­

2022

Mykhailo Maidiuk

Technická univerzita v Liberci

Z Algoritmizace a programování

Technická dokumentace

**Úkol a předpoklady, pod kterými bude program fungovat**

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**.

***Normovaným vektorem*** **U**n k vektoru u nazveme vektor, který má délku 1 a jehož směr je totožný se směrem původního vektoru. **Normovaný tvar** získáme vydělení každé složky

původního vektoru jeho normou, tedy.

Systém n vektorů o n složkách nazveme ***ortogonálním systémem*** právě tehdy, když všechny vzájemné skalární součiny (dvou navzájem různých! vektorů) jsou **rovny** 0 a žádný z vektorů **nemá** **nulovou délku**.

**Aby program fungoval, musí uživatel zadat:**

1. Počet vektorů musí obsahovat pouze **celá čísla**. A musí být **pozitivní**.

2. Při vyplňování hodnot vektorů může zadávat jak **celá** (33, 5, ...), tak **zlomková čísla** (1.11, -8.4, …). Hodnoty mohou být větší než **nula nebo menší**.

3.Uživatel si nesmi psat znaky jako “ !, ?, \*” а jiný.

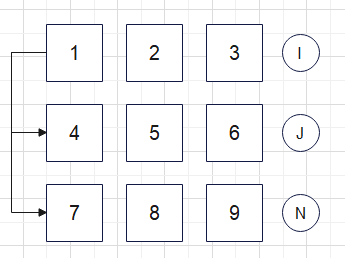
4. Počet vektorů musí být větší než dva.

**Řešení, jak jsem k problému přistupoval**

Abychom mohli vypočítat ortogonalitu vektorů, musíme vědět, co to je. Abychom mohli do konzole odeslat normalizovaný systém, musíme také vědět, co to je.

**Popíšu algoritmus řešení**:

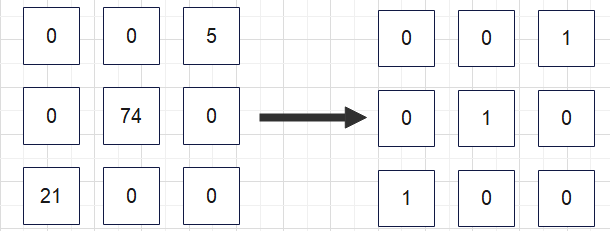
1. Určuji počet vektorů v matici.
2. Velikost pole se rovná počtu vektorů.
3. Kód kontroluje vektorovou ortogonalitu takovým způsobem, že iteruje každou dvojici vektorů ve **vektorovém poli** a vypočítává bodový součin těchto dvou vektorů.



1. Skalární součin se vypočítá iterací přes **každý složky** vektorů a vynásobením odpovídajících prvků dohromady.

**x = i \* j; x2 = j \* n;**

1. Po kontrole všech párů vektorů, pokud jsou ortogonální, vrátí true jinak false.
2. Pokud je **nepravda**, zobrazíme nápis, že vektory nejsou ortogonální, jinak pokračujeme
3. Převédeme vektor do normalizovane formy.



1. Pro každý **vektor** funkce vypočítá délku vektoru sečtením druhých mocnin každého prvku v řádku a udělaním druhé odmocniny součtu.

**X = A1\*A1;**

**Y =** √x;

1. Poté každou složku vektoru vidělime délkou vektoru a výslednou hodnotu uložíme do odpovídajícího prvku.

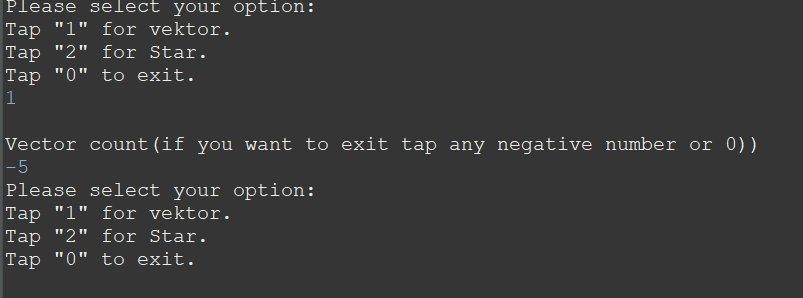
**Z = A1 / Y;**

1. Výsledek vypíšeme do konzole

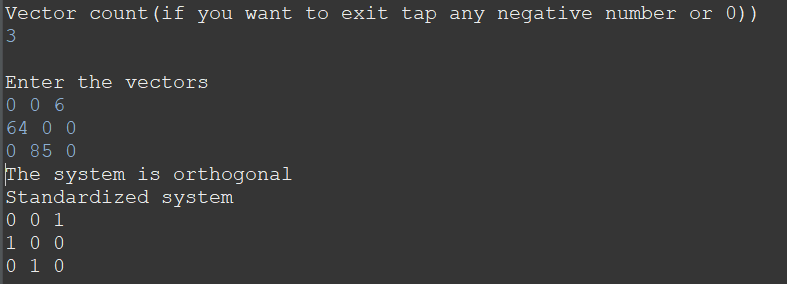
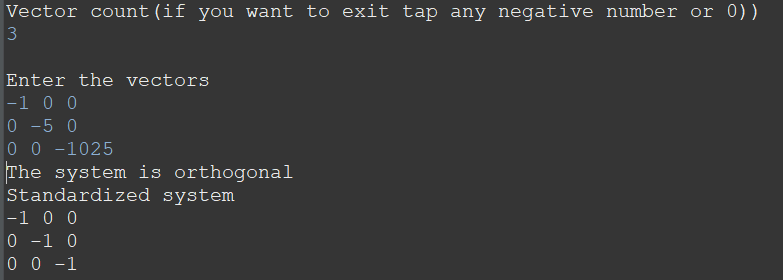
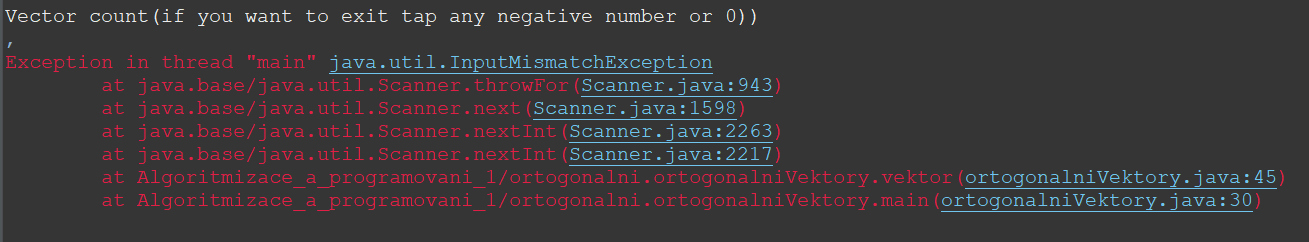
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Číslo testu** | **Typ testu, popis vstupů** | **Očekávaný výsledek** | **Skutečný výsledek** | **Prošel (ano/ne)** |
| №1 | Zadám záporný  počet vektorů | Program se zastaví | Program se zastaví | **Ano** |
| №2 | V matici uvedu všechna kladna čísla | The system is orthogonal | The system is orthogonal | **Ano** |
| №3 | V matici uvedu všechna záporná čísla | The system is orthogonal | The system is orthogonal | **Ano** |
| №4 | Zadam čárku v počtu vektorů | Error | Error | **Ano** |
| №5 | Ve vektoru uvedu všechna neceločíselná čísla | The system is orthogonal | The system is orthogonal | **Ano** |
| №6 | Zadá otazník ve vektoru | Error | Error | **Ano** |

**Protokol z testování**

**Screenshoty výsledků akceptačních testů**



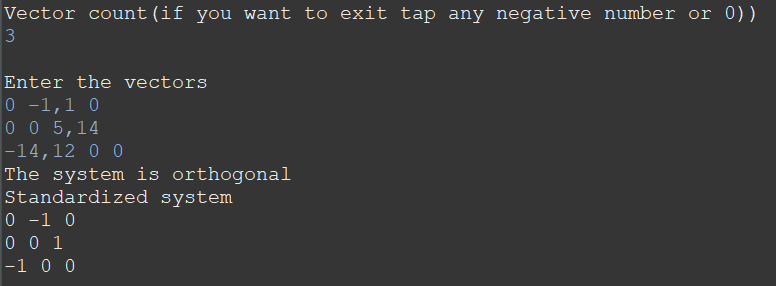
№1



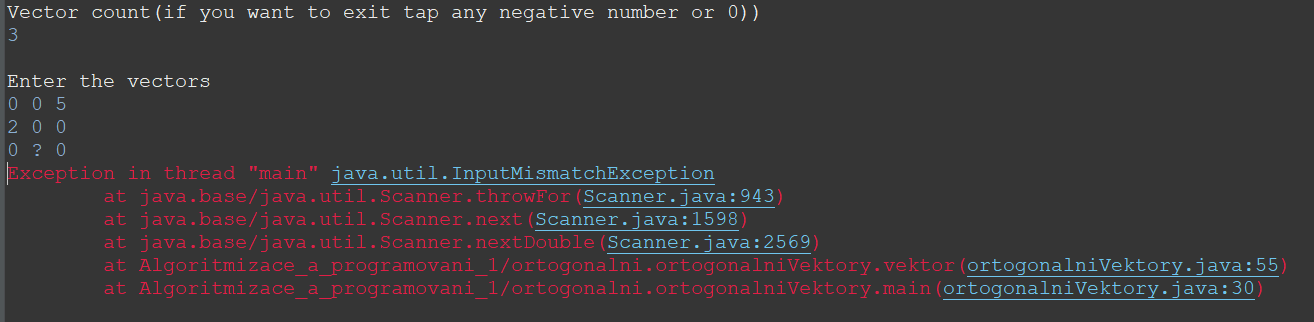
№4

№2

№3



№5



№6