

NETROM Code Golf

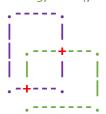
Ediția a VII-a

Problema 1: Mașini inteligente

Să se scrie o funcție care simulează mișcarea în plan a unor mașini inteligente ce se deplasează autonom cu o viteză constantă pe o traiectorie dată de un dreptunghi. Dându-se o configurație inițială, funcția trebuie să deseneze în consolă configurația finală a simulării după ce mașinile se deplasează independent, dar respectând regula de prioritate de dreapta, pe o durată de timp t.

Numim o mașină inteligentă deoarece ea se poate deplasa singură, cu viteză constantă, pe o traiectorie specificată în plan cartezian (cu punctul O in colțul stânga-sus, axa Ox spre dreapta și axa Oy în jos) printr-un dreptunghi de lățime w (pe axa Ox) si înălțime h (pe axa Oy), plecând dintr-un punct inițial fix de coordonate (x, y) reprezentând colțul stânga-sus al acestui dreptunghi, având direcția de deplasare dată ca fiind fie în sens orar, fie în sens trigonometric și poate opri pentru a acorda prioritate de dreapta la intersecția traiectoriei sale cu a altei mașini inteligente. Mașinile se deplasează circular doar pe laturile dreptunghiului, în sensul de deplasare stabilit inițial, cu viteza constantă de o poziție într-o unitate de timp. În momentul în care două mașini inteligente sesizează că urmează să se întâlnească la același moment de timp în unul dintre punctele de intersecție ale traiectoriilor lor, mașina din stânga va opri fix în poziția dinaintea acestuia și va aștepta un moment de timp (va întârzia o unitate de timp) ca mașina din dreapta să treacă prin acel punct de intersecție pentru a evita o eventuală coliziune, după care va reporni și își va continua drumul.

Traiectoria mașinilor se desenează folosind caracterele '|` și '-' (pentru laturi), '.' (pentru vârfurile dreptunghiurilor), iar pozitia mașinii, cu caracterul 'X'. Se vor desena spații în zonele libere. Când 2 traiectorii se intersectează, caracterul '+' se va desena în locul caracterelor pentru laturi (mașinile se vor intersecta doar pe laturi, nu și în colțuri). O traiectorie de lățime w și înălțime h va avea o reprezentare grafică de w caractere lungime pentru linii și h caractere înălțime pentru coloane (sunt incluse vârfurile). Mașinile se vor plimba pe toate caracterele traiectoriei ('|', '-', '+' și '.'). Spre exemplu, dacă am avea 2 mașini cu w1 = 7, h1 = 5, respectiv w2 = 9, h2 = 4, care se intersectează în 2 puncte, reprezentarea va fi ca în desenul de mai jos:



Prototip funcție:

void f(unsigned char n, unsigned char p[][5], unsigned char t)

unde:

- n: reprezintă numărul de mașini inteligente;
- p: reprezintă lista de masini inteligente, pentru fiecare masina fiind specificate următoarele:
 - o p[][o]: coordonata x a poziției de start a mașinii (colțul stânga-sus al dreptunghiului traiectoriei sale);
 - o p[][1]: coordonata y a poziției de start a mașinii (colțul stânga-sus al dreptunghiului traiectoriei sale);

- o p[][2]: lățimea dreptunghiului traiectoriei mașinii (pe axa Ox);
- o p[][3]: înălțimea dreptunghiului traiectoriei mașinii (pe axa Oy);
- o p[][4]: sensul/direcția de deplasare a mașinii având una din valorile:
 - o = în sens orar (pe direcția dreapta-jos-stânga-sus)
 - 1 = în sens trigonometric (pe direcţia jos-dreapta-sus-stânga)
- t: reprezintă momentul de timp la care se oprește simularea și se dorește reprezentarea grafică a stării finale a masinilor

Observatii:

- În cadrul unui ciclu de testare, funcția se va apela o singură dată;
- Fiecare maşină ce urmează o traiectorie dreptunghiulară având punctul de start (x,y) va porni (la momentul de timp t = 0) din punctul (x, y).
- Dreptunghiurile traiectoriilor mașinilor sunt distincte două câte două și fiecare traiectorie se poate intersecta cu cel mult o altă traiectorie, intersecția realizându-se numai în puncte de pe laturile dreptunghiurilor (2 sau 4).
- Se garantează că nu vor exista în plan dreptunghiuri ce vor avea în comun o latură sau o porțiune de latură;
- Se garantează că valorile de intrare sunt întotdeauna corecte (p va avea mereu lungimea n, p[][4] va avea valori numai în setul <0, 1>).
- Coordonatele (x, y) pot avea valori între o și 255. Lățimea și înălțimea pot avea valori între 3 și 255.

Exemplu:

— > Ox

Să considerăm cazul a 2 mașini cu următoarele traiectorii (sensul mașinilor: 1 – orar, 2 – trigonometric):

Explicație:

||| .+.

Simulare:

```
t = 1
.X---.
   .+.
| X||
t = 2
.-X--.
  .+.
  X+.
t = 3
.--X-.
.х.
t = 4
.---X.
   .+.
  | .+X
t = 5
.---X
  .+.
  ||X
  .+.
t = 6 // mașinile se vor întâlni la t + 1 în punctul de coordonate (6, 3)
  Χ
  .+X
  t = 7
   .XX
  .+.
```

Problema 2: Snake

Scrie o funcție care simulează un joc Snake, desfășurat pe o suprafață de joc de dimensiune 20x20, ce pornește de la o configurație inițială dată și în care mișcările pe suprafața de joc sunt date ca input funcției, prin apelarea ei secvențială.

Jocul de Snake presupune existența unei suprafațe de joc reprezentată ca o matrice, în ale cărei celule sunt amplasate un număr variabil de puncte, precum și un număr variabil de obstacole. Jocul începe cu un șarpe de dimensiunea unei singure celule, plasat în colțul din stânga sus al suprafeței de joc (1x1) și continuă prin deplasarea șarpelui sus-jos/dreapta-stânga, până când, fie șarpele ia toate punctele, fie se lovește de unul din obstacole, de marginea tablei sau de el însuși, cazuri în care jucătorul pierde.

Prototip funcție:

void f(char d)

unde:

- d: reprezintă directia curentă (o: sus, 1: dreapta, 2: jos, 3: stânga)



Starea inițială a suprafeței de joc este dată într-un set de variabile globale, după cum urmează:

- unsigned char p[400][2]: poziția punctelor pe suprafața de joc (p[][0] linia, p[][1] coloana)
- unsigned char o[400][2]: poziția obstacolelor pe suprafața de joc (o[][0] linia, o[][1] coloana)
- unsigned char np: numărul total de puncte
- unsigned char no: numărul total de obstacole

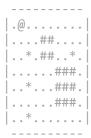
Funcția trebuie să afișeze în consolă suprafața curentă de joc, dupa cum urmează:

- '-' pentru marginile de sus și de jos, respectiv \' pentru marginile laterale
- '@' pentru corpul şarpelui
- ,*' pentru punct
- ,#' pentru obstacole
- ,.' pentru celule libere

În cazul în care jocul s-a terminat, se afisează în plus (pe o linie nouă) mesajul: 'Ai pierdut! Ai strans:

<n> puncte.' sau 'Ai castigat! Ai strans: <n> puncte.'

Exemplu:



Observatii:

- Atunci când în urma deplasării, dată de direcția curentă, șarpele ajunge într-o celulă ce conține un punct, punctul dispare și șarpele crește în dimensiune cu o celulă, celula cu punctul devenind parte din corpul sarpelui
- Atunci când în urma deplasării, dată de direcția curentă, șarpele ajunge într-o celulă ce conține un obstacol sau o altă celulă din corpul lui sau iese din marginile suprafeței de joc, jocul se termină
- În cadrul unui ciclu de testare, funcția se va apela în mod repetat. Dacă funcția este apelată după terminarea jocului, aceasta va trebui să afișeze din nou ultimul mesaj.

- Funcția se poate folosi de o variabilă globală (s) de tipul unsigned long, ce va fi inițializată cu o (zero) înainte de primul apel al funcției din cadrul unui ciclu de testare. (Atenție: În cadrul unei execuții a programului pot exista mai multe cicluri de testare)
- funcția f se apelează în mod repetat, jocul fiind resetat la configurația inițială de fiecare dată când variabila globală s are valoarea o
- pozițiile obstacolelor precum și pozițiile punctelor pleacă de la 1 (colțul din stânga sus are coordonatele 1x1)
- nu există niciun obstacol și niciun punct la poziția 1x1
- poziția punctelor și a obstacolelor se garantează a fi corectă (în interiorul sprafeței de joc fără să se suprapună)

Exemplu: Se dă o suprafață de joc cu 5 puncte la coordonatele următoare: p[o][o]=3;p[o][1]=3 p[1][0]=3;p[1][1]=9 p[2][0]=5;p[2][1]=6 p[3][0]=8;p[3][1]=3p[4][0]=10;p[4][1]=3 și 4 obstacole la coordonatele următoare: 0[0][0]=2;0[0][1]=5 0[1][0]=2;0[1][1]=6 0[2][0]=3;0[2][1]=5 0[3][0]=3;0[3][1]=6 Funcția f se apelează consecutiv cu următorii parametri: f(1) f(1) f(1) f(1) f(1) f(1) f(1) f(1) f(2) f(2)outputul va fi următorul:

l	•	(a)	۰			۰			۰			۰	٠	l
				#	#									
			*	#	#		*							
					*									
														I
														ĺ
			*											ĺ
														I
l			*											ĺ
l														ĺ
														I
ĺ														ĺ
ĺ														ĺ
														ĺ
														ĺ

0

۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰			۰	۰	۰	•	۰	٠
۰	۰	۰	۰	۰	۰	۰	•	۰	۰	۰		•		۰	۰	۰	۰	۰	۰
_	_	_	_	_	_	_	_	_	_	_		_	_	_	_	_	_	_	_
_	-	_	-	-	-	-	-	-	_	-			-	-	-	-	-	-	_
۰	۰	@		•	•	۰	۰	۰	۰	۰			۰	۰	۰	۰	۰	۰	۰
٠	۰	٠	۰	#	#	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
٠	۰	*	۰	#	#	•	•	*	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	•
					*														
٠			٠					٠	٠	٠		٠		٠	٠	٠		٠	
		*																	
٠	•		٠	•	٠	٠	•	٠	٠	٠	•	٠	•	٠	٠	٠	٠	٠	٠
٠	٠	*	٠	٠	٠	٠	٠	٠	٠	۰	٠	٠	٠	٠	۰	٠	٠	•	٠
۰	۰		۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	٠	۰	۰	۰	٠	۰	۰	۰	۰	۰	٠	۰	٠	۰	۰	۰	٠	۰	٠
۰	٠	۰	۰	٠	•	۰	•	۰	۰	۰		۰		۰	۰	۰		۰	٠
٠		•								•						•			
٠			٠						٠	٠		٠			٠	٠		٠	
٠										٠						٠			
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
_		_	_	_		_	_	_	_	_	_	_	_	_	_		_	_	_
_			_		_		_	_		_	_	_	_	_	_	_	_	_	_
۰	۰	٠	_		•	۰	۰	۰	۰	۰			۰	۰	۰	۰	۰	۰	٠
۰	۰	0 Na	۰	#	•	۰	۰	0 Na	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰	*	۰	#	#	۰	۰	*	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰	۰	۰	٠	٠	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
٠			۰		*	•			•	•	•	۰	•		•	•	•	۰	•
		*																	
٠			٠						٠	٠		٠			٠				
٠		*								٠						٠			
٠	٠	٠	۰	٠	•	٠	•	•	۰	۰	٠	•	٠	•	۰	۰	٠	•	•
٠	٠	۰	۰	۰	٠	۰		۰	۰		٠	۰		۰	۰	۰	٠	۰	٠
۰	۰	۰	۰	۰	۰	۰	٠	۰	۰		٠	۰	٠	۰	۰	۰	٠	۰	٠
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰																		
		۰	۰	۰	۰	۰	•	۰	۰	۰	۰	۰	۰	Ť	۰	۰	۰	۰	٠
۰	٠				•				•	•	•	•			•	•	•	•	•
					•				•		•	•					•	•	•
												•						•	•
		•			•		•	•		•		•				•		•	•
•		•	• • • • • •	•	•	• • • • -		• • • • • •	• • • • -	•		•			• • • • • •	•	•	•	•
•	•	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • • • • • • • • • • • • • • • • • •	• • • •	• • •	• • • •	•	• • • •	• • • • • • • • • • • • • • • • • • • •	•
• • • •	• • • •	• • • • •	• • • • •	· · · · @	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • • •	• • • •	• • • •	• • • • •
• • • • •	• • • • • •	• • • • • •		_	#	• • • • • •		• • • • • •	• • • • • •	-		• • • • • •	• • • • • •		• • • • • •	• • • • • •	• • • • • •	• • • • • •	• • • • • •
• • • • • •	• • • • •	• • • • • • *		#	#		• • • • • • •	• • • • • • *	• • • • • • •	-						• • • • • • •	• • • • •	• • • • • • •	• • • • • • •
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •			#			• • • • • • •		• • • • • • •	-						• • • • • • • • •	• • • • • • •	• • • • • • •	• • • • • • •
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •			#	#		• • • • • • • •		• • • • • • • •	-						• • • • • • • • • • • • • • • • • • • •	• • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •			#	#		• • • • • • • • •		• • • • • • • •	-						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •			#	#		• • • • • • • • • •		• • • • • • • • • •	-						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •		*		#	#		• • • • • • • • • • •			-						• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •				#	#					-						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •		* * * * *		#	#					-						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •		*		#	#					-						• • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •

	۰	۰	٠	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠	۰	•
	_	_		_	_		_	_		_		_	_	_	_		_	_
	٠	۰	٠	٠	۰	٠	۰	۰	۰	٠	۰	٠	٠	۰	۰	٠	۰	•
• •	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
					•				•	•								
		٠						٠			٠			٠	٠			
- •	,	•	-	•	,	,	•	,	•	•	,	,	,	,	,	•	•	,
• •	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
	۰	۰	٠	٠	۰	۰	۰	۰	۰	•	۰	۰	۰	۰	۰	٠	۰	٠
	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
				(a)														
			#	#														
	*			#			*											
• •		۰	π	π	۰	۰		۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
	۰	۰	٠	٠	۰	۰	۰	۰	۰	•	۰	۰	۰	۰	۰	٠	۰	٠
				*														
• •	۰	۰	٠	٠	۰	٠	۰	۰	۰	٠	۰	٠	۰	۰	۰	٠	۰	•
• •	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
	*																	
	_	_		_	#	#	#		_	_	_	_	_	_	_		_	_
	*	۰	٠	٠				۰	۰	٠	۰	۰	٠	۰	۰	•	۰	۰
• •	- 10	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
					•													
		٠						٠			٠			٠	٠			
	,		-	-	_	,			-			,	,					
• •	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠	۰	٠
• •	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	٠	۰	۰
• •	۰	۰	٠	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
• •	۰	۰	٠	٠	٠	٠	٠	۰	۰	٠	۰	•	•	۰	۰	۰	۰	٠
	•	•	•		•			•	•		•	•	•	•	•			
· ·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
· · · · · · · · · · · · · · · · · · ·	· -	-	· - -	•	· -	· -	· -	•	•		•	•	•	· -	•	•	•	•
· · · · · · · · · · · · · · · · · · ·	• - -	· - -	· -	•		· - -	·	•	• - -	· - -	· - -	· - -	· - -	· - -	· - -	· - -	· - -	• - -
· · · · · · · · · · · · · · · · · · ·	· ·	· - -	· · ·		- - @	· - -	· - -	· - -	· - -	· ·	·	· ·	·	· - -	·	· - -	·	· - -
· · · · · · · · · · · · · · · · · · ·	• •			· : #	_	•	· · ·	• • •	· · ·	· · ·	· · ·	• •	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·
· · · · · · · · · · · · · · · · · · ·	• • *			#		• • • •	• • • *	• • • •				• • •	• • • •	• • •	• • •	• • • • •	• • • •	• • •
· · · · · · · · · · · · · · · · · · ·	· · *		#	#		• • • •		• • • •				• • • •	• • • •	• • • •	• • • •	· · · ·	· · · ·	· · · ·
	• • • • * •		#	#		• • • • •		• • • • • • • • • • • • • • • • • • • •				· · · · ·	• • • • • • • • • • • • • • • • • • • •	· · · · ·	· · · · · · · · · · · · · · · · · · ·		· · · · ·	• • • • • • • • • • • • • • • • • • • •
	•		#	#		• • • • • • •		• • • • • • • • •				• • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •	•		#	#		• • • • • • •		• • • • • • •					• • • • • • • •	• • • • • • •	• • • • • • •	• • • • • • •		• • • • • • • •
••••••••••••••	•		#	#		• • • • • • • •		• • • • • • •					• • • • • • • •	• • • • • • • •	• • • • • • • •	• • • • • • •	• • • • • • •	• • • • • • • • • • • • • • • • • • • •
			#	#		• • • • • • • •		• • • • • • • • •					• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • •	• • • • • • • • •	• • • • • • • •	• • • • • • • • • • • • • • • • • • • •
	•		#	#		• • • • • • • • •		• • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
			#	#				• • • • • • • • • •					• • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • •		• • • • • • • • • • • • • • • • • •
			#	#		• • • • • • • • • •		• • • • • • • • • •					• • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
	• • *		#	#		• • • • • • • • • • • •		• • • • • • • • • • •					• • • • • • • • • • •	• • • • • • • • • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
	• • *		#	#				• • • • • • • • • • •					• • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
	• • *		#	#														• • • • • • • • • • • • • • • • • •
	• • *		#	#				• • • • • • • • • • • •					• • • • • • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • •		
	• • *		#	#														• • • • • • • • • • • • • • • • • • • •
	• • *		#	#														• • • • • • • • • • • • • • • • • • • •
	• • *		#	#														
	• • *		#	#														• • • • • • • • • • • • • • • • • •
	• • *		#	#														• • • • • • • • • • • • • • • • • •
	• • *		#	#														
	• • *		#	#														
	• • *		#	#														
	• • *		#	#														
	• • *		#	#														
	• • *		#	#														
	• • *		#	#														
	• • *		##	## • * • • • • • • • • • • •			*											
	• • *		##	## • * • • • • • • • • • • • • • • • • •			*											
	• • *		##	## • * • • • • • • • • • • •			*											
	• • *		##	## •* • • • • • • • • • • • • #			*	· · · · · · · · · · · · · · · · · ·										
			##	## •* • • • • • • • • • • • • #			*											

					*														
۰	۰	۰	۰	۰	т	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
		٠	٠			٠			٠	٠		٠		٠	٠		٠		
		*																	
۰	۰		۰	٠	۰	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	•
۰	۰	۰	۰	۰	٠	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
		*																	
۰	۰	۰	۰	٠	٠	۰	٠	۰	۰	۰	٠	۰	٠	٠	۰	۰	٠	۰	•
۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
٠	•	٠	٠	٠	٠	٠	٠	٠	٠	٠	•	٠	٠	٠	٠	٠	•	٠	•
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	•	۰												۰		۰		۰	
			٠																
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	•	٠	٠	۰	۰	٠	۰	۰	٠	۰	•	۰	۰	۰	٠	۰	•
									٠										
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
_	_		_	_	_		_	~	-	_	_	_	_	_	_	_	_	_	_
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
								(a)											
				#	#			_											
۰	۰	。 少				۰	۰	・	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	•
۰	۰	*	۰	#	ff	٠	۰	*	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
				٠					۰										
		_	_		*				,										
۰		٠	۰	٠		۰	•	٠	۰	۰	•	۰	٠	٠	۰	۰	•	۰	•
۰	۰	۰	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
	•	•	•			•	•		۰	•			•			•		•	
		*						٠	٠			٠			٠				
-	-		,	-	•	,	•	,	•	,	•	,	•	•	,	•	•	•	-
۰	۰	۰	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
		*	•						۰										
٠	,	-	-	,	-	Ĩ	•	-	•	-	,	-	•	•	-	,	,	,	•
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
	•	•	•			•	•		۰	•			•						
	,	-	_	,	_		,	-		_	,		_	,		,	,	,	-
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	•	۰	۰	٠	٠	۰	۰	٠	۰	۰	•	۰	۰	۰	۰	۰	•	۰	
									٠			٠		٠	٠				
٠	,	-	-	,	-	Ĩ		-	•	-	,	-	•	•	-	,	,	,	•
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
	•	•	•			•	•		۰	•			•						
		٠	٠			٠				٠		٠		٠	٠	٠		٠	
		_	ĺ	Ī	_	_	ĺ	_		_	_	ĺ	_	ĺ	ĺ	ĺ	_	ĺ	_
_	_	_	_	-	-	_	_	-	_	-	_	_	_	_	_	_	_	_	_
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		٠	٠			٠			٠			٠		٠	٠	٠		٠	
				#	#			(a)											
۰	۰							_	۰	۰	۰								٠
۰	۰	不	۰	#	#	۰	۰	*	۰	۰	٠	۰	•	۰	۰	۰	٠	۰	٠
									٠										
					*														
۰	۰	۰	۰	۰		۰	۰	۰	۰		٠		۰					۰	•
۰	۰	۰	۰	۰	٠	۰	٠	٠	۰	۰	٠	۰	۰	۰	۰	۰	٠	۰	•
									٠										
		*																	
۰	۰		۰	۰	۰	۰	۰	۰	۰		٠		۰	۰	۰	۰	۰	۰	•
۰	۰	۰	•	٠	٠	۰	۰	٠	۰	۰	٠	۰	•	۰	۰	•	۰	•	•
٠		*																	
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	٠
۰	۰	۰	۰	٠	٠	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	٠	۰	٠
۰	۰	۰	۰	۰	۰	۰	٠	۰	۰		٠		٠	۰	۰	۰	٠	۰	•
۰	۰	۰	۰	٠	٠	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	۰	۰	۰	٠
	۰	۰	۰	٠	٠	۰	٠	٠	۰	۰	٠	۰	٠	۰	۰	۰	٠	۰	•
۰																			
•	۰	۰	۰	۰	۰	۰	٠	۰	۰	۰	۰	۰	-					-	

۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
٠	•	٠	٠			٠	•	A	٠	٠	•	٠	•	•	٠	٠	•	٠	٠
۰	۰	•	۰	#	#	۰	۰	@	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۰	٠	*	۰	#	#	۰	۰	@	۰	۰	٠	۰	٠	۰	۰	۰	٠	۰	٠
۰							۰		۰					۰	۰	۰			٠
					*														
		*																	
		*																	
			٠			٠		٠		٠		٠			٠	٠		٠	
·	•			•			•		-		•				-	-	•		•
۰	٠	۰	۰	۰	۰	۰	٠	•	-	-	-	-	-	-	-	۰	٠	۰	۰
۰	۰	۰	۰	۰	۰	۰	٠	٠	٠	۰	٠	۰	٠	٠	٠	٠	٠	۰	۰
۰	٠	۰	۰	٠	٠	۰	۰	٠	۰	-	-	-	-	-	-	۰	٠	۰	۰
۰	٠	۰	۰	•	٠	۰	٠	٠	۰	۰	٠	۰	٠	۰	۰	۰	٠	۰	
٠						۰	•												
٠									•										
_	_	_	_	_	_	_	_	_	_	_	_	_		_	_				