Лабораторная работа № 4:
«Выпуклая оболочка»
Техническое задание

Задание A (convex hull, 7 баллов)

Программа решает задачу построения выпуклой оболочки конечного множества точек в трёхмерном пространстве.

Условие: *задан набор целочисленных точек в трёхмерном пространстве.*

Задача: *построить выпуклую оболочку точек (многогранник), как множество решений системы линейных неравенств $Ax \leq b$ (граней многогранника).*

Программа получает название файла с входными данными в качестве параметра командной строки:

```
> convexhull.exe test42.txt
```

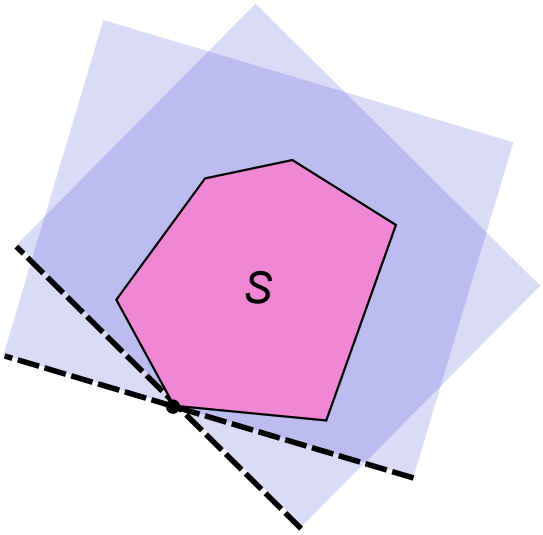
Структура входного файла: сначала указывается формат входного файла **V** (от vertex – вершина, множество точек), затем число **целочисленных** точек трехмерного пространства, и, наконец, координаты построчно

```
V
4
0 0 0
0 0 1
0 1 0
1 0 0
```

Считанные координаты точек программа выводит в поток. Если входной файл не задан, то множество точек вводится вручную через поток.

Программа строит выпуклую оболочку (многогранник) на основе следующего принципа.

Идея: *если через 3 точки, не лежащие на одной прямой, можно провести такую плоскость, что все оставшиеся точки попадут строго в одно полупространство или на саму плоскость, то эта плоскость называется опорной и ее уравнение входит в выпуклую оболочку (опорная плоскость содержит грань искомого многогранника).*



Результат работы выводится в поток в виде системы неравенств:

```
Number of faces: 4
-x <= 0
-y <= 0
-z <= 0
x + y + z <= 1
```

При желании можно отформатировать вывод с неравенствами вида $\geq$.

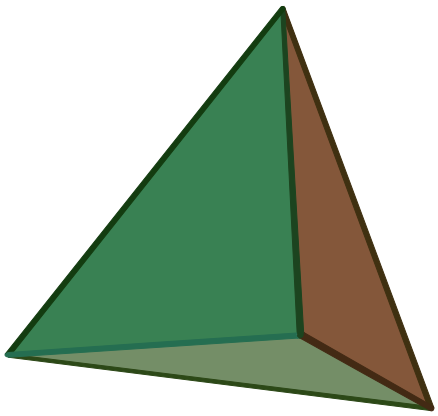
Замечание: система в ответе не должна содержать линейно зависимых неравенств:

```
x + y + z <= 1
3x + 3y + 3z <=3
```

Примеры задания трехмерных многогранников как множество вершин и как решение системы неравенств:

Тетраэдр:

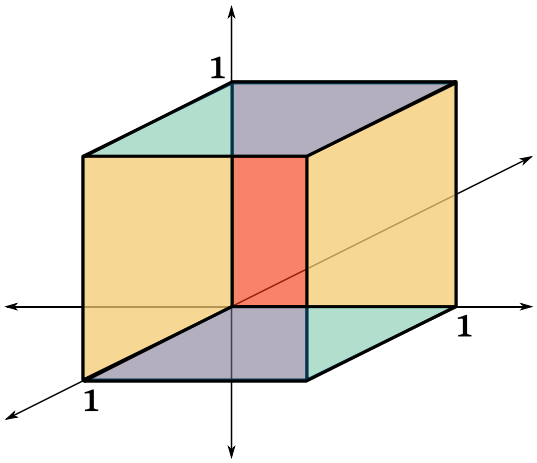
$$conv\{(0,0,0),(0,0,1),(0,1,0),(1,0,0)\} \Leftrightarrow \{x \geq 0, y \geq 0, z \geq 0, x + y + z \leq 1\}.$$



Куб:

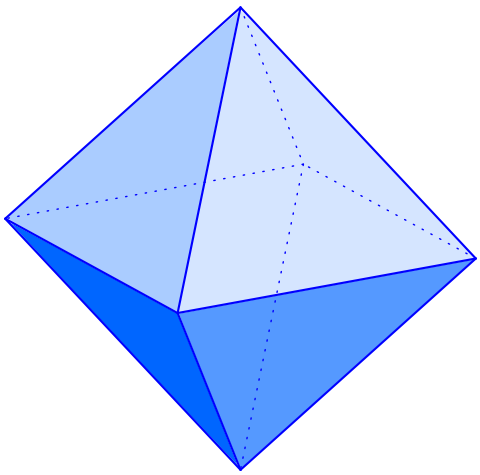
$$conv\{(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)\} \Leftrightarrow$$

$$\Leftrightarrow \{0 \le x \le 1, 0 \le y \le 1, 0 \le z \le 1\}$$



Октаэдр:

$$conv\{(\pm 1,0,0),(0,\pm 1,0),(0,0,\pm 1)\} \Leftrightarrow \{\pm x \pm y \pm z \le 1\}.$$



Задание B.1 (vertex enumeration, 4 балла)

Программа решает обратную задачу «перечисления вершин».

Условие: *задан трехмерный выпуклый многогранник как множество решений системы линейных неравенств* $Ax \le b$.

Задача: *найти все его вершины.*

Формат входного файла обозначается буквой **H** (от hyperplane, гиперплоскость - грань). Затем указываются число неравенств и, построчно, коэффициенты неравенств

$$ax + by + cz \leq d \quad \Leftrightarrow \quad a\ b\ c\ d$$

Пример:

```
H
4
-1  0  0  0
0  -1  0  0
0  0  -1  0
1  1  1  1
```

Считанную систему программа выводит в поток в форматированном виде, как в части **A**.

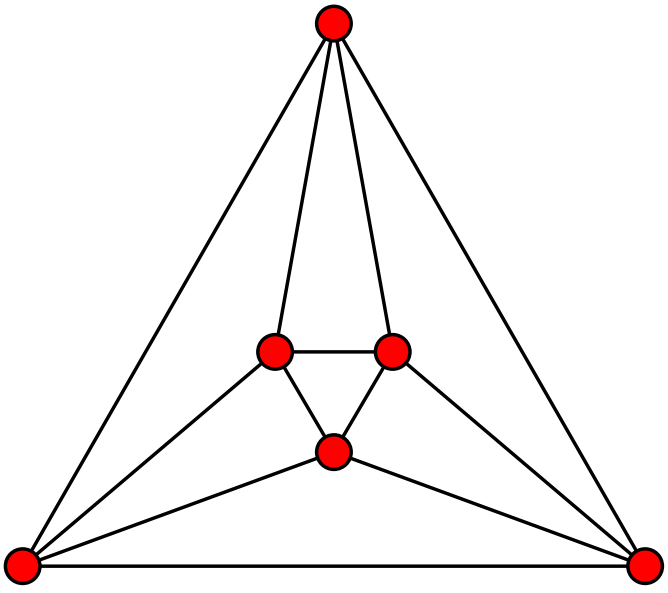
Найденные вершины программа записывает в поток, обозначая их буквами латинского алфавита. Если букв не хватит (хотя я честно обещаю не вводить многогранники на более 26 вершинах), следует добавлять к букве – цифру (`A1`)

```
Number of vertices: 4

A: 0  0  0
B: 0  0  1
C: 0  1  0
D: 1  0  0
```

Задание B.2 (1-skeleton, 4 балла)

Полиэдральный граф – неориентированный граф, образованный из вершин и рёбер выпуклого многогранника. Как пример, можно рассмотреть полиэдральный граф октаэдра (см. также теорему Штейница в лекции).



- Если на вход подано множество точек, то программа строит выпуклую оболочку, затем находит вершины многогранника (задание **B.1**.), после чего строит полиэдральный граф многогранника.
- Если на вход подана система линейных неравенств, то сразу выполняется **B.1** (можно реализовать только **B.1** и **B.2**, без **A**).

Граф выводится в двух форматах. Первый – матрица смежности (вершины в алфавитном порядке):

```
Adjacency matrix
  A B C D
A 0 1 1 1
B 1 0 1 1
C 1 1 0 1
D 1 1 1 0
```

Во втором формате каждой грани сопоставляются принадлежащие ей вершины и ребра:

```
Face: x + y + z <= 1
Vertices: B, C, D
Edges: BC, BD, CD
Face: -x <= 0
Vertices: A, B, C
```

и т.д.

Задание B.3 (collision detection, 3 балла)

В программу добавляется третий режим работы. На вход поступают два файла с трёхмерными многогранниками

```
> convexhull.exe test42.txt test13.txt
```

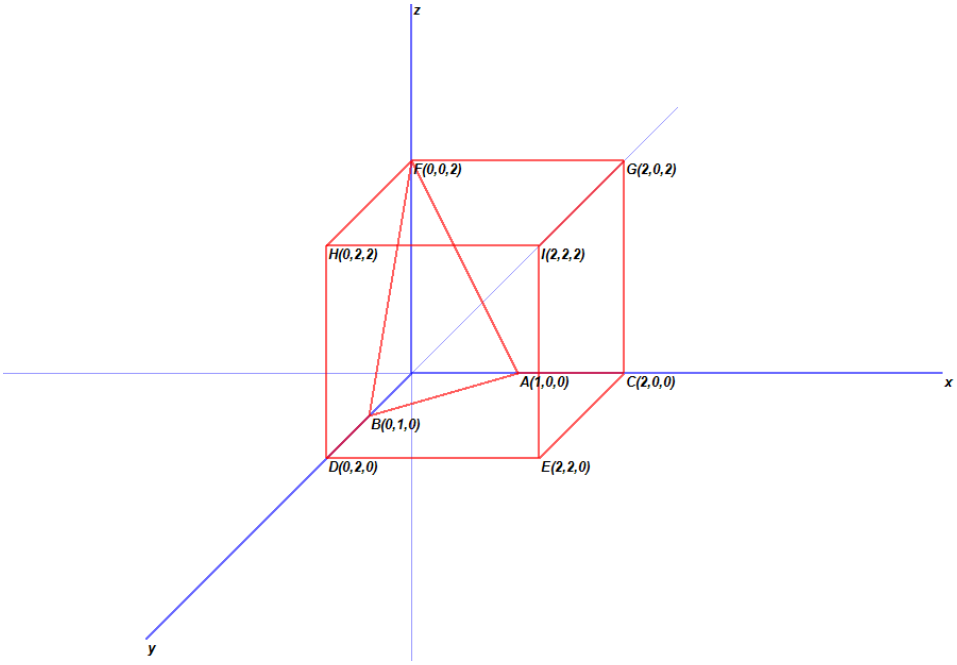
Для них выполняются остальные задания, если они реализованы, после чего **через разность Минковского** определяется пересекаются многогранники или нет.

Лемма. *Многогранники пересекаются тогда и только тогда, когда их разность Минковского содержит начало координат* $(0,0,0)$.

На выход подается сообщение об обнаруженной коллизии.

Задание C (3D graphics, 6 баллов)

Программа реализует графический вывод построенной выпуклой оболочки. Например, вот так (но можно сделать и лучше):



Грани делаются прозрачными (рисуются только ребра, как раз те, что в **B.2**). Обязательно отрисовываются оси, и выводятся координаты точек.

Если реализовано задание B.3, то на одной картинке отрисовываются оба многогранника.

Графику можно реализовать с помощью GDI+:

https://docs.microsoft.com/en-us/windows/desktop/gdiplus/-gdiplus-gdi-start

https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/advanced/about-gdi-managed-code

Или любым другим способом, который будет работать 😀.

Язык C, C++ или Python.

Требования к программе.

Требование № 1. В названии файла с исходным кодом указывается фамилия и инициалы автора. Например, программа Иванова Ивана Ивановича будет называться

`Ivanov_II.cpp или Ivanov_II.py`

Задания должны быть реализованы в рамках одной универсальной программы.

Требование № 2. Код следует сопроводить подробными комментариями, чтобы его можно было прочесть и понять.
Требование № 3. Использование стандартных библиотек возможно, только если они не связаны с элементами курса «Алгебра и геометрия». Библиотеки NumPy, SymPy и SciPy запрещаются.

Как обычно, 3 лучших программы дополнительно получают:

3 балла – 1 место, 2 балла – 2 место, 1 балл – 3 место.

Срок сдачи: 9 апреля 2023