**Programozási alapismeretek**

**komplex beadandó**

Készítette: Fábián Tímea Nikolett

Neptun kód: RDDZXA

E-mail: [timeanikolett2002@gmail.com](mailto:timeanikolett2002@gmail.com)

Kurzuskód: IP-18PROGEG

Gyakorlatvezető neve: Wolosz Jakab

**2022. január 04.**

*Tartalom*

[**Feladat** 3](#_Toc92194903)

[**Kód** 3](#_Toc92194904)

[**Specifikáció I.** 6](#_Toc92194905)

[**Algoritmus I.** 6](#_Toc92194906)

[**Specifikáció II.** 7](#_Toc92194907)

[**Algoritmus II.** 7](#_Toc92194908)

# **Feladat**

Egy N résztvevőjű kutyaszépségversenyen M különböző szempont szerint pontoznak minden kutyát. Minden szemponthoz adott egy maximális pontszám. Az összetett versenyből automatikusan kiesik az a kutya, amely valamelyik szempont szerint nem éri el a szempontonként megadott alsó ponthatárt – az adott szempontból sem értékelhető, más szempontokból viszont igen.

Készíts programot, amely megadja azokat a kutyákat, amelyek átlagosan teljesítettek.

# **Kód**

#include <iostream>

**using namespace std**;

**int** main()

{

**int** N;

**cin** >> N;

**int** M;

**cin** >> M;

**int** MIN[M];

**int** MAX[M];

**for**(**int** i=0; i<M; i++)

{

**cin** >> MAX[i];

}

**for**(**int** i=0; i<M; i++)

{

**cin** >> MIN[i];

}

**int** pont[N][M];

**for**(**int** i=0; i<N; i++)

{

**for**(**int** j=0; j<M; j++)

{

**cin** >> pont[i][j];

}

}

**int** versenyzok[N]; //aktualis versenyzok

**for**(**int** i=0; i<N; i++)

{

versenyzok[i]=i+1;

}

**int** helyiMaximumok[M];

**int** helyiMinimumok[M];

**for**(**int** j=0; j<M; j++) //helyi max és min értékek megkeresése

{

**int** maxPont = MAX[j];

**int** minPont = MIN[j];

**int** helyiMax = minPont;

**int** helyiMin = maxPont;

**for**(**int** i=0; i<N; i++)

{

**if**(pont[i][j]>helyiMax)

{

helyiMax = pont[i][j];

}

}

helyiMaximumok[j] = helyiMax;

**for**(**int** i=0; i<N; i++)

{

**if**(pont[i][j]<helyiMin)

{

helyiMin = pont[i][j];

}

}

helyiMinimumok[j] = helyiMin;

}

**for**(**int** i=0; i<N; i++)

{

**for**(**int** j=0; j<M; j++)

{

**if**(pont[i][j]==helyiMaximumok[j])

//kiszuri a jo pontszamosokat

{

versenyzok[i] = -1;

}

**else** **if**(pont[i][j]==helyiMinimumok[j] && helyiMinimumok[j]<MIN[j])

//kiszuri a rossz pontszamosokat

{

versenyzok[i] = -1;

}

}

}

**int** db = 0;

**int** atlagos[N];

**for**(**int** i=0; i<N; i++)

{

**if**(versenyzok[i] != -1)

{

atlagos[db] = versenyzok[i];

db++;

}

}

**cout** << db;

**for**(**int** i=0; i<db; i++)

{

**cout** << " " << atlagos[i];

}

**return** 0;

}

# Text, letter Description automatically generated**Specifikáció I.**

# **Algoritmus I.**

Table

Description automatically generated

# **Specifikáció II.**

Text, letter

Description automatically generated

# **Algoritmus II.**

Diagram, table

Description automatically generated