long)a * (long long)b;

        Stotal += S;
        if (S < Smin) Smin = S;
        if (S > Smax) Smax = S;
    }
    in.close();

    ofstream out("output.txt");
    out << Stotal << " " << Smax << " " << Smin << endl;
    out.close();
    return 0;
}
```

2. Как назвать магазин

Прасковья Ильинична заведует в селе Степанчиково замечательным магазином, где есть и продукты, и различные нужные в быту вещи, и даже детские игрушки. И страсть как хочется Прасковье Ильиничне придумать для магазина красивое и даже мелодичное название, чтобы каждый, прочитавший его, испытал душевную радость.

Прасковья Ильинична приготовила два списка слов: один со словами, которые она считает «красивыми», другой со словами «о природе». Название должно состоять из двух разных слов, и наша творческая заведующая хочет, чтобы, во-первых, оба эти слова были выбраны только из приготовленных списков, а во-вторых, чтобы среди этих двух слов было и «красивое» и «о природе» (возможно это одно и то же слово, возможно разные – в любом порядке).

Помогите определить, удовлетворяет ли возможное название этим строгим правилам.

Входные данные

На первой строке записаны два целых числа n и m в диапазоне от 1 до 20, разделенные пробелом, – количество слов в каждом из приготовленных списков.

На следующих n строках располагаются по одному в строке слова из первого списка, далее на m строках располагаются по одному в строке слова из второго списка.

Длина каждого слова не превышает 100 символов. Слова состоят из строчных латинских букв.

На последней строке написаны два слова, разделенные пробелом – возможное название магазина.

Длина последней строки не превышает 200 символов

Выходные данные

Вывести знак «+» без кавычек, если название подходит, вывести знак «-» без кавычек, если название не подходит.

Примеры

Входные данные	Выходные данные
2 3 beautiful fly sun honeybee fly beautiful sun	+
2 3 beautiful fly sun honeybee fly	+

honeybee fly	
1 1 beautiful sun beautiful house	-

Решение:

```

fin = open('input.txt', 'r')
lines = fin.read().splitlines()
fin.close()

n, m = [int(x) for x in lines[0].split()]
name = lines[-1].split()
sn = set(lines[1:n+1])
sm = set(lines[n+1:-1])

s = sn | sm

answer = '-'
if (
    ( name[0] in s ) and ( name[1] in s ) and
    ( (name[0] in sn) or (name[1] in sn) ) and ( (name[0] in sm)
or (name[1] in sm) )
):
    answer = '+'

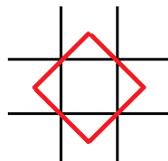
fout = open('output.txt', 'w');
fout.write(answer + '\n')
fout.close()

```

3. Веселый комбайн

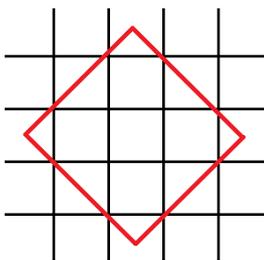
Сельский комбайнер дядя Паша большой знаток своего дела, а кроме того он человек тонкой душевной организации. Когда дядя Паша управляет комбайном, то всегда старается вести его так чтобы на поле образовывались красивые узоры.

Все квадратное поле в воображении веселого комбайнера разбито на маленькие квадратики, где он и рисует свои узоры. А самая любимая фигура у нашего комбайнера – «цветочек». На рисунке показан цветочек размера 2, у которого каждая сторона проходит по двум воображаемым квадратикам (всего такой цветочек проходит через четыре квадратика, а пятый квадратик – в центре цветочка).



Еще дядя Паша хорошо учился в школе, поэтому ему нравится масштабировать узоры. На рисунке показан цветочек размера 3, у которого каждая сторона проходит по трем воображаемым квадратикам (всего такой

цветочек проходит через восемь квадратиков, а внутри цветочка – пять квадратиков).



А больше всего дяде Паше нравится, чтобы узоры были симметричными. Поэтому он хочет, чтобы соблюдались следующие правила:

1. Центр каждого цветочка должен лежать на одной из диагональных клеток квадратного поля.

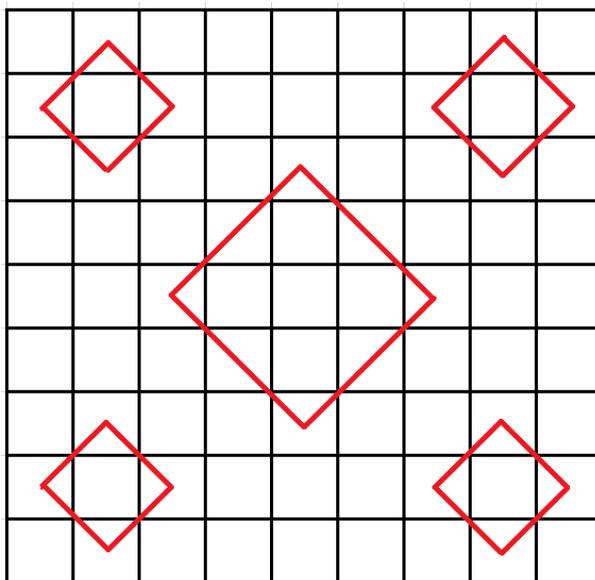
2. Весь рисунок должен быть симметричен относительно обеих диагоналей квадратного поля. Гарантируется что сторона квадратного поля состоит из нечетного числа клеток.

3. Вдоль каждого луча, идущего от каждой угловой клетки к центральной клетке квадратного поля, сначала рисуется цветочек размера 2 (как можно ближе к угловой клетке), потом цветочек размера 3 (в свою очередь как можно ближе к угловой клетке) и т.д.

4. Любые две клетки, по которым проходят стороны двух разных цветочков, не должны соприкасаться по сторонам или иметь общую вершину.

5. Цветочков на узоре должно быть как можно больше.

Например, для поля 9x9 узор будет выглядеть так:



А вот на поле размером 7x7 можно нарисовать только цветочки размером 2, т.к. цветочек размером 3 туда просто не вместится.

Входные данные

В единственной строке входного файла записано одно натуральное нечетное число в диапазоне от 3 до 99 – сторона квадрата.

Выходные данные

Необходимо вывести узор, нарисованный веселым комбайнером, в виде таблицы: 0 означает, что клетка пустая, 1 – что через эту клетку проходит рисунок узора. Разделять символы пробелами не нужно.

Примеры:

Входные данные	Выходные данные
5	00000 00100 01010 00100 00000
7	0100010 1010101 0100010 0000000 0100010 1010101 0100010
15	010000000000010 101000000000101 010010000010010 000101000101000 001000101000100 000101000101000 000010000010000 000000000000000 000010000010000 000101000101000 001000101000100 000101000101000 010010000010010 101000000000101 010000000000010

Пояснения к примеру

В первом примере на поле вмещается только один цветочек размера 2. Он располагается посередине поля, так как нужна симметрия относительно обеих диагоналей

Во втором примере симметрично можно разместить либо один узор посередине (это будет цветочек размера 2, так как нужно размещать цветочки по очереди: сначала размера 2, потом размера 3 и так далее), либо 4 цветочка размера 2 по одному в каждом углу квадратного поля. Так как нужно нарисовать как можно больше цветочков, то в итоге рисуем 4 цветочка в углах.

В третьем примере на каждом луче, идущем от каждой угловой клетки к центральной удалось разместить по два цветочка – размера 2 и затем размера 3.

Решение:

```
#include <fstream>
using namespace std;

int f[100][100];

void printFlower(int n, int m, int size)
{
    for (int i = 0; i < size-1; ++i)
    {
        f[n - (size - 1) + i][m + i] = 1;
        f[n + i][m + (size - 1) - i] = 1;
        f[n + (size - 1) - i][m - i] = 1;
        f[n - i][m - (size - 1) + i] = 1;
    }
}

int main()
{
    int N;
    fstream in("input.txt");
    in >> N;
    in.close();

    int n = 2;
    int a = 3;

    // n - max flower's size
    while (N >= a + 2 * n + 2)
    {
        ++n;
        a += 2 * n;
    }

    int size = n;
    if (N < a + 2 * n)
    {
        // center
        printFlower(N / 2, N / 2, n);
        --size;
    }

    int pos = 1;
    for (int i = 2; i <= size; ++i)
    {
        printFlower(pos, pos, i);
        printFlower(pos, N - 1 - pos, i);
        printFlower(N - 1 - pos, pos, i);
        printFlower(N - 1 - pos, N - 1 - pos, i);

        pos += i + 1;
    }
}
```

```

}

ofstream out("output.txt");
for (int i = 0; i < N; ++i)
{
    for (int j = 0; j < N; ++j)
        out << f[i][j];
    out << endl;
}
out.close();

return 0;
}

```

4. Молочная ферма

Коровы села Степанчиково приносят много молока, гораздо больше чем в соседних деревнях и селах. А все потому что молочник Федор Михайлович включает своим любимым коровам классическую музыку. В прошлом году Вивальди играл, а сейчас вот – Гендель.

Молоко нужно где-то хранить, и у Федора Михайловича есть целый склад бидонов – разной вместимости и разного веса. Стальные потяжелее, алюминиевые полегче, а самые легкие пластмассовые, бидонов каждого типа может быть один или несколько.

И, конечно, хочется, чтобы итоговая масса бидонов была как можно меньше, ведь их потом надо переносить. Надо помочь Федору Михайловичу выбрать нужные бидоны.

Входные данные

Первая строка содержит два целых числа S и n ($1 \leq S \leq 200$, $1 \leq n \leq 20$) – сколько литров молока нужно разлить по бидонам Федору Михайловичу, и число возможных типов бидонов соответственно.

В каждой из следующих n строк записаны через пробел три числа – вместимость очередного типа бидона (целое число в диапазоне от 1 до 100), его масса (целое число в диапазоне от 1 до 20), количество имеющихся бидонов такого типа (целое число в диапазоне от 1 до 10).

Выходные данные

Нужно вывести одно целое число – итоговую суммарную массу выбранных бидонов, в которые можно будет налить нужное количество молока. Среди всех возможных наборов бидонов, вмещающих нужное количество молока, следует выбрать тот, суммарная масса бидонов в котором – минимальна. Может оказаться так, что при этом суммарная вместимость набора больше требуемой – такая ситуация допустима (см. примеры).

Гарантируется, что решение всегда возможно.

Примеры:

Входные данные	Выходные данные
10 2 15 5 2 5 3 2	5

10 3 15 5 2 5 3 2 6 1 1	4
----------------------------------	---

Решение:

```
#include <fstream>
using namespace std;

struct can {
    int volume;
    int weight;
    int count;
};

const int MaxTypesCount = 20;
const int MaxWeight = 20;
const int MaxCansCount = 10;
const int MaxPosition = MaxTypesCount * MaxWeight *
MaxCansCount;

can cans[MaxTypesCount];
int a[MaxTypesCount][MaxPosition+1];

int main()
{
    int S, n;
    ifstream in("input.txt");
    in >> S >> n;
    for (int i = 0; i < n; ++i)
    {
        in >> cans[i].volume >> cans[i].weight >> cans[i].count;
    }
    in.close();

    // first can
    for (int w = 0; w <= MaxPosition; ++w)
    {
        int k = w / cans[0].weight;
        if (k > cans[0].count)
            k = cans[0].count;
        a[0][w] = cans[0].volume * k;
    }

    for (int num = 1; num < n; ++num)
    {
        can p = cans[num];
        for (int w = 0; w <= MaxPosition; ++w)
        {
            int maxV = 0;
            for (int k = 0; (k <= p.count) && (w - p.weight * k
>= 0); ++k)
            {
```

```

        int V = p.volume * k + a[num - 1][w - p.weight * k];
        if (V > maxV)
            maxV = V;
    }
    a[num][w] = maxV;
}

int maxVolume = 0;
int minWeight = MaxPosition + 1;

for (int num = 0; num < n; ++num)
{
    int w = 0;
    for (; (w < MaxPosition) && (a[num][w] < S); ++w);
    if (w < minWeight)
        minWeight = w;
}

ofstream out("output.txt");
out << minWeight << endl;
/*
for (int num = 0; num < n; ++num)
{
    for (int w = 0; w < 70; ++w)
    {
        out.width(4);
        out << a[num][w] << " ";
    }
    out << endl;
}
//*/
out.close();

return 0;
}

```