

Exercici 3:

Continuant amb l'aplicació de la "sopa de lletres", el que se'ns demana ara és que dissenyem un algorisme que sigui capaç d'omplir una quadrícula amb lletres aleatòries i, a continuació, hi insereixi les diferents paraules que caldrà cercar després. Les paraules sobreescriran les lletres, per la qual cosa, el resultat final serà el de una "sopa de lletres" (les paraules estaran camuflades entre les lletres prèviament escrites).

Tenim la següent seqüència d'entrada on apareix la informació que necessitem:

```
< N M W word1 word2 word3 ... wordW >
```

On:

- N i M les dimensions de la quadrícula de la sopa de lletres: N files x M columnes.
- W el nombre de paraules a afegir a la sopa de lletres.
- wordi la paraula i-èsima que cal afegir a la sopa de lletres, en majúscules.

El següent algorisme llegeix la seqüència d'entrada, crea una quadrícula de N x M, l'omple de lletres aleatòriament i, per últim, hi sobreesciu les paraules situant-les al atzar sobre la quadrícula. El resultat es mostra en una seqüència de sortida amb el següent format:

```
< N M c1 c2 c3 c4 ... cNxM >
```

On:

N i M són les mateixes dimensions de la quadrícula que hem llegit de l'entrada.
ci és el caràcter i-èsim de la sopa de lletres, **en majúscules**.

Es demana que completeu els **punts suspensius** que falten per tal que l'algorisme es comporti tal com s'ha descrit.

NOTA: Utilitzeu les estructures de dades, accions i funcions declarades en els exercicis anteriors quan sigui necessari. També podeu utilitzar la funció següent, que us proporciona un nombre aleatori entre dos enters donats:

funcio getRandomNumber(min: **enter**, max: **enter**): **enter**

algorisme fillSoup

var

soup: **tSoup**;
search: **tSearch**;
word: **tWord**;
n,m,w,i: **enter**;

fvar

n:= readInteger();
m:= readInteger();
initSoup(n,m, soup);
initSearch(search);

fillWithRandomCharacters(n, m, soup);
w:= readInteger();

```

    per i:= 1 fins w fer
        word:= .....
        addWordToSearch( search, word );
        hideWordIntoSoup(word, n, m, soup);
    fper
writeSoup( n, m, soup);

```

falgorisme

```

funcio getRandomCharacter(): caracter
    var
        min, max, code: enter;
    fvar
        .....

```

ffuncio

```

accio fillWithRandomCharacters(ent n: enter, ent m: enter, entsor soup: tSoup)
    var
        c: caracter;
        i,j: enter;
    fvar
        per i:= 1 fins n fer
            per j:= 1 fins m fer
                c:= getRandomCharacter();
                .....
            fper
        fper

```

faccio

```

accio writeSoup(ent n,m: enter, ent soup: tSoup)
    var
        i,j: enter;
    fvar
        per i:= 1 fins n fer
            per j:= 1 fins m fer
                writeChar( ..... );
            fper
        fper

```

faccio

```

accio hideWordIntoSoup( entsor word: tWord, ent n: enter, ent m: enter, entsor soup:
tSoup )
    var
        i,j,ori,len: enter;
    fvar
        len:= getWordLength(word); ori:= getRandomNumber(1,3);

```

```

si ori = 1 llavors { case: VERTICAL }
i:=getRandomNumber(1,n-len+1);
j:=getRandomNumber(1,m);
.....
sino
    si ori = 2 llavors { case: HORIZONTAL }
        i:=getRandomNumber(1,n);
        j:=getRandomNumber(1,m-len+1);
    sino { case: DIAGONAL }
        .....
    fsi
fsi
faccio

accio addWordToSearch( entsor search: tSearch, ent word: tWord )

```

```

.....

```

faccio

```

accio initSearch( sor search: tSearch )

```

```

.....
faccio

```

```

accio initSoup( ent n,m: enter, sor soup: tSoup )

```

```

.....

```

facció

Algoritme Inserit:

Obtenir lletra aleatoria:

```

funcio getRandomCharacter(): caracter
var
    min, max, code: enter;
fvar
    min := 65;    // A
    max := 90;    // Z
    code := getRandomNumber(min, max);
    retorna chr(code);
ffuncio

```

Omplir la sopa lletres aleatòries:

```
accio fillWithRandomCharacters(ent n: enter, ent m: enter, entsor soup: tSoup)
var
  c: character;
  i, j: enter;
fvar
  per i := 1 fins n fer
    per j := 1 fins m fer
      c := getRandomCharacter();
      soup[i][j] := c;
    fper
  fper
faccio
```

Escriure la sopa:

```
accio writeSoup(ent n, m: enter, ent soup: tSoup)
var
  i, j: enter;
fvar
  writeInteger(n);
  writeInteger(m);
  per i := 1 fins n fer
    per j := 1 fins m fer
      writeChar(soup[i][j]);
    fper
  fper
faccio
```

Inserir paraula:

```
accio hideWordIntoSoup(entsor word: tWord, ent n: enter, ent m: enter, entsor soup: tSoup)
var
  i, j, k, ori, len: enter;
fvar
  len := getWordLength(word);
  ori := getRandomNumber(1, 3);

  si ori = 1 llavors {VERTICAL}
    i := getRandomNumber(1, n - len + 1);
    j := getRandomNumber(1, m);
    per k := 1 fins len fer
      soup[i + k - 1][j] := word[k];
    fper
  sino si ori = 2 llavors {HORIZONTAL}
    i := getRandomNumber(1, n);
```

```
j := getRandomNumber(1, m - len + 1);
per k := 1 fins len fer
    soup[i][j + k - 1] := word[k];
fper
sino {DIAGONAL}
    i := getRandomNumber(1, n - len + 1);
    j := getRandomNumber(1, m - len + 1);
    per k := 1 fins len fer
        soup[i + k - 1][j + k - 1] := word[k];
    fper
fsi
faccio
```