PROBLEMA 1.

Implementați o soluție cât mai eficientă (O(n^2)) pentru problema **3-SUM**: Să se afișeze tripletele (distincte) de elemente ale unui vector de numere întregi care au suma 0.

Exemplu:

date.in	date.out
8 3 1 2 -5 -2 10 7 3	(-5, 2, 3) (-5, -2, 7) nu neapărat în această ordine.

a) Care este complexitatea timp a algoritmului propus în 3SUM.cpp? Cum se poate optimiza?
b) Care este complexitatea timp a algoritmului propus în 3SUMb.cpp. Cum se poate optimiza?
c) Modificați 3SUM.cpp astfel încât tripletele afișate să fie distincte în cazul în care vectorul conține duplicate.

date.in	date.out
9 3 1 2 **2** -5 -2 10 7 3	(-5, 2, 3) (-5, -2, 7) nu neapărat în această ordine.

EXTRA

Decision problem: Fiind dată o mulțime ***S***, există un triplet $(a, b, c) \in S^3$ a.î $a + b + c = 0$?

3SUM-Hard Problems: Pot fi reduse la 3SUM în $O(n^2)$. Exemple:

- Dându-se o mulțime *S* de *n* puncte în plan, având coordonate întregi pe trei drepte paralele $y = 0$, $y = 1$, și $y = 2$ să se determine dacă există o dreapta care să conțină 3 puncte din *S* și să nu fie paralelă cu $y = 0$.
- Dându-se o mulțime de benzi în plan, reuniunea acestora conține un dreptunghi ale cărui laturi sunt paralele cu axele sistemului de coordonate?
- Dându-se o mulțime de triunghiuri în plan, reuniunea acestora conține un alt triunghi dat *T* ?

PROBLEMA 2.

Se dau *n* paralelipipede prin dimensiunile acestora *(L,l,h)*. Să se determine sfera de raza maximă care se poate înscrie într-unul dintre cele *n* paralelipipede sau într-un paralelipiped format prin suprapunerea a două dintre cele *n* paralelipipede. Spre exemplu (1,5,4) și (4,5,6) pot forma un paralelipiped (5,4,7) iar sfera de rază maximă ce se poate înscire în (5,4,7) are raza 2.

TEMA 1.

Distanța între documente

Fie două documente text, F_1 și F_2, și $\{c_1, c_2, \dots, c_n\}$ mulțimea cuvintelor care apar în cel puțin unul din cele două documente. Pentru $1 \leq i \leq n$, fie v_{i1}, v_{i2} numărul de apariții al cuvântului i în primul, respectiv în al doilea document. Distanța cosinus dintre cele două documente, notată $dcos(F_1, F_2)$, dintre F_1 și F_2 se calculează după formula:

$$dcos(F_1, F_2) = \frac{\sum_{i=1}^{n} v_{i1} v_{i2}}{\sqrt{\sum_{i=1}^{n} v_{i1}^2} \sqrt{\sum_{i=1}^{n} v_{i2}^2}}$$

Fiind date două fișiere text, *F1* și *F2*, să se calculeze distanța cosinus dintre cele două fișiere. Fiecare dintre cele două fișiere va fi parcurs o singură dată. Cuvintele dintr-un fișier sunt separate prin spații, virgulă, punct.

Exemplu: dacă F_1 conține textul: *primul laborator primul exercitiu* și F_2 conține textul: *primul exercitiu usor*

$\{c_1 =' primul', c_2 = 'laborator', c_3 = 'exercitiu'\ c_4 = 'usor'\}$
$\{v_{11} = 2,\ v_{21} = 1,\ v_{31} = 1,\ v_{41} = 0\}$
$\{v_{12} = 1,\ v_{22} = 0,\ v_{32} = 1,\ v_{42} = 1\}$

$$dcos(F_1, F_2) = \frac{2 * 1 + 1 * 0 + 1 * 1 + 0 * 1}{\sqrt{6}\sqrt{3}} = 0{,}7071$$

Bibliografie:

- https://en.wikipedia.org/wiki/Cosine_similarity
- *On a class of $O(n^2)$ problems in computational geometry*. Anka Gajentaan, Mark H. Overmars in Computational Geometry: Theory and Applications