Exercici 2:

Utilitzant quan calgui els tipus definits en l'exercici anterior, declareu les accions i funcions especificades. No és necessari dissenyar-les.

A. Declareu una acció/funció getCharacter que a partir d'una sopa tSoup, retorni el caràcter que es troba en una fila i columna donades.

```
funcio getCharacter (sopa : tSoup, fila : sencer, columna : sencer) : caracter
var
        car : caràcter
fvar
 si fila >= 1 i fila <= num_filas(sopa) i columna >= 1 i columna <= num_columnes(sopa)
llavors
        car := sopa[fila, columna]
        retorna car
 sinó
        retorna
 fsi
ffuncio
```

B. Declareu una acció/funció setCharacter que col·loqui un caràcter donat en una determinada posició d'una tSoup.

```
acció setCharacter (sal sopa : tSoup, fila : sencer, columna : sencer, caràcter : caràcter)
fvar
 si fila >= 1 i fila <= num_filas(sopa) i columna >= 1 i columna <= num_columnes(sopa)
llavors
        sopa[fila, columna] := caràcter
 finsi
facció
```

C. Declareu una acció/funció setWord que col·loqui una tWord en una determinada posició d'una tSoup, i amb una orientació donada. A més, la paraula és marcada com a "no trobada" i inicialitzada amb la seva posició i orientació.

```
acció setWord (sal sopa : tSoup, paraula : tWord, fila : sencer, columna : sencer, orientació :
tLloc)
var
        i : sencer
        longitudParaula : sencer
        novaFila, novaColumna : sencer
fvar
 longitudParaula := longitud(paraula.paraula)

 per a := 1 fins a longitudParaula fer
        novaFila := fila
```

```
        novaColumna := columna
        segons orientacio fer
        cas horitzontal: novaColumna := columna + i - 1
        cas vertical: novaFila := fila + i - 1
        cas diagonal:
        novaFila := fila + i - 1
        novaColumna := columna + i - 1
        fi_segon

 si novaFila >= 1 i novaFila <= num_filas(sopa) i novaColumna >= 1 i novaColumna <=
num_columnes(sopa) llavors
 caracter_de(paraula.paraula[i])

        paraula.paraula[i].horizontal := novaColumna
        paraula.paraula[i].vertical := novaFila
        paraula.paraula[i].diagonal := orientacion
 finsi
 fpara

        paraula.trobar := fals
        paraula.posicion.horizontal := columna
        paraula.posicion.vertical := fila
        paraula.posicion.diagonal := orientacion

facció
```

D. Declareu una acció/funció initSoup que inicialitzi una sopa tSoup. amb les dimensions donades.

```
acció initSoup (sal sopa : tSoup, numFiles : sencer, numColumnes : sencer)
var
        i, j : sencer
fvar
 redimensionar(sopa.sopa, numFiles, numColumnas) // Funció hipotètica

        per a i := 1 fins numFiles fer
        per j := 1 fins numColumnes fer
        sopa.sopa[i, j] := vacía
        fpara
        fpara
       facció
```

E. Declareu una acció/funció readWord que llegeixi i retorni una paraula tWord de l'entrada estàndard.

```
funcio readWord () : tWord
var
        paraulaLeida : array [1..CARACTERS] DE caracter
        longitudLleida : sencer
```

```
        i : sencer
        novaParaula : tWord
fvar
        llegir_cadena(paraulaLeida) // Funció hipotètica

        longitudLeida := longitud(paraulaLeida)

        per a i := 1 fins a CARACTERS fer
        novaParaula.paraula[i].horitzontal := 0
        novaParaula.paraula[i].vertical := 0
        novaParaula.paraula[i].diagonal := horitzontal
        fpara
        novaParaula.posició.horitzontal := 0
        novaParaula.posicion.vertical := 0
        novaParaula.posicion.diagonal := horitzontal
        novaParaula.trobar := FALS

        torna novaParaula
ffuncio
```