



# Crossproduct. Векторное произведение

Имя входного файла: `crossproduct.in`  
Имя выходного файла: `crossproduct.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны два вектора в  $\mathbb{R}^3$ . Вычислите их векторное произведение.

## Формат входных данных

Во входном файле записаны три целых числа — координаты первого вектора, затем три целых числа — координаты второго вектора.

## Формат выходных данных

Выведите три целых числа — координаты вектора, являющегося их векторным произведением.

## Пример

| <code>crossproduct.in</code> | <code>crossproduct.out</code> |
|------------------------------|-------------------------------|
| 1 0 0<br>0 1 0               | 0 0 1                         |



## Angle3d. Угол между векторами

Имя входного файла: `angle3d.in`  
Имя выходного файла: `angle3d.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны два вектора в  $\mathbb{R}^3$ . Вычислите угол между ними.

### Формат входных данных

Во входном файле записаны три целых числа — координаты первого вектора, затем три целых числа — координаты второго вектора.

### Формат выходных данных

Выведите одно действительное число — угол между данными векторами в радианах с точностью не менее 6 значащих знаков.

### Пример

| <code>angle3d.in</code> | <code>angle3d.out</code> |
|-------------------------|--------------------------|
| 1 0 0<br>0 1 0          | 1.570796327              |



# Orthogonal. Ортогональный вектор

Имя входного файла: `orthogonal.in`  
Имя выходного файла: `orthogonal.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дан ненулевой вектор. Найдите любой ненулевой целочисленный вектор, ортогональный данному.

## Формат входных данных

Во входном файле записаны три целых числа — координаты вектора  $x_0, y_0, z_0$ . Числа целые, не превосходят 1000 по модулю.

## Формат выходных данных

Выведите три целых числа — координаты ненулевого вектора, ортогонального данному.

## Пример

| <code>orthogonal.in</code> | <code>orthogonal.out</code> |
|----------------------------|-----------------------------|
| 1 1 1                      | 1 -1 0                      |



# Hands. Руки Ктулху

Имя входного файла: `hands.in`  
Имя выходного файла: `hands.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

У Ктулху на каждой руке — три пальца-отрезка, которые соединяются вместе в одной точке. Вы видите руку Ктулху, при этом конец первого пальца находится в точке  $(x_1, y_1, z_1)$ , конец второго пальца в точке  $(x_2, y_2, z_2)$ , конец третьего пальца — в точке  $(x_3, y_3, z_3)$ , а начала пальцев соединяются в точке  $(x_0, y_0, z_0)$ .

Определите, какую руку Ктулху вы видите — правую или левую.

## Формат входных данных

Программа получает на вход 12 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_0, y_0, z_0$ . Числа — целые, не превосходят 1000 по модулю. Гарантируется, что четыре данные точки не лежат в одной плоскости.

## Формат выходных данных

Выведите слово `RIGHT`, если вы видите правую руку Ктулху или слово `LEFT`, если видите левую руку.

## Пример

| <code>hands.in</code>            | <code>hands.out</code> |
|----------------------------------|------------------------|
| 1 0 0<br>0 1 0<br>0 0 1<br>0 0 0 | RIGHT                  |



# Volume. Объем тетраэдра

Имя входного файла: `volume.in`  
Имя выходного файла: `volume.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны четыре точки в  $\mathbb{R}^3$ . Определите объем тетраэдра с вершинами в этих точках.

## Формат входных данных

Программа получает на вход 12 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4$ . Числа — целые, не превосходят 1000 по модулю.

## Формат выходных данных

Выведите одно действительное число — объем тетраэдра с вершинами в данных точках, с точностью не менее  $10^{-3}$ .

## Пример

| <code>volume.in</code>           | <code>volume.out</code> |
|----------------------------------|-------------------------|
| 0 0 0<br>1 0 0<br>0 1 0<br>0 0 1 | 0.1666666667            |



## Online. Точки на прямой

Имя входного файла: `online.in`  
Имя выходного файла: `online.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны три точки в  $\mathbb{R}^3$ . Определите, лежат ли они на одной прямой.

### Формат входных данных

Программа получает на вход 9 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ . Числа — целые, не превосходят 1000 по модулю.

### Формат выходных данных

Если три данные точки лежат на одной прямой, выведите слово YES, иначе выведите слово NO.

### Пример

| <code>online.in</code>  | <code>online.out</code> |
|-------------------------|-------------------------|
| 1 0 0<br>0 1 0<br>0 0 1 | NO                      |



# Onplane. Точки на плоскости

Имя входного файла: `onplane.in`  
Имя выходного файла: `onplane.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны четыре точки в  $\mathbb{R}^3$ . Определите, лежат ли они на одной плоскости.

## Формат входных данных

Программа получает на вход 12 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4$ . Числа — целые, не превосходят 1000 по модулю.

## Формат выходных данных

Если четыре данные точки лежат на одной плоскости, выведите слово YES, иначе выведите слово NO.

## Пример

| <code>onplane.in</code>          | <code>onplane.out</code> |
|----------------------------------|--------------------------|
| 1 0 0<br>0 1 0<br>0 0 1<br>0 0 0 | NO                       |



# Distance1. Расстояние от точки до прямой

Имя входного файла: `distance1.in`  
Имя выходного файла: `distance1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана точка и прямая, заданная двумя точками, в  $\mathbb{R}^3$ . Найдите расстояние от этой точки до прямой.

## Формат входных данных

Программа получает на вход 9 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ . Числа — целые, не превосходят 1000 по модулю. Точки 2 и 3 не совпадают.

## Формат выходных данных

Выведите расстояние от точки 1 до прямой, проходящей через точки 2 и 3 с точностью не менее  $10^{-3}$ .

## Пример

| <code>distance1.in</code> | <code>distance1.out</code> |
|---------------------------|----------------------------|
| 1 0 0<br>0 0 0<br>1 1 0   | 0.7071067812               |





## Distance2. Расстояние от точки до отрезка

Имя входного файла: distance2.in  
Имя выходного файла: distance2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана точка и концы отрезка в  $\mathbb{R}^3$ . Найдите расстояние от этой точки до отрезка.

### Формат входных данных

Программа получает на вход 9 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ . Числа — целые, не превосходят 1000 по модулю. Точки 2 и 3 не совпадают.

### Формат выходных данных

Выведите расстояние от точки 1 до отрезка с концами в точках 2 и 3 с точностью не менее  $10^{-3}$ .

### Пример

| distance2.in            | distance2.out |
|-------------------------|---------------|
| 0 0 0<br>1 1 1<br>2 2 2 | 1.7320508076  |



# Plane1. Уравнение плоскости по трем точкам

Имя входного файла: `plane1.in`  
Имя выходного файла: `plane1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Постройте уравнение плоскости, проходящей через три заданные точки.

## Формат входных данных

Программа получает на вход 9 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ . Числа — целые, не превосходят 1000 по модулю, точки не лежат на одной прямой.

## Формат выходных данных

Выведите четыре целых числа  $A, B, C, D$  — коэффициенты уравнения плоскости  $Ax + By + Cz + D = 0$ , проходящей через заданные точки.

## Пример

| <code>plane1.in</code>  | <code>plane1.out</code> |
|-------------------------|-------------------------|
| 0 0 0<br>0 0 1<br>0 1 0 | 1 0 0 0                 |



## Plane2. Точки на плоскости

Имя входного файла: plane2.in  
Имя выходного файла: plane2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

По данному уравнению плоскости  $Ax + By + Cz + D = 0$  найдите три точки, не лежащие на одной прямой, через которые проходит данная плоскость.

### Формат входных данных

Программа получает на вход 4 целых числа — коэффициенты  $A, B, C, D$  уравнения плоскости. Числа — целые, не превосходят 1000 по модулю.

### Формат выходных данных

Программа должна вывести 9 действительных чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$  — координаты трех точек, задающих данную плоскость с точностью, не менее 6 значащих цифр.

### Пример

| plane2.in | plane2.out              |
|-----------|-------------------------|
| 1 1 1 -1  | 1 0 0<br>0 1 0<br>0 0 1 |



# Perpendicular1. Основание перпендикуляра

Имя входного файла: perpendicular1.in  
Имя выходного файла: perpendicular1.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана точка и плоскость. Найдите основание перпендикуляра, опущенного из данной точки на плоскость.

## Формат входных данных

Программа получает на вход 3 целых числа  $x_0, y_0, z_0$  — координаты точки и 4 целых числа — коэффициенты  $A, B, C, D$  уравнения плоскости. Числа — целые, не превосходят 1000 по модулю.

## Формат выходных данных

Программа должна вывести 3 действительных числа — координаты основания перпендикуляра, опущенного из данной точки на данную плоскость с точностью не менее 6 значащих знаков.

## Пример

| perpendicular1.in | perpendicular1.out        |
|-------------------|---------------------------|
| 1 1 1             | 0.3333333333 0.3333333333 |
| 1 1 1 -1          | 0.3333333333              |



# Intersection. Пересечение прямой и плоскости

Имя входного файла: `intersection.in`  
Имя выходного файла: `intersection.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана прямая и плоскость. Найдите точку пересечения прямой и плоскости.

## Формат входных данных

Программа получает на вход 6 целых чисел  $x_1, y_1, z_1, x_2, y_2, z_2$  — координаты двух несовпадающих точек, задающих прямую и 4 целых числа — коэффициенты  $A, B, C, D$  уравнения плоскости. Числа — целые, не превосходят 1000 по модулю.

## Формат выходных данных

Программа должна вывести 3 действительных числа — координаты точки пересечения, данной прямой с данной плоскостью с точностью не менее шести значащих цифр. Если прямая и плоскость не пересекаются — выведите одно число 0. Если прямая целиком лежит в плоскости, выведите одно число 1.

## Пример

| <code>intersection.in</code> | <code>intersection.out</code>          |
|------------------------------|--|
| <code>0 0 0</code>           | <code>0.3333333333 0.3333333333</code> |
| <code>1 1 1</code>           | <code>0.3333333333</code>              |
| <code>1 1 1 -1</code>        |  |



# Tetrahedron. Правильный тетраэдр

Имя входного файла: `tetrahedron.in`  
Имя выходного файла: `tetrahedron.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Даны координаты трех точек в  $\mathbb{R}^3$ , являющиеся вершинами равностороннего треугольника.  
Найдите четвертую точку, образующую с тремя данными правильный тетраэдр.

## Формат входных данных

Программа получает на вход 9 действительных чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$  — координаты трех точек.

## Формат выходных данных

Программа должна вывести 3 действительных числа — координаты четвертой вершины правильного тетраэдра с точностью не менее шести значащих цифр.

## Пример

| <code>tetrahedron.in</code>  | <code>tetrahedron.out</code>           |
|------------------------------|--|
| <code>0 0 0</code>           | <code>0.5000000000 0.2886751333</code> |
| <code>1 0 0</code>           | <code>0.8164965809</code>              |
| <code>0.5 0.8660254 0</code> |  |



## Distance3. Расстояние между двумя прямыми

Имя входного файла: distance3.in  
Имя выходного файла: distance3.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Найдите расстояние между двумя заданными прямыми в  $\mathbb{R}^3$ .

### Формат входных данных

Программа получает на вход 12 чисел  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4$ . Числа — целые, не превосходят 1000 по модулю. Точки 1 и 2, а также 3 и 4 не совпадают.

### Формат выходных данных

Выведите расстояние между прямыми, первая проходит через точки 1 и 2, вторая проходит через точки 3 и 4. Точность вывода не менее шести значащих цифр.

### Пример

| distance3.in                     | distance3.out |
|----------------------------------|---------------|
| 0 0 0<br>1 0 0<br>0 2 0<br>0 2 1 | 2             |



## Perpendicular2. Перпендикуляр к прямой

Имя входного файла: perpendicular2.in  
Имя выходного файла: perpendicular2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана точка и прямая, заданная двумя точками. Найдите основание перпендикуляра, опущенного из данной точки на прямую.

### Формат входных данных

Программа получает на вход 3 целых числа  $x_0, y_0, z_0$  — координаты точки, из которой опущен перпендикуляр, в следующих двух строках — числа  $x_1, y_1, z_1, x_2, y_2, z_2$  — координаты двух различных точек на прямой.

### Формат выходных данных

Программа должна вывести 3 действительных числа — координаты основания перпендикуляра, опущенного из данной точки на данную прямую с точностью не менее 6 значащих знаков.

### Пример

| perpendicular2.in | perpendicular2.out         |
|-------------------|----------------------------|
| 0 0 0             | 0.333333 0.666667 0.333333 |
| 0 1 0             |                            |
| 1 0 1             |                            |





# Tangent. Касательные к сфере

Имя входного файла: `tangent.in`  
Имя выходного файла: `tangent.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти:

Дана сфера и две точки. Постройте касательные к сфере плоскости, проходящие через данные точки, и найдите точки касания плоскостей со сферой.

## Формат входных данных

Программа получает на вход 3 целых числа  $x_0, y_0, z_0$  — координаты центра сферы. Во второй строке записано целое положительное число  $r$  — радиус сферы. В следующих двух строках — числа  $x_1, y_1, z_1, x_2, y_2, z_2$  — координаты двух различных точек, через которые должны проходить касательные.

## Формат выходных данных

Программа должна вывести целое число  $k$  — количество различных касательных плоскостей, которые можно провести через данные точки к сфере. В следующих  $k$  строках необходимо вывести координаты точек касания данных плоскостей со сферой.

## Пример

| <code>tangent.in</code>      | <code>tangent.out</code>  |
|------------------------------|---|
| 1 1 1<br>1<br>0 0 0<br>1 0 0 | 2<br>1 0 1<br>1 1 0   |
| 1 1 1<br>3<br>0 4 3<br>2 1 5 | 2<br>2 3 3<br>-0.28571428571428537<br>1.8571428571428577 3.5714285714285716 |
| 0 0 0<br>1<br>0 1 0<br>0 1 1 | 1<br>0 1 0  |