

Milý řešiteli,

vítáme Tě u 1. série úloh 5. ročníku korespondenčního semináře MoRoUS. Jako v minulých letech i nyní Ti přinášíme úlohy ze světa profesora Morouse a jeho robotických společníků.

Řešení úloh Ti umožní nejen nahlédnout do zajímavých zákoutí umělé inteligence a robotiky, ale také zúčastnit se jarního soustředění s inspirativními lidmi, obzor rozšiřujícími přednáškami a nezapomenutelnými zážitky ;) A na nejlepší řešitele čekají na konci soutěže tematické ceny!

# 1. série 2018/2019

Termín odeslání 1. série: **9. 12. 2018**

## Kam posílat řešení?

Až budeš mít řešení hotové, pošli nám je včetně všech nákresů, programků a prostě všeho co by nám usnadnilo opravování Tvé úlohy. Stačí, když pošleš řešení jen některých úloh nebo jejich částí, i k těm Ti pošleme komentáře a přičteme za ně body ;)

Řešení posílej nejlépe e-mailem na adresu `seminar@morous.fel.cvut.cz`, nebo poštou (řešení každé úlohy v tomto případě napiš na samostatný papír A4) na adresu

Korespondenční seminář Morous,  
Katedra kybernetiky FEL ČVUT,  
Karlovo náměstí 13,  
121 35 Praha

## Úloha č. 1: Skladiště (20 bodů)

Robotí skladiště potravin je pravoúhlá síť chodeb, ve které se nacházejí kromě obvyklých kovových pilin a vazelíny i různé laskominy, jež jsou předmětem touhy některých méně zodpovědných robotů. Proto jsou na každé křižovatce instalována pohybová čidla, z jejichž záznamu lze zjistit, ve kterém časovém okamžiku byl zachycen pohyb. Bohužel, čidla byla nakoupena ve slevové akci a některá se až o pět sekund předcházejí/zpožďují. Neví se ale, která to jsou.

V sektoru  $B3$  (označen křížkem) došlo v čase  $t_0$  k vloupání do krabice s vanilkovými šroubky. Pachatel (robot T2D1) byl posléze dopaden v sektoru  $E5$  (označen hvězdičkou) v čase  $t_d = 9$ , nicméně bez šroubků.

Víme, že T2D1 se pohybuje konstantní rychlostí 1 sektor za sekundu a že se nikdy nevrátí do stejného sektoru, ve kterém už byl.

### Úkol 1.1

Kudy se T2D1 mohl pohybovat, a které chodby tedy bude potřeba prohledat, zda tam laskominy neukryl?

	1	2	3	4	5	
A	.	.	.	.	.	$t_d = 9s$
B	.	2	×	.	.	
C	.	-1	2	.	.	
D	.	.	4	7	8	
E	.	.	5	6	★	

*Pozn.: Čísla v tabulce značí, ve kterých okamžicích čidlo na daných souřadnicích zachytilo pohyb. Časy jsou uvedeny v sekundách relativně k času vloupání.*

## Úkol 1.2

Jak bude vypadat obecný návod pro nalezení všech tras, kudy se mohl T2D1 pohybovat, máš-li časové signály z čidel?

Tvůj návod by měl úspěšně vyřešit například tento případ záznamu:

	1	2	3	4	5	
A	.	.	.	.	.	$t_d = 10s$
B	.	×	0	5	.	
C	.	2	5	3	.	
D	.	2	4	9	★	
E	.	.	.	.	.	

Kolik je v tomto případě možných tras, po kterých se T2D1 mohl pohybovat?

*Pozn.: Předpokládej, že místo a čas loupeže a místo a čas dopadení je znám vždy, čidla mohou (jako v předchozím případě) chybovat.*

## Úkol 1.3

Po této závažné události byl povolán opravář. Nyní se žádná čidla nepředcházejí ani nezpožďují, některá ale nefungují vůbec. Jak mohla v tomto případě vypadat trasa, po které se pachatel pohyboval? T2D1 byl dopaden 8 sekund po loupeži.

	1	2	3	4	5	
A	.	.	.	.	.	$t_d = 8s$
B	.	.	×	.	.	
C	.	2	.	.	.	
D	.	.	.	.	.	
E	.	.	5	★	.	

## Úkol 1.4

Jak bude vypadat obecný návod pro nalezení všech míst, kudy se mohl T2D1 pohybovat, máš-li časové signály z čidel?

Tvůj návod by měl fungovat pro tento:

	1	2	3	4	5	
A	×	1	.	.	.	$t_d = 6s$
B	.	2	.	.	.	
C	.	3	.	.	.	
D	.	4	5	.	.	
E	.	.	*	.	.	

i tento případ:

	1	2	3	4	5	
A	.	.	.	.	.	$t_d = 5s$
B	.	×	.	.	.	
C	.	.	.	.	.	
D	.	.	*	.	.	
E	.	.	.	.	.	

Kolik budou mít tato zadání možných řešení? Může se stát, že zadání nebude mít řešení?

*Pozn.: Místo a čas loupeže a místo a čas dopadení je opět znám vždy, čidla mohou (jako v předchozím případě) vynechávat.*

## Úkol 1.5

Nikdo nic neopraví stoprocentně. Zjistilo se, že i po zásahu opraváře se některá čidla zpoždují/předcházejí a k tomu některá zcela vynechávají. Jak v tomto případě zjistit, kudy mohl T2D1 jít?

Opět chceme obecný návod, který vyřeší všechny možné situace, např.

	1	2	3	4	5	
A	×	.	.	.	.	$t_d = 12s$
B	.	.	2	.	.	
C	.	.	.	.	.	
D	.	3	.	.	.	
E	.	.	.	.	★	

Jakými způsoby lze co nejdříve odhalit neplatné řešení (např. nemá smysl hledat trasy větší délky než 12)? Nad touto otázkou se opravdu zamysli. Skladiště může být mnohem větší a i když mají roboti k dispozici počítače s opravdu velkou výpočetní kapacitou, nemuseli by se výsledku dočkat.





## Pokročilé příkazy

Pro počítač existuje ale mnohem více příkazů a je také připojen k numerické klávesnici. Toho hned můžeme využít.

- Protože takto skládat čísla je zdlouhavé, počítač obsahuje příkaz INPUT (IN). Ten program pozastaví a počká, než uživatel zadá na klávesnici číslo a stiskne „enter“. Příkaz pak toto číslo napíše do  $U_1$ .
- Další příkaz SUBTRACT (S) slouží na odčítání čísel. Provedení vypadá takto:  $U_1 := U_1 - U_2$
- Příkaz Go (G) nám umožňuje pohybovat se po  $\mathcal{PP}$ . Příkaz má opět parametr „kam“ (celé kladné číslo), který značí na kolikáté políčko  $\mathcal{PP}$  má program skočit. Program se pak začne vykonávat od daného políčka. Toto nám silně rozšiřuje obor možností. Vezměte v úvahu následující příklad:

$\mathcal{PP}$ 

A	P	G 1
---	---	-----

Program se vykonává zleva doprava. Vždy, když narazí na příkaz Go 1, skočí program na první příkaz a klasicky bude dále pokračovat zleva doprava. Kouzelný počítač tak bude do nekonečna tisknout zvyšující se mocniny dvojky.

- Poslední příkaz, který budete potřebovat, je příkaz IF. Pokud  $U_1$  je kterékoliv číslo kromě 0, příkaz neudělá nic. Program ho pouze přečte a bude pokračovat vykonáváním dalšího příkazu (příkazu, který se na  $\mathcal{PP}$  nachází vpravo od IF). Pokud  $U_1$  je 0, program ve čtení  $\mathcal{PP}$  přeskočí první příkaz za IF a vykoná až druhý.

$\mathcal{PP}$ 

IN	IF	G 1	P
----	----	-----	---

Program nechá uživatele zadat číslo. To se zapíše do  $U_1$ . Pokud uživatel napsal 0, program přeskočí příkaz Go 1, přečte rovnou příkaz P, který vypíše 0. Program skončí.

Pokud uživatel zadal jiné číslo, program přečte příkaz Go 1 a program udělá skok na první pole  $\mathcal{PP}$ .



## Úkol 2.2

S využitím nových příkazů, napište program, který nechá uživatele zadat číslo  $n$  (předpokládejme, že uživatel zadá kladné číslo). Pak postupně vytiskněte čísla  $2, 4, 8, \dots, 2^n$ . Program pak musí skončit.

*Pozn.: V tomto případě nezáleží, kolik příkazů vaše PP obsahuje.*

## Úkol 2.3

Nechte uživatele zadat dvě čísla (předpokládejme, že uživatel zadá dvě kladná čísla). Vytiskněte větší z nich. Pokud jsou stejná, vytiskněte obě.

*Pozn.: Opět nezáleží kolik příkazů vaše PP obsahuje.*

## Způsob zápisu

Každý program zapisujte do samostatného textového souboru. Pořadí řádků odpovídá pořadí příkazů na PP.

Příkaz	Zápis	
ADD	A	
SUBTRACT	S	
PRINT	P	
INPUT	IN	
MOVE	M $x$	$x$ je celé kladné číslo
GO	G $x$	$x$ je celé kladné číslo
IF	IF	

## Témátko

Témátka můžeš odesílat v průběhu celého roku. Je jen na Tobě, jestli k němu napíšeš program, nakreslíš obrázkové řešení, vyrobíš řešení v reálu, či jen popíšeš své myšlenky. Pokus se ale vždy přijít s nějakým svým nápadem a dobře ho zdůvodni.

### Témátko č. 1: Konec zdlouhavému psaní zpráv? (20 bodů)

Co kdyby počítače „uměly číst myšlenky“ a napsaly zprávu za Tebe? Zamysli se nad tím, jak co nejlépe vytvořit systém, který bude uživateli nabízet slova v průběhu psaní zprávy. Jaké informace a jakým způsobem využiješ? Jak se přizpůsobit konkrétnímu uživateli?