

# ESTRUCTURES DE DADES I ALGORISMES

## PRÀCTICA 3. Primavera 2022

### Primer exercici: conversió String -> int

Escriviu una funció RECURSIVA que donada una cadena de caràcters no buida que només conté dígitos la transformi en el corresponent número enter. Per exemple, si se li dona la cadena "13045" proporciona com el resultat el número enter 13045. Si se li dona la cadena "0" proporciona com a resultat el número enter 0 i si se li dona la cadena "0050" proporciona el número enter 50. Us pot ser útil pensar en una estratègia "cua-cap" (ull, "cua-cap" més que no pas "cap-cua").

Se us subministra un punt d'entrada en execució que prova de determinar si la conversió és correcta o no.

### Segon exercici: generació de palíndroms

La següent és una definició recursiva de mot (cadena, String) palíndrom:

- 1.- La cadena buida ("" ) és un palíndrom (de mida 0)
- 2.- Si c és un caràcter qualsevol, la cadena que només conté c ("c") és un palíndrom (de mida 1)
- 3.- Si c és un caràcter qualsevol i S és un palíndrom, la cadena cSc és un palíndrom (de mida 2 + la mida de S)

Escriviu el cos de la funció RECURSIVA

```
public static String genPalindrome (int length)
```

que genera un mot palíndrom de la mida indicada en el paràmetre.

Disposau d'una funció que genera caràcters a l'atzar. Utilitzeu-la per generar un "caràcter qualsevol".

Heu de posar a prova la vostra implementació de la funció genPalindrome. Feu-ho de la següent manera: escriviu un punt d'entrada en execució que, iterativament, generi 10000 palíndroms de mides aleatòries en l'interval [0, 20]. Per a cada un dels palíndroms generats verifiqueu si efectivament és un mot palíndrom o no utilitzant la funció checkPalindrome que se us subministra. Si el vostre programa és correcte i per a algun dels mots el resultat és que no és un palíndrom llavors ja podeu estar segurs que la vostra implementació de la funció no és correcta. El codi d'aquest programa també forma part de l'exercici i cal lliurar-lo.

### Tercer exercici: vector "girat"?

Heu d'escriure una funció RECURSIVA que donats dos vectors d'enters de la mateixa longitud, determini si un és la versió "girada" de l'altre (conté els mateixos elements però en l'ordre contrari). Per exemple, els vectors {5, 6, 8, 9} i {9, 8, 6, 5} són l'un la versió girada de l'altre. També ho són els vectors {1, 2, 3, 4, 5} i {5, 4, 3, 2, 1}. En canvi, els vectors {7, 3, 8, 9, 5} i {5, 9, 8, 3, 4} no ho són, com tampoc no ho són els vectors {1, 2, 3, 5} i {5, 2, 3, 1}.

Per tal de posar a prova la vostra implementació se us subministra una classe anomenada VectorPair. Els objectes d'aquesta classe encapsulen (bo i que de manera pública) un parell de vectors d'enters anomenats one i two. Si en invocar el constructor de la classe el paràmetre donat té valor true, llavors one i two són l'un la versió girada de l'altre. En canvi, si el paràmetre donat té valor false, one i two no són l'un la versió girada de l'altre.

La prova haurà de consistir en: generar 1000 parells de vectors que siguin l'un la versió girada de l'altre i verificar-ho amb la funció prèviament escrita. Després, generar 1000 parells de vectors que no siguin l'un la versió girada de l'altre i verificar-ho amb la funció prèviament escrita. Si en algun cas

el resultat no coincideix amb l'esperat, ja podreu estar segurs que la vostra implementació de la funció no és correcta. El codi d'aquest programa també forma part de l'exercici i cal lliurar-lo.

#### **Quart exercici: *torturant les as -una altra vegada-***

Basant-vos en **l'estratègia de les dues meitats** escriviu una funció RECURSIVA que donada una cadena de caràcters en generi una altra en què els caràcters 'a' (a minúscula) que apareixen en posicions parells siguin substituïdes pel caràcter '\*' i les que apareixen en posició senar siguin eliminades.

Per exemple, si la cadena subministrada és "abracadabral" la cadena resultant hauria de ser "\*brcdbrl\*". Si la cadena fos "paranormalitats" la cadena resultant seria "prnorm\*lit\*ts". La cadena "pana" donaria lloc a "pn" i la cadena "ana" donaria lloc a "\*n\*".

Recordeu que els caràcters en les cadenes es numeren a partir de zero.

Per tal de posar a prova la funció anterior, feu també el següent:

- Escriviu una funció no recursiva amb capçalera  
**public static** String tortura (String s)  
que faci de façana (o llançadora) de la funció recursiva
- Escriviu una versió iterativa de la funció que *tortura les as* amb capçalera  
**public static** String torturaIte (String s)
- Escriviu una funció **també RECURSIVA** que generi una cadena de caràcters aleatòria (un únic paràmetre, la mida de la cadena) sobre l'alfabet {a, b, c, d, e, f}. La seva capçalera haurà de ser  
**public static** String genCadena (**int** n)

Amb aquests elements, el punt d'entrada en execució que se us subministra prova de verificar que les versions iterativa i recursiva de la funció generen sempre el mateix resultat.

### Què se us subministra?

Se us subministra un arxiu comprimit que conté una carpeta que al seu torn conté un projecte Eclipse. Descomprimiu l'arxiu i importeu el projecte al vostre workspace (**Import →...**) Heu de completar aquest projecte i lliurar-lo.

### Què s'ha de lliurar i com?

Abans de fer el lliurament, **heu de canviar el nom del projecte** (ULL, per canviar el nom del projecte no n'hi ha prou amb canviar el nom de la carpeta que el conté. Dins d'Eclipse cal utilitzar **Refactor → Rename** per tal de fer aquest canvi). Una vegada fet això **també s'ha de canviar el nom de la carpeta** (és el mateix que havíeu de fer en els lliuraments anterior)

El projecte (i la carpeta que el conté) s'ha d'anomenar

PRAC3\_Individual\_Cognom1Cognom2

O bé

PRAC3\_Bipersonal\_Cognom1Cognom2\_Cognom1Cognom2

O bé

PRAC3\_Tripersonal\_Cognom1Cognom2\_Cognom1Cognom2\_Cognom1Cognom2

en funció de si es tracta d'un lliurament unipersonal, "bipersonal" o "tripersonal"

Finalment cal comprimir tota la carpeta amb el seu contingut (ull, s'ha de comprimir a nivell de carpeta, no a nivell de contingut) i fer el lliurament a través d'e-campus.

**Els lliuraments mal identificats i/o que no respectin les especificacions donades no seran avaluats (= qualificació 0.0).**

### On s'ha de lliurar?

El lliurament es farà a través d'e-campus i **no s'acceptarà cap altra via**, ni el tan sols el correu electrònic. Feu atenció a la data i hora límit.

### Quan s'ha de lliurar?

La data límit de lliurament s'anunciarà a eCampus