**Semestrální Úloha číslo 29**

**Technická dokumentace**

**Zpracoval: Jan Vyhlídko**

**Předmět: ALG1**

**Datum: 20.1. 2023**

**Zadání :** Zapište **program**, který bude **testovat, zda zadaný systém** *n* **vektorů o** *n* **složkách je**

**ortogonální**. Pokud ano nechť program dále **provede převod vektorů na normovaný tvar.**

**Specifikace vstupu**

Program má umožnit při jednom spuštění zpracování libovolného počtu zadání. Pro každou úlohu nechť program nejdříve načte celé číslo *n* (představující počet vektorů i počet složek každého z vektorů). Poté program načte jednotlivé vektory a vypíše výsledek testu ortogonality v podobě krátké zprávy tvaru „System je ortogonalni“ nebo „System není ortogonální“. Po načtení záporného nebo nulového *n* nechť program skončí svoji činnost.

* **Specifikaci požadavků : vektory, které uživatel bude zadávat musí být stejně dlouhé, program pracuje s presnosti 0.000001, musí být zadány pouze číslici, jinak program spadne.**

**Návrh řešení**

1. **prvně dám while cyklus, v jeho podmínce načtu počet vektorů, pokud to číslo bude záporné ukončím cyklus.**
2. **všechny dané vektory si načtu do matice a vytvořím pomocné pole, které bude obsahovat délky vektorů, které si ke každému vektoru spočítám. abych mohl na konci vydělit každý člen delkou vektoru.**
3. **delku vektoru spočítám : první prvek vektoru na druhou + druhy prvek vektoru na druhou .atd a to celé je pod druhou odmocninou**
4. **pokud zjistím že délka nějakého vektoru je nulová -> zapíšu si, že není ortogonální a ukončím cyklus.**
5. **kontrola skalárního součinu vektorů -> budu procházet po řádcích matice, kdy budu kontrolovat první řádek s druhým pak třetím a atd.(budu pocitat skalarni soucin mezi nimi), pak pujdu na druhý řádek a budu kontrolovat skalární součiny s třetím řádkem pak s třetím atd. , když projdu až na konec a ortogonalni není false. tak je matice ortogonální a provedu normalizovaný systém.**
6. **skalární součin spočítám : první prvek prvního vektoru \* první prvek druhého vektoru + druhý prvek prvního vektoru \* druhý prvek druhého vektoru atd.**
7. **pro vytvoření normalizovaného systému musím každý prvek vektoru vydělit délkou vektoru ve kterém se nachází.**

**Závěr**

| **Číslo testu** | **Typ testu, popis vstupů** | **Očekávaný výsledek** | **Skutečný výsledek** | **Prošel (ano/ne)** |
| --- | --- | --- | --- | --- |
| **1** | **jeden vektor,**  **vektor : 1** | **je ortogonální** | **je ortogonální** | **ano** |
| **2** | **pocet vektoru :3**  **vektory:**  **2 3,1 0**  **0 -1 0**  **0 0 1** | **neni ortogonální** | **neni ortogonální** | **ano** |
| 3 | pocet vektoru : 4  vektory:  2 0 0 0  0 5 0 0  0 0 0 4  0 0 3,6 0 | **je ortogonální**  Normalizovany system:  1,000 0,000 0,000 0,000  0,000 1,000 0,000 0,000  0,000 0,000 0,000 1,000  0,000 0,000 1,000 0,000 | **je ortogonální**  Normalizovany system:  1,000 0,000 0,000 0,000  0,000 1,000 0,000 0,000  0,000 0,000 0,000 1,000  0,000 0,000 1,000 0,000 | ano |
| 4 | pocet vektroru: 0 | program se ukončí | program se ukončí | ano |
| 5 | pocet vektoru : 2  vektory :  0 0  0 0 | **neni ortogonální** | **neni ortogonální** | ano |
| 6 | pocet vektoru : 2  vektory :  1 0  0 1 | **System je ortogonalni**  **Normalizovany system**  **1,000 0,000**  **0,000 1,000** | **System je ortogonalni**  **Normalizovany system**  **1,000 0,000**  **0,000 1,000** | ano |

\*