



Cal fer un programa en Java que faça una variant del joc de l'*Scrabble*, on només participarà 1 jugador.

IMPORTANT: Només és obligatori el primer punt (construir el tauler i mostrar-lo). No obstant, és una bona pràctica de cara l'examen que intenteu fer-ho tot.

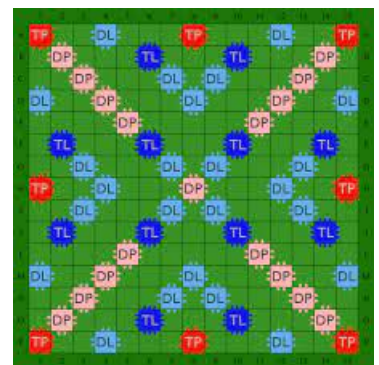
Si alguna part no saps com fer-la, consulta al professor.

1. Construir tauler

1.1. Demanar quantitat de files i quantitat de columnes (ja que el tauler podrà ser un rectangle). Tant les files com les columnes han de ser imparells.

1.2. Per al tauler caldrà definir 2 matrius de caràcters, amb estos noms:

- *taulerp*: per a guarda quin "color" té cada casella:
 l --> doble tant de lletra
 L --> triple tant de lletra
 p --> doble tant de paraula
 P --> triple tant de paraula



- *taulerf*: per a guardar quina lletra (fitxa) hi ha en cada casella.



Tant les coordenades de les files i de les columnes seran números que començaran per 1 (i acabaran en la quantitat de files o columnes del tauler).

1.3. Posar els "colors" de les caselles (de la matriu *taulerp*):

1.3.1. Inicialitza cada casella del tauler amb un espai en blanc.

1.3.2. Posar les 'p': es farà una "aspa" des del centre.

		p				p		
			p		p			
				p				
			p		p			
		p				p		

Pista:

1r. Busca la posició del centre (fila i columna).

2n. Fes un bucle per a l'aspa "dreta-baix". En cada passada del bucle anirà canviant la fila i columna i es marcarà la posició corresponent, fins que "entropesse" en una vora (de baix o de la dreta, segons el tauler).

3r. Fes el mateix per a les altres 3 aspes. Un bucle per a cada aspa, clar.

1.3.3. Posar les 'P': als 4 cantons i a la meitat dels 4 costats.

P		p		P		p		P
			p		p			
P				p				P
			p		p			
P		p		P		p		P

No passa res si alguna 'P' sobreescriu alguna lletra posada abans.

No ho facis amb bucles.
Fes una instrucció per a cada 'P'.

1.3.4. Posar 5 'I' i les seues simètriques. És a dir: fer 5 voltes estos passos:

1.3.4.1. Triar posició aleatòria. Cal buscar 2 números aleatoris (un per a la fila i altre per a la columna). Per a buscar un número aleatori entre 1 i N:

$\text{num} = (\text{int})(\text{Math.floor}(\text{Math.random()} * N)) + 1;$

No cal controlar si han eixit posicions aleatòries repetides.

1.3.4.2. Es posarà eixe color ('I') en eixa posició triada.

P		p		P		p		P
			p		p			
P				p				P
		I	p		p			
P		p		P		p		P

No passa res si les alguna 'I' sobreescriu alguna lletra posada abans.

1.3.4.3. També es posarà eixe color en les parts **simètriques** del tauler, tant en horitzontal com en vertical.

P		p		P		p		P
		I	p		p	I		
P				p				P
		I	p		p	I		
P		p		P		p		P

Caldrà calcular les coordenades d'eixes altres 3 caselles i posar allí també la 'I'.

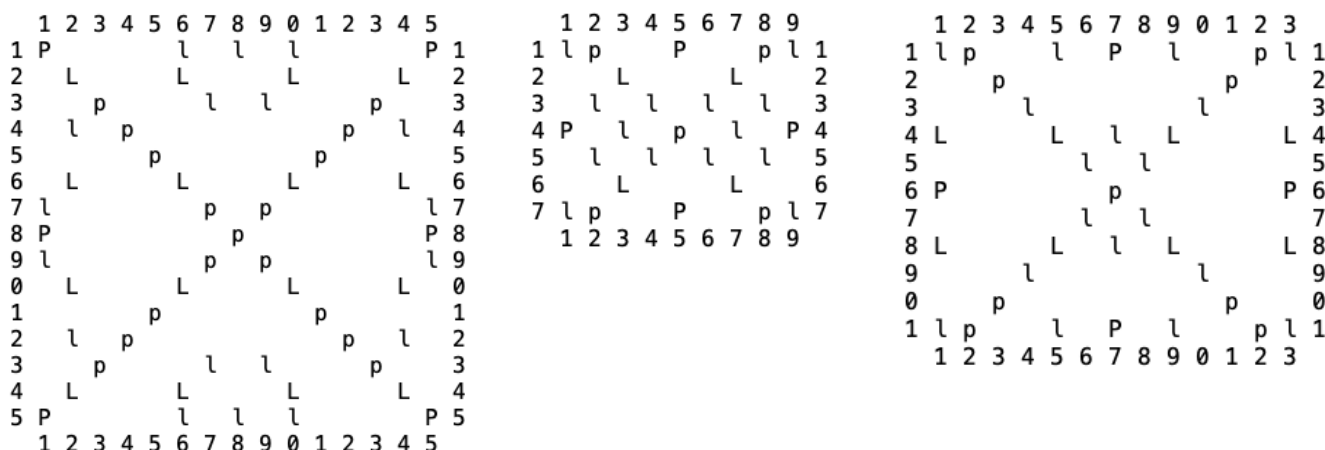
1.3.5. Posar les 'L'. Se'n posarà una cada 4 files i cada 4 columnes:

P			p			P			p			P
		I		p					p		I	
	L				L			L				L
					p		p					
P						p						P
					p		p					
	L				L			L				L
		I		p					p		I	
P			p			P			p			P

Per a este exemple de les 'L' hem suposat un tauler més gran.

A ser possible, que les 'L' estiguen en posicions concèntriques, com en l'exemple. Però si no t'ix, no passa res: comença a comptar de 4 en 4 des de l'inici de la matriu.

1.4. Es mostrarà el tauler de punts. Exemples:



2. Puntuacions de lletres i quantitat de fitxes de cada lletra:

Volem guardar esta informació:

- 0 punts: **(blanc)** x 2
- 1 punt: **E** x13, **A** x12, **I** x8, **R** x8, **S** x8, **N** x6, **O** x5, **T** x5, **L** x4, **U** x4
- 2 punts: **C** x3, **D** x3, **M** x3
- 3 punts: **B** x2, **G** x2, **P** x2
- 4 punts: **F** x1, **V** x1, **Y** x1
- 5 punts: **K** x1
- 8 punts: **H** x1, **J** x1, **Q** x1, **Z** x1
- 10 punts: **Ç** x2, **X** x2



Per a guardar això, caldrà definir 2 *HashMap*, amb estos noms:

2.1. *qpunts*, amb quants punts té cada lletra.

E	1
A	1
I	1
...	...
X	10

Caldrà afegir al *HashMap qpunts* les parelles corresponents de lletra - quantitat de punts.

2.2. *qfitxes*, amb quantes fitxes hi ha de cada lletra.

E	13
A	12
I	8
...	...
X	2

Caldrà afegir al *HashMap qfitxes* les parelles corresponents de lletra - quantitat de fitxes.

3. Jugar

3.1. Bucle amb les següents parts:

- Mostrar el tauler de fitxes
- El jugador agafa fitxes noves
- El jugador posa una paraula (si no es pot posar, acaba la partida)
- Mostrar puntuació

Ara es detallarà què ha de fer cadascuna de les accions d'este bucle.

3.1.1. Mostrar el tauler de fitxes

Recorda que tenim 2 taulers: el tauler de punts (*tpunts*) i el de fitxes (*tfitxes*). El tauler de punts l'hem mostrat al principi del programa. Ara, dins del bucle caldrà mostrar el tauler de fitxes (es mostrarà per a cada paraula que pose el jugador).

Exemples:

```
 1 2 3 4 5 6 7 8 9
1
2
3
4
5
6
7
8
9
1 2 3 4 5 6 7 8 9
```

A l'inici de la partida

```
 1 2 3 4 5 6 7 8 9
1
2
3 E X E M P L E 3
4       A       4
5 V E C T O R   5
6       R       6
7         I     7
8         U     8
9
1 2 3 4 5 6 7 8 9
```

Durant la partida

3.1.2. El jugador agafa fitxes noves

Caldrà guardar en una estructura de dades les fitxes que té el jugador en cada moment.



Quina estructura de dades seria la millor?

Abans de triar una estructura, cal mirar avantatges i inconvenients de cadascuna:

- ✓ Vector de 7 components? L'inconvenient és que per afegir lletres al vector haurem de buscar posicions buides.
- ✓ HashSet? Esta estructura de conjunt seria apropiada ja que no importa l'ordre en el qual tenim les nostres fitxes, però no ens serveix ja que en un conjunt no es poden repetir els elements (i podríem tindre diverses lletres B, per exemple).
- ✓ HashMap? Podríem tindre un diccionari lletra-quantitat, però per a afegir o llevar alguna lletra caldria vore primer si està, vore la quantitat i, després, modificar eixa quantitat.
- ✓ ArrayList? Esta serà la millor opció ja que, encara que les lletres estan ordenades, podem llevar o posar lletres en qualsevol lloc, i no hem de fer cap altra operació.

3.1.2.1. Per tant, fora del bucle declararem un ArrayList de caràcters, anomenat "meues", ja que guardarà les "meues fitxes".

3.1.2.2. Automàticament s'assignaran al jugador les fitxes que li falten fins arribar a 7. Per exemple, si en un moment determinat el jugador té 3 fitxes, se li n'assignaran 4 (no es pregunta res per teclat).

Per cada fitxa que haja de ser assignada:

a) Obtenim una lletra aleatòria.

Per a això tindrem en un String totes les lletres disponibles. Triarem un número aleatori entre eixes posicions i agafarem la lletra corresponent:

```
String lletres = "ABCÇDEFGHIJKLMNOPQRSTUVWXYZ";  
char lletra = lletres.charAt( (int)(Math.floor(Math.random()*lletres.length())) );
```

b) Assignem la fitxa al jugador:

Si queden fitxes d'eixa lletra:

- L'afegim a l'ArrayList "meues".
- Descomptem en 1 la quantitat de fitxes que queden d'eixa lletra.

Si no, tornem a agafar altra lletra aleatòria.

3.1.2.3. Es mostren les 7 fitxes del jugador.

Per exemple:

Les meues lletres: A B D O G A R

3.1.3. El jugador posa una paraula

3.1.3.1. Es demana per teclat:

- La paraula que es vol posar.
- La posició inicial de la paraula en el tauler (fila, columna).
- La direcció: (d)reta / (e)squerra / (b)aix / (a)munt

(S'imagina que la paraula que vol posar el jugador existeix al diccionari).

3.1.3.2. Es posa la paraula al tauler. Si no és possible, acaba la partida.

```
// Iniciem acumuladors:
puntsParaula = 0;
q2tp = 0; // quantitat de doble tants de paraula que ens trobem
q3tp = 0; // quantitat de triple tants de paraula que ens trobem

for (cada lletra de la paraula a posar) {
    si (la posició en tauler està fora dels límits) o
        (en tauler hi ha lletra distinta en eixa posició) o
        (en tauler la pos està buida i no tenim la fitxa) {
        fi partida
    sino {
        si (en tauler la pos està buida i la tenim nosaltres){
            la llevem de "meues"
            la posem en eixa posició
        }
        // (si en tauler ja està la fitxa no haurem de fer res)
    }
    Avancem posició en tauler (Compte! Segons la direcció triada)
    Avancem posició en la paraula
    Calculem puntsLletra (tenint en compte doble o triple tants de lletra)
    Acumulem puntsParaula
    Si cal, incrementem q2tp
    Si cal, incrementem q3tp
}
Calculem puntsJugada amb puntsParaula, q2tp i q3tp
Acumulem puntsPartida amb els punts de la jugada.
```

Caldrà implementar este algoritme en Java.

I si no volguérem acabar la partida si no cap la paraula?

Problema: Caldria desfer les lletres posades de la paraula que estàvem posant i tornar a posar-les a la llista "meues".

Solució:

- Abans d'intentar posar la paraula, fer una còpia de seguretat (en altres variables, amb altra zona de memòria) del tauler de fitxes (tfixes) i de les fitxes nostres (meues).
- Si veiem que no cap la paraula, tornarem a deixar-ho com estava (fent que les variables originals "apunten" a les de la còpia de seguretat).

3.1.4. Mostrar puntuació

Es mostraran els punts de la jugada que s'acaba de fer i els punts acumulats de totes les jugades fins a eixe moment. Per exemple:

Punts de la jugada: 20

Punts acumulats: 87

Opcionalment també es podrien detallar els punts de la jugada indicant quants dobles tants de paraula hi havia, etc.