

**Warning**

This page is located in archive. Go to the latest version of this [course pages](#). Go the latest version of [this page](#).

## HW 06 - Maticové počty

<b>Termín odevzdání</b>	27.11.2021 23:59 PST <b>HW06B</b> 08.01.2022 23:59 PST
<b>Povinné zadání</b>	2b <a href="#">kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]</a>
<b>Volitelné zadání</b>	3b <a href="#">kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]</a>
<b>Bonusové zadání</b>	3b <a href="#">kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]</a>
<b>Počet uploadů</b>	20
<b>Podpůrné soubory</b>	<a href="#">b0b36prp-hw06.zip [/b211/_media/courses/b0b36prp/hw/b0b36prp-hw06.zip]</a> <a href="#">b0b36prp-hw06b.zip [/b211/_media/courses/b0b36prp/hw/b0b36prp-hw06b.zip]</a>

U této úlohy bude cvičící ručně hodnotit dodržení [Coding Stylu \[/b211/courses/b0b36prp/resources/tessun/start\]](#). Hodnocení je popsáno [zde \[/b211/courses/b0b36prp/hw/start\]](#).

V této úloze budete mít za úkol implementovat maticové operace sčítání, odčítání a násobení. Všechny prvky všech matic (i v průběhu výpočtu) se vejdou do 32-bitového znaménkového typu integer.

Pokud nebude vstup ve správném formátu nebo nepůjde provést příslušnou maticovou operaci, tak vypište " **Error: Chybny vstup!** " a konec řádku na standardní chybový výstup a ukončete program s návratovou hodnotou **100** . Není třeba kontrolovat bílé znaky (mezery a nové řádky); stačí tedy ověřit, že jsou na vstupu dvě celá čísla reprezentující velikost matice a následně správný počet celých čísel odpovídající velikosti matice (například pomocí `scanf()` ).<sup>1)</sup>

Pokud budete používat dynamickou alokaci paměti ( `malloc()` , `calloc()` ), tak doporučujeme použít program [Valgrind](#) (<http://valgrind.org/>) ještě před nahráním do odevzdávacího systému pro kontrolu práce s pamětí.

Velikost vstupních matic (povinného a volitelného zadání) je přesně specifikována vstupním formátem, proto je **doporučeno načítat vstup postupně po celých číslech** a nikoliv po řádcích např. funkcí `getline()` . V případě bonusového zadání se může použít funkce `getline()` zdát výhodné, ale ani tady to není nutné. Opět můžete načítat první řádek postupně po celých číslech a tím identifikovat počet sloupců. Následně můžete již alokovat potřebnou velikost paměti, např. funkcí `realloc()` a postupovat identicky s povinným/volitelným zadáním.

Pro testování funkčnosti program před jeho odevzdáním lze využít přiložené vstupní a referenční výstupní soubory. Dále je možné testovat také generátorem a referečním řešením viz [Testování HW programů před odevzdáním \[/b211/courses/b0b36prp/tutorials/testing\]](#). Pro generování volitelného zadání použijte dodatečný přepínač **-optional** .

## Povinné zadání

Na standardním vstupu jsou pouze dvě matice s jednou operací (+, -, \*). Vaším úkolem je provést zadanou operaci a vypsát na standardní výstup výslednou matici.

## Formát vstupu

Na standardním vstupu jsou dvě nebo i více matic oddělených jedním řádkem se znakem operace (+, -, \*). Každá matice má na prvním řádku nejprve svoji velikost ( $n$ ,  $m$ ) a následuje  $n$  řádků vždy s  $m$  hodnotami matice. Jednotlivé hodnoty jsou oddělené mezerami (*whitespaces*).

## Formát výstupu

Formát výstupu je stejný, ale obsahuje pouze jednu matici. Nezapomeňte, že na konci řádku není mezera a i za posledním řádkem je znak nového řádku.

### Příklad 1 - pub01-m

$$\begin{pmatrix} 76 & 98 & -31 \\ 30 & 30 & 32 \end{pmatrix} - \begin{pmatrix} 89 & 25 & 38 \\ 1 & -32 & -38 \end{pmatrix} = \begin{pmatrix} -13 & 73 & -69 \\ 29 & 62 & 70 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 2 3 76 98 -31 30 30 32 - 2 3 89 25 38 1 -32 -38 </pre>	<pre> 2 3 -13 73 -69 29 62 70 </pre>	žádný	0

### Příklad 2 - pub02-m

$$\begin{pmatrix} -59 & 78 & -85 \end{pmatrix} \times \begin{pmatrix} 78 \\ -28 \\ -97 \end{pmatrix} = \begin{pmatrix} 1459 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 1 3 -59 78 -85 * 3 1 78 -28 -97 </pre>	<pre> 1 1 1459 </pre>	žádný	0

### Příklad 3 - pub03-m

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 2 3 16 41 -98 * 3 1 96 -67 49 </pre>	žádný	Error: Chybny vstup!	100

### Příklad 4 - pub04-m

$$\begin{pmatrix} 81 & -96 & -56 & -9 \\ -19 & 66 & 37 & -21 \\ 20 & 49 & -71 & -49 \\ 45 & -96 & 20 & 8 \end{pmatrix} \times \begin{pmatrix} -89 & -96 \\ 76 & 75 \\ 65 & 2 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 4 4 81 -96 -56 -9 -19 66 37 -21 20 49 -71 -49 45 -96 20 8 * 3 2 -89 -96 76 75 65 2 </pre>	žádný	Error: Chybny vstup!	100

## Volitelné zadání

Na vstupu je sekvence matic o maximální délce 100 spolu se zadanými operacemi. Operace vyhodnocujte podle jejich priority a vypište až výslednou matici. To odpovídá tomu, jako kdyby byl následující výraz s maticemi A až F:

$$A + B * C + D * E - F$$

ozávorkován následujícím způsobem:

$$A + (B * C) + (D * E) - F$$

### Příklad 1 - pub01-o

$$\begin{pmatrix} 6 & 4 \end{pmatrix} + \begin{pmatrix} -6 & 7 \end{pmatrix} + \begin{pmatrix} -6 & -4 \end{pmatrix} = \begin{pmatrix} -6 & 7 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 1 2 6 4 + 1 2 -6 7 + 1 2 -6 -4 </pre>	<pre> 1 2 -6 7 </pre>	žádný	0

### Příklad 2 - pub02-o

$$\begin{pmatrix} 0 & 4 & -9 \\ -9 & 6 & -4 \\ 3 & 5 & -2 \\ -1 & 7 & 5 \end{pmatrix} \times \begin{pmatrix} -10 & -9 & -8 & 9 \\ -4 & 0 & -9 & 1 \\ 4 & 6 & -9 & 5 \end{pmatrix} + \begin{pmatrix} 0 & -9 & 3 & -6 \\ 10 & -9 & 8 & -7 \\ -1 & 0 & 5 & 1 \\ 3 & 2 & -9 & 9 \end{pmatrix} = \begin{pmatrix} -52 & -63 & 48 & -47 \\ 60 & 48 & 62 & -102 \\ -59 & -39 & -46 & 23 \\ 5 & 41 & -109 & 32 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre> 4 3 0 4 -9 -9 6 -4 3 5 -2 -1 7 5 * 3 4 -10 -9 -8 9 -4 0 -9 1 4 6 -9 5 + 4 4 </pre>	<pre> 4 4 -52 -63 48 -47 60 48 62 -102 -59 -39 -46 23 5 41 -109 32 </pre>	žádný	0

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
0 -9 3 -6 10 -9 8 -7 -1 0 5 1 3 2 -9 9			

### Příklad 3 - pub03-o

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
2 2 3 x10 -9 5 + 2 2 8 -5 1 8 + 2 2 -4 6 -2 8	žádný	Error: Chybny vstup!	100

### Příklad 4 - pub04-o

$$\begin{pmatrix} -1 \\ 4 \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix} + \begin{pmatrix} 8 & 5 \\ 10 & -8 \end{pmatrix} \times \begin{pmatrix} 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 63 \\ 14 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
2 1 -1 4 + 2 1 -1 0 + 2 2 8 5 10 -8 * 2 1 5 5	2 1 63 14	žádný	0

## Bonusové zadání

Na vstupu jsou matice zadány ve formátu  $název = [x_{11} \ x_{12} \ \dots \ x_{1n}; x_{21} \ x_{22} \ \dots \ x_{2n}; \dots; x_{m1} \ x_{m2} \ \dots \ x_{mm}]$  a maticová operace je poté zadána pomocí názvů jednotlivých matic.

- názvy matic jsou velká písmena, přičemž seznam matic předchází zadané operaci
- definice jedné matice zabírá jeden řádek
- členy matice se zapisují do hranatých závorek
- jednotlivé členy na řádku jsou odděleny mezerou, jednotlivé řádky jsou odděleny středníkem
- výpočet je oddělen od definic matic volným řádkem
- výpočetní operace se skládá z libovolného množství operandů.

- operandy jsou pouze +, -, \* přičemž operace se vyhodnocují podle priority stejně jako u volitelného zadání
- výstupem programu je výsledek maticové operace formátovaný podle pravidel popsaných výše
- počet matic je omezen počtem písmen v abecedě, ale délka výrazu není nijak omezena

## Příklad 1 - pub01

$$B = \begin{pmatrix} 5 & 2 & 4 \\ 0 & 2 & -1 \\ 3 & -5 & -4 \end{pmatrix}, \quad E = \begin{pmatrix} -6 & -5 & -8 \\ -1 & -1 & -10 \\ 10 & 0 & -7 \end{pmatrix}, \quad R = \begin{pmatrix} -1 & -7 & 6 \\ -2 & 9 & -4 \\ 6 & -10 & 2 \end{pmatrix}$$

$$R + E + B = \begin{pmatrix} -2 & -10 & 2 \\ -3 & 10 & -15 \\ 19 & -15 & -9 \end{pmatrix}$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
B=[5 2 4; 0 2 -1; 3 -5 -4] E=[-6 -5 -8; -1 -1 -10; 10 0 -7] R=[-1 -7 6; -2 9 -4; 6 -10 2]  R+E+B	[-2 -10 2; -3 10 -15; 19 -15 -9]	žádný	0

## Příklad 2 - pub02

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
K=[-10 0 2; -6 10 -6; -9 2 0] D=[0 6 7] M=[10 -5 -4]  D*K+M	[-89 69 -40]	žádný	0

## Příklad 3 - pub03

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
R=[6 9; -3 9; -9 10] K=[2 -8 8; -1 2 -4]  K+K*R*K	[-96 332 -384; 78 -252 312]	žádný	0

## Příklad 4 - pub04

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
M=[-9 5 9; -7 8 7; 10 -3 3]  M*M+M-M	[136 -32 -19; 77 8 14; -39 17 78]	žádný	0

...

## Příklad 10 - pub10

Poslední veřejný příklad je inspirován jedním ze způsobů výpočtu Fibonacciho čísla [[https://cs.wikipedia.org/wiki/Fibonacciho\\_posloupnost](https://cs.wikipedia.org/wiki/Fibonacciho_posloupnost)]. Výsledná matice bude obsahovat sumy prvních 24, 25<sup>2)</sup> a 26 Fibonacciho čísel.

$$\sum_{i=1}^{25} A^i = \sum_{i=1}^{25} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^i$$

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
<pre>A=[1 1; 1 0]</pre> <pre>A+A*A+A*A*A+A*A*A*A+A*A*A*A+A*A*A*A+A+A*A*A*A+A+ . . .</pre>	[317809 196417; 196417 121392]	žádný	0

## Odevzdání a hodnocení

Veřejné příklady + Makefile:

- Povinné (man) a volitelné (opt) zadání: [b0b36prp-hw06.zip](#) [[/b211/\\_media/courses/b0b36prp/hw/b0b36prp-hw06.zip](#)]
- Bonusové zadání: [b0b36prp-hw06b.zip](#) [[/b211/\\_media/courses/b0b36prp/hw/b0b36prp-hw06b.zip](#)]

	Povinné zadání	Volitelné zadání	Bonusové zadání
Název v BRUTE	HW06		HW06B
Odevzdávané soubory	main.c		
Argumenty při spuštění	žádné		
Návratová hodnota	0 100 ; "Error: Chybny vstup!" → stderr		
Kompilace pomocí	clang -pedantic -Wall -Werror -std=c99		
Procvičované oblasti	pole variabilní délky indexování v poli	dynamická alokace paměti	dynamická alokace paměti práce s pointry

<sup>1)</sup>

Upřesněno 24. 11. 2017 na základě studentského dotazu.

<sup>2)</sup>

Možno ověřit na [WolframAlpha](http://www.wolframalpha.com/input/?i=sum+Fibonacci%5Bx%5D+from+1+to+25) [<http://www.wolframalpha.com/input/?i=sum+Fibonacci%5Bx%5D+from+1+to+25>]

[courses/b0b36prp/hw/hw06.txt](#) · Last modified: 2021/08/23 09:39 by faiglj