

# IZP - Projekt č. 3 - Maticové operace

Než začnete programovat, přečtěte si pozorně celé zadání.  
Poslední změna: 23. září 2010

## Motivace

Cílem této úlohy je zopakovat si nebo se naučit

- vytváření obecných řešení, která na rozdíl od ad hoc řešení umožňují zvládat složitější úlohy bez nadměrného úsilí,
- zapojit prostorovou představivost a abstraktní myšlení, než začnete programovat,
- základní matematické operace s maticemi,
- práci s vícerozměrnými poli,
- práci s dynamicky alokovanou pamětí,
- načítání dat ze souborů
- ošetřování problémů, které mohou nastat při práci se soubory.

## Zadání úlohy

Napište program, který bude provádět operace s vektory a maticemi obecných rozměrů obsahující celá čísla (použijte datový typ `int`). Program musí ošetřit situace, kdy vektory a matice nemají pro operaci vhodné rozměry nebo jiné vlastnosti dané podstatou operací. Implementujte tyto operace:

Povinně všechny tyto operace:

- Kontrola vstupních dat
- Vektorový součet dvou vektorů ( $\mathbf{a} + \mathbf{b}$ )
- Skalární součin dvou vektorů ( $\mathbf{a} \cdot \mathbf{b}$ )
- Součin matic ( $\mathbf{A} * \mathbf{B}$ )
- Maticový výraz ( $\mathbf{A} * \mathbf{B} * \mathbf{A}$ )

Povinně dvě z následujících operací:

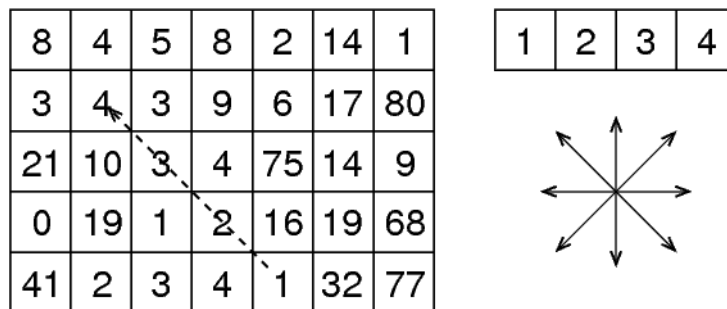
- Osmisměrka
- Počet bublin
- Bludiště
- Kulečník

Před programováním se zamyslete nad mezními stavy těchto operací (různé rozměry vektorů a matic, různé hodnoty prvků, ...). Zvláštní pozornost věnujte analýze problému násobení matic a zvolených operací.

## Bližší specifikace

### Osmisměrka

Vstupem úlohy jsou vektor a matice libovolných rozměrů. Úkolem je zjistit, zda se zadaný vektor nachází v zadané matici v kterémkoli z osmi možných směrů. Jde o analogii známého rébusu. Úloha má řešení, pokud lze v matici najít lineární posloupnost prvků shodných s prvky zadaného vektoru, přičemž tato spojnice může ležet kdekoli v matici a může být natočena kterýmkoli z osmi možných směrů (shora dolů, zdola nahoru, zleva doprava, zprava doleva, zleva šikmo dolů doprava, zprava šikmo dolů doleva, zleva šikmo vzhůru doprava a zprava šikmo vzhůru doleva). Jednu možnou úlohu demonstruje následující obrázek:



Úkolem je zjistit, zda se zadaný vektor nachází v některém z osmi směrů v zadané matici. Pokud ano, vytiskněte na standardní výstup text `true`, pokud ne, vytiskněte text `false`.

### Počet bublin

Vstupem úlohy je matice libovolných rozměrů, která bude obsahovat pouze prvky s hodnotami rovnými a většími než nula. Pro lepší přiblížení podstaty úlohy zavedme tuto představu: matice je diskretizovaným řezem pórovitým materiálem, kde prvky s hodnotou nula představují prázdné místo - bubliny a prvky s hodnotou větší než nula představují místo vyplněné materiálem.

Samostatnou bublinu tvoří buďto jediný nulový prvek, který ve svém čtyřokolí nemá žádný další nulový prvek nebo ji tvoří množina nulových prvků, které spolu sousedí ve svém čtyřokolí. Čtyřokolí je množina prvků bezprostředně sousedících se zkoumaným prvkem, které se liší ve vertikálním nebo horizontálním indexu (jen v jednom z nich) o jedničku od souřadnic zkoumaného prvků.

Úkolem je spočítat, kolik takto definovaných samostatných bublin obsahuje zadaná matice. Výslednou hodnotu vytiskněte jako dekadické číslo na standardní výstup.

Například v matici na demonstračním obrázku je 5 bublin:

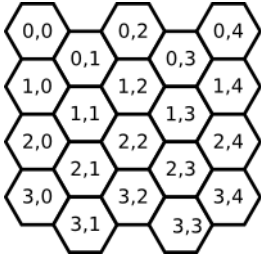
8	4	5	8	2	0	0
3	4	3	0	0	17	80
21	10	0	4	8	0	0
0	19	0	0	16	0	0
41	2	3	4	1	32	77

**Premiové rozšíření - prostorová varianta**

Zadání je v principu stejné, ale místo matice se zde bude pracovat s vektorem matic, který lze nyní interpretovat jako diskretizovaný výřez prostoru. Bublina je nyní definována podobně jako v plošné variantě, ale sousednost je nyní definována v prostorovém šestiokolí. Šestiokolí je množina prvků bezprostředně sousedících se zkoumaným prvkem, které se liší v indexu ve směru osy x nebo ve směru osy y nebo ve směru osy z (jen v jednom z nich) o jedničku od souřadnic zkoumaného prvků.

**Bludiště**

Cílem úkolu je najít cestu bludištěm v podobě šestiúhelníkové mapy. Vstupem je matice představující bludiště. Sloupce matice představují vertikálně sousedící šestiúhelníky. Řádky vstupní matice představují šestiúhelníky sousedící střídavě jihovýchodním a severovýchodním směrem. Například matice o velikosti 4 řádků a 5 sloupců reprezentuje následující mapu:



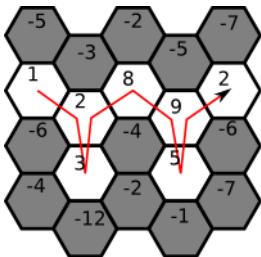
První zadaný prvek matice představuje hodnotu šestiúhelníku označeného indexy 0,0; následující prvek je hodnota šestiúhelníku 0,1 atd.

Záporné hodnoty prvků šestiúhelníků představují zeď bludiště, nezáporné hodnoty pak průchozí prostor. Do bludiště se vždy vstupuje zleva na šestiúhelník na pozici 1,0. Implementujte průchod bludištěm podle pravé ruky tzn. průchod, ve kterém je vždy po pravé ruce zeď (očekávejte, že šestiúhelník na pozici 2,0 je vždy zeď). Východ z bludiště je průchozí šestiúhelník, který je na hranici mapy.

Úkolem je na výstup vypisovat hodnoty šestiúhelníků, kterými se prochází. Poslední vypsána hodnota tedy bude hodnota východu z bludiště. Pokud úloha nemá řešení, vypište na výstup pouze text false. Celá cesta bude vytisknuta na jediný řádek a hodnoty budou odděleny jednou mezerou.

**Příklad:** Průchod následujícím bludištěm definované maticí o 4 řádcích a 5 sloupcích vytvoří posloupnost:

1 2 3 2 8 9 5 9 2

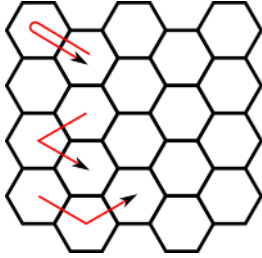


**Kulečník**

Vstupem je matice reprezentující šestiúhelníkovou mapu představující plochu hypotetického kulečnickového stolu (šestiúhelníková mapa je principiálně stejná, jako v předchozí úloze Bludiště). Sloupce matice představují vertikálně sousedící šestiúhelníky, řádky vstupní matice představují šestiúhelníky sousedící střídavě jihovýchodním a severovýchodním směrem. Navíc k mapě kulečnicku je vstupními parametry programu udána počáteční pozice (souřadnice) kulečnickové koule, směr a síla jejího odpalu.

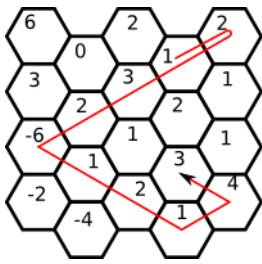
Směr odpalu je zadán jedním z šesti řetězců: S, SV, JV, J, JZ a SZ (tj. severní směr, severovýchodní, jihovýchodní, jižní, jihozápadní a severozápadní). Síla odpalu je celé číslo vyjadřující délku cesty v počtu šestiúhelníků, kterými koule putuje.

Výstupem programu je posloupnost hodnot šestiúhelníků, po kterých se koule pohybuje. Posloupnost vypisujte na jediný řádek, hodnoty odděľujte jednou mezerou. Při pohybu za normálních okolností udržuje koule stejný směr. V případě, že koule narazí na okraj mapy, změni patřičným způsobem směr pohybu. V šestiúhelníkové mapě mohou nastat tři typy odrazů naznačených na následujícím obrázku:



**Příklad:** Při spuštění s následující maticí, počáteční pozicí kulečnickové koule "0,3", směrem odpalu "SV" a silou odpalu 11 bude trasa koule následující:

1 2 1 3 2 -6 1 2 1 4 3



### Formát souborů s vektory, maticemi a vektory matic

Na začátku každého souboru bude hodnota N, což bude celé číslo 1, 2 nebo 3. Hodnota 1 znamená vektor, hodnota 2 znamená matici a hodnota 3 znamená vektor matic.

Další údaje budou znamenat rozměry. U vektorů bude následovat jedna hodnota znamenající *počet prvků vektoru*. U matice budou následovat dvě hodnoty s významem *počet řádků* *počet sloupců* v tomto pořadí. U vektoru matic budou následovat tři hodnoty s významem *počet\_matic* *počet řádků* *počet sloupců* v tomto pořadí.

Na zbylých řádcích budou prvky dané struktury oddělené mezerami. Každý řádek matice bude odpovídat jednomu řádku v souboru (u vektorů a vektorů matic to bude vypadat analogicky). Není chybou, pokud mezi hodnotami bude více mezer. Na konci souboru může být prázdný řádek (pro zjednodušení). Tento formát je společný pro vstup i pro výstup. U vstupních souborů není potřeba kontrolovat formátování (řádkování, mezery), výstupní soubory musí této specifikaci vyhovovat co nejpřesněji.

Příklad vektoru o třech prvcích:

```
1
3
1 2 3
```

Příklad matice o pěti řádcích a sedmi sloupcích:

```
2
5 7
1 2 3 4 3 2 1
2 3 4 5 4 3 2
3 4 5 10 5 4 3
2 3 4 5 4 3 2
1 2 3 4 3 2 1
```

Příklad vektoru matic o rozměrech 2x3x4, tedy 2 matice po 3 řádcích a 4 sloupcích (prázdný řádek mezi první a druhou maticí je nepovinný):

```
3
2 3 4
1 2 3 4
2 3 4 5
3 4 5 10
```

2 3 4 5  
1 2 3 4  
2 3 4 5

Specifikace parametrů

Program při jednom spuštění provede jedinou operaci, která bude zadána parametrem podle následující tabulky.

Operace	Parametry	Popis	Váha při hodnocení
Nápověda	proj3 -h	Vytiskne nápovědu k použití programu.	
Testovací výstup	proj3 --test data.txt	Pouze zkontroluje, zda jsou data ve vstupním souboru ve správném formátu a správně naformátovanou strukturu opět vytiskne na standardní výstup.	1%
Vektorový součet	proj3 --vadd a.txt b.txt	Sečte dva vektory ( <b>a+b</b> ) a ošetří případné chyby.	5%
Skalární součin vektorů	proj3 --vscal a.txt b.txt	Vypočte skalární součin ( <b>a*b</b> ) dvou vektorů a ošetří případné chyby.	5%
Součin dvou matic	proj3 --mmult A.txt B.txt	Vypočte součin dvou matic v daném pořadí ( <b>A*B</b> ) a ošetří případné chyby.	10%
Maticový výraz A*B*A	proj3 --mexpr A.txt B.txt	Vypočte maticový výraz ( <b>A*B*A</b> ) a ošetří případné chyby.	10%
Osmisměrka	proj3 --eight v.txt M.txt	Vyhledá v matici M vektor v.	30%
Bubliny	proj3 --bubbles M.txt	Spočítá počet bublin v matici M.	30%
Bubliny - rozšíření do 3D	proj3 --extbubbles 3DM.txt	Spočítá počet bublin v prostoru 3DM.	30% + prémie
Bludiště	proj3 --maze M.txt	Vyhledá cestu z bludiště v matici M.	30%
Kulečník	proj3 --carom coordinates direction power M.txt	Sleduje cestu kulečnickové koule po stole M. coordinates - souřadnice koule ve formátu "ř,s" direction - směr koule (S, SV, JV, ...) power - celočíselně zadaná síla úderu	30%

**Poznámka:** Program musí umět zpracovat soubory s libovolnými jmény.

Formát výstupních dat

Výstup každé operace se bude provádět na standardní výstup (stdout). Chybové zprávy se budou vypisovat na standardní chybový výstup (stderr).

Pokud je výsledkem operace matice, bude vypsána ve stejném formátu jako [vstup](#) (tedy včetně rozměrů, aby šlo použít výsledek opět jako vstup programu). Pokud operace nemá legální výsledek, vypište na standardní výstup `false`. V tomto případě nevypisujte chybové hlášení.

Maximální počet získaných bodů

Za tuto úlohu lze získat až **10 bodů**. Součástí řešení není dokumentace.

Prémie

Lze získat premiový bod za implementaci rozšíření. Například:

- viz rozšíření úlohy o počtu bublin do 3D
- další zajímavé a netriviální (smysluplné) operace s maticí(cemi): determinant, zvětšení/zmenšení rozlišení (jako u obrázků), ...
- vizualizace operací na obrazovce (možno i jako doplňkový program využívající výstup tohoto programu)
- Výjimečně dobrý návrh celé aplikace.
- ...

Co se zejména hodnotí

V této úloze se zejména hodnotí:

- Správnost implementace zvolených operací.
- Kvalita (tedy i efektivita) a správnost algoritmů pro práci s poli. Bude se zejména hlídat překročení rozsahu pole.
- Kvalita návrhu datových struktur a práce s nimi.

- Správná práce s dynamickými datovými typy.
  - Správná práce se soubory.
- 

## Pomůcka

Pracujte s dynamicky alokovanými poli. Většina chyb pak způsobí pád aplikace, což je výhodné pro ladění. U jiného typu alokace nelze ošetřit nedostatek paměti! Překročení rozsahu pole a jiné chyby lze v dynamickém poli ohlídat pomocí programu [valgrind](#).

**Tip:** Zkuste svůj program otestovat například takto:

```
$ ./proj3 --mmult m1.txt m2.txt > m12.txt
$ ./proj3 --mmult m2.txt m3.txt > m23.txt
$ ./proj3 --mmult m12.txt m23.txt > m1223.txt
```

Výstup vašeho programu musí jít bez úprav použít opět na vstupu.

---

Autor: [David Martinek](#). Poslední modifikace: 23. září 2010. Pokud v tomto dokumentu narazíte na chybu, dejte mi prosím vědět.