

Round 1

`void pick_all()`

Funkcia `void pick_all()` (1b)

Vytvorte funkciu `void pick_all()`, pomocou ktorej robot Karel pozbiera všetky značky na danej pozícii.

Return

Funkcia nevráti žiadnu hodnotu.

`void find_south_east()`

Funkcia `void find_south_east()` (2b)

Vytvorte funkciu `void find_south_east()`, ktorá premiestni robota Karla do pravého dolného rohu sveta

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Na začiatku sa robot Karel nachádza na ľubovoľnej pozícii vo svete.

Koncová situácia

Na konci sa robot Karel nachádza na juhovýchode, teda v pravom dolnom rohu sveta. Je otočený smerom na západ.

`bool facing_vertical()`

Funkcia `facing_vertical()` (2b)

Vytvorte funkciu `bool facing_vertical()`, ktorá zistí, či robot Karel má vertikálny smer (sever alebo juh).

Return

Funkcia vráti hodnotu `true`, ak robot smeruje na sever alebo juh, v opačnom prípade funkcia vráti hodnotu `false`.

void climb_stairs()

Funkcia `void climb_stairs()` (4b)

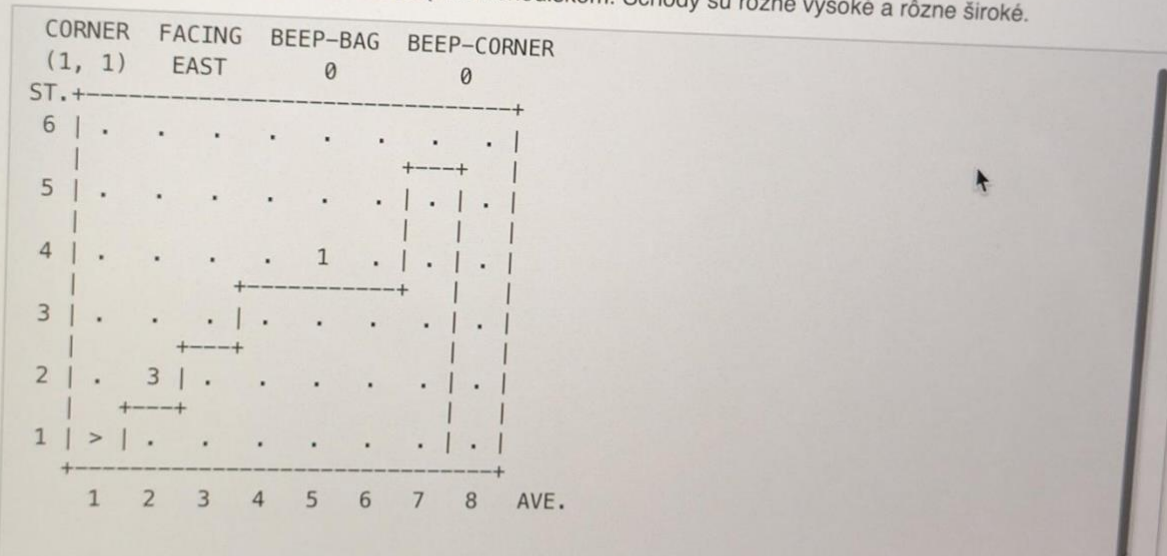
Vytvorte funkciu `void climb_stairs()`, pomocou ktorej sa robot Karel premiestni vrchol schodiska. Cestou pozbiera všetky značky a položí ich na posledný schod. Schody sú rôzne vysoké a rôzne široké.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Na začiatku sa robot Karel nachádza pred schodiskom. Schody sú rôzne vysoké a rôzne široké.



Koncová situácia

Na konci sa robot Karel aj všetky značky nachádzajú na vrchole schodiska. Schody sú rôzne vysoké a rôzne široké.

void olympics()

Funkcia `olympics()` (3b)

Vytvorte funkciu `void olympics()`, pomocou ktorej robot Karel preskočí ľubovoľne vysoké prekážky. Robot sa zastaví na značke.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Robot Karel stojí na začiatku prekážkovej trate.

CORNER	FACING	BEEP-BAG	BEEP-CORNER
(1, 1)	EAST	10	0

ST.	+-----+										
6			
5			
4			
3			
2				
1		>		.		.		.			
				.		.		.			
	+-----+										
	1	2	3	4	5	6	7	8	9	10	AVE.

void chessboard()

Funkcia `chessboard()` (4b)

Vytvorte funkciu `void chessboard()`, pomocou ktorej Karel dokáže vytvoriť šachovnicu ľubovoľnej veľkosti.

Čierne políčko je bez značky Biele políčko obsahuje značku. Karel bude začínať v ľavom dolnom rohu, kde je vždy políčko čierne.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Karel začína v ľavom dolnom rohu sveta (čierne políčko).

CORNER	FACING	BEEP-BAG	BEEP-CORNER	
(5, 5)	NORTH	100	0	
ST.	+-----+			
5			
4			
3			
2			
1	> . . .			
	+-----+			
	1 2 3 4 5	AVE.		

void x_mas()

Funkcia `x_mas()` (4b)

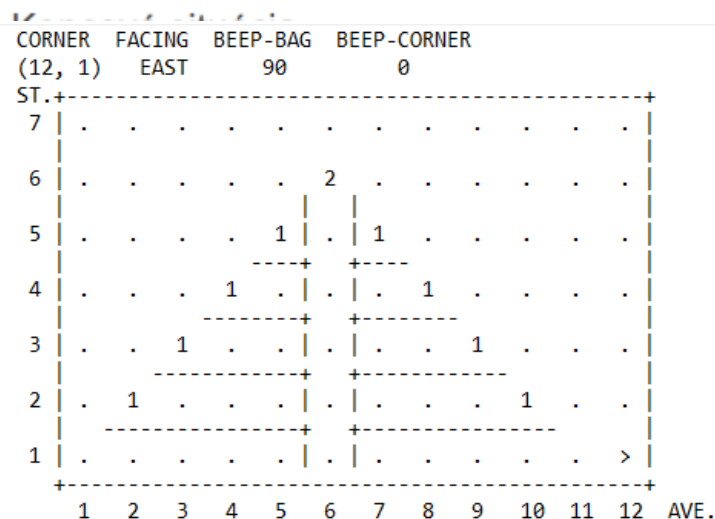
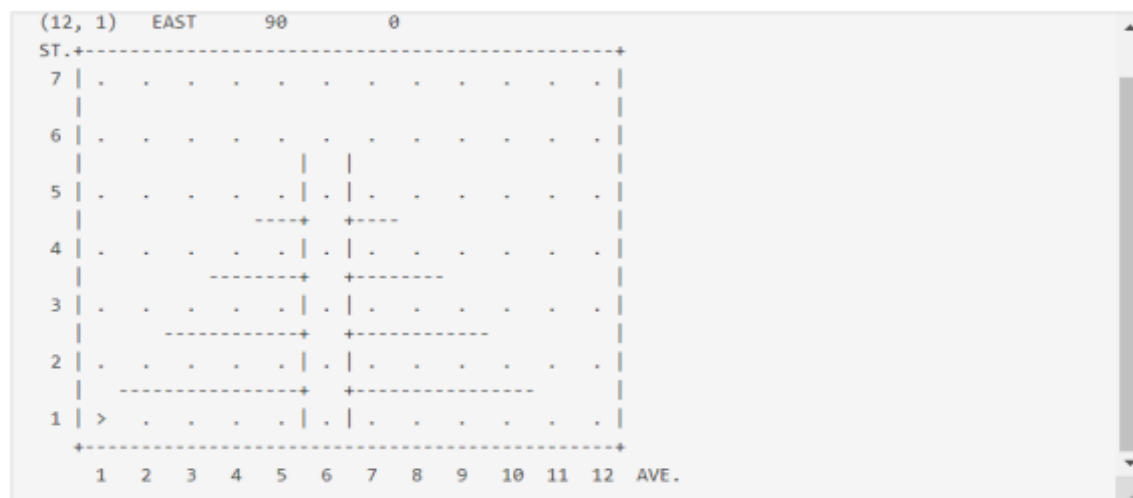
Vytvorte funkciu `void x_mas()`, pomocou ktorej pomôžete Karlovi ozdobiť vianočný strom, ak strom je už postavený. Ozdoby (značky) nech sa nachádzajú na kraji vetiev stromu a vrchol stromu nech zdobí dvojnásobná značka.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Karel začína v juhozápadnom rohu sveta.



void divide_even_beepers()

Funkcia `void divide_even_beepers()` (3b)

Vytvorte funkciu `void divide_even_beepers()`, pomocou ktorej robot Karel rozdelí kopu značiek na dve rovnaké polovice. Počet značiek je vždy párny.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Na začiatku sa robot Karel nachádza pred kopou značiek. Pred robotom sa nachádzajú aspoň dve pozície, na ktoré je možné rozdeliť kopu značiek. Batoh robota je prázdny.

CORNER	FACING	BEEP-BAG	BEEP-CORNER
(1, 2)	EAST	0	0
ST. +-----+			
3	. . .		
2	> 10 .		
1	. . .		
+-----+			
	1 2 3 AVE.		

Koncová situácia

Na konci sa robot Karel nachádza na svojej pôvodnej pozícii, otočený smerom na východ. Kopa značiek je rozdelená na dve polovice. Prvá polovica značiek sa nachádza na svojej pôvodnej pozícii a druhá o jednu pozíciu napravo.

CORNER	FACING	BEEP-BAG	BEEP-CORNER
(1, 2)	EAST	0	0
ST. +-----+			
3	. . .		
2	> 5 5		
1	. . .		
+-----+			
	1 2 3 AVE.		

`bool no_beepers()`

`void find_center()`

Funkcia `no_beepers()` (2b)

Vytvorte funkciu `bool no_beepers()`, ktorá zistí, či aktuálna pozícia sveta aj batoh sú prázdne.

Return

Funkcia vráti hodnotu `true`, ak sú aktuálna pozícia sveta aj batoh prázdne, v opačnom prípade funkcia vráti hodnotu `false`.

Funkcia `void find_center()` (3b)

Vytvorte funkciu `void find_center()`, pomocou ktorej robot Karel nájde stred sveta.

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Na začiatku sa robot Karel nachádza na ľubovoľnej pozícii. V batohu má dostatočný počet značiek (ak ich potrebuje na pomoc).

CORNER (1, 2)	FACING EAST	BEEP-BAG 10	BEEP-CORNER 0
ST. +-----+			
3 . . .			
2 > . .			
1 . . .			
+-----+			
1 2 3	AVE.		

`void turn_to_south()`

`void mark_empty_line()`

Funkcia `void turn_to_south()` (1b)

Vytvorte funkciu `void turn_to_south()` , pomocou ktorej sa robot Karel otočí na juh.

Return

Funkcia nevráti žiadnu hodnotu.

Funkcia `void mark_empty_line()` (2b)

Vytvorte funkciu `void mark_empty_line()` , ktorá označuje jeden celý riadok alebo stĺpec značkami (podľa smeru natočenia).

Return

Funkcia nevráti žiadnu hodnotu.

Počiatočná situácia

Na začiatku sa robot Karel nachádza na ľubovoľnej pozícii vo svete.

Koncová situácia

Podľa smeru robota bude na konci označovaný celý riadok alebo celý stĺpec značkami. Na pozícii robota nezáleží.