

Strings

Per a cada exercici proposat, teniu anotat el següent format:

(*directoryName*, *ClassName*, *methodName1*, *methodName2*,...)

Segueix les següents instruccions:

- a) Crea un directori de nom *StringsCognomsNom*.
- b) Per a cada exercici, crea un directori amb el nom *directoryName* ubicat dins del directori *StringsCognomsNom*.
- c) Per a cada exercici, crea una classe de nom *ClassName* dins del directori *directoryName*. El nom del fitxer on està la classe serà *ClassName.java*.
- d) Per a cada exercici, codifica dins la classe un o més mètodes amb els noms *methodName1*, *methodName2*...

1. (*numberOfVowels*, Program, *numberOfVowels*) Donada una cadena calcula quants d'ells són vocals. S'ha de considerar que tant el caràcter **a** com el caràcter **A** són vocals.

2. (*firstVowel*, Program, *firstVowel*) Donada una cadena, troba la primera vocal. Si no té vocals, retornarem un espai en blanc.

3. (*allVowels*, Program, *allVowels*) Donada una cadena, determina si apareixen totes les vocals.

4. (*moreVowels*, Program, *moreVowels*) Donada una cadena, determina si hi ha més vocals que altres caràcters.

5. (*onlyDigits*, Program, *onlyDigits*) Donada una cadena, determina si només conté dígit.

6. (*answerphone*, Program, *answerphone*) Donades les dades d'una persona que marxa de viatge (nom, sexe(H/D), telèfon i ciutat on ha marxat), crea la cadena que hauria de dir un contestador automàtic:

a. Si el sexe és 'H' la cadena serà: "Hola! Sóc **el** [nom]. Has trucat al [telèfon] i ara no estic a casa perquè estic a [ciutat] de vacances!!!"

b. Si el sexe és 'D' la cadena serà: "Hola! Sóc **la** [nom]. Has trucat al [telèfon] i ara no estic a casa perquè estic a [ciutat] de vacances!!!"

7. (*charInString*, Program, *isCharInString*) Donats un caràcter i una cadena, determina si el caràcter es troba dins de la cadena o no. Fes dues versions del programa: una usant el mètode *indexOf(c)* i una altra versió sense usar aquest mètode.

8. (*occurrencesOfChar*, Program, *occurrencesOfChar*) Donats un caràcter i una cadena, calcula quantes vegades apareix el caràcter dins la cadena.

9. (*reverse*, Program, *reverse*) Donada una cadena, crea una altra cadena amb els caràcters de la cadena donada invertits.

10. (removeSpaces, Program, removeSpaces) Donada una cadena, crea una altra cadena igual a la primera sense espais en blanc.
11. (areEquals, Program, areEquals) Donades dues cadenes, determina si són iguals o no. Fes dues versions del programa: una usant el mètode areEquals(s1,s2) i una altra versió sense usar aquest mètode.
12. (palindrome, Program, isPalindrome) Donada una cadena determina si és un palíndrom.
13. (isSubstring, Program, isSubstring) Donades dues cadenes, determina si la primera és una subcadena de la segona. Fes dues versions del programa: una usant el mètode indexOf(s) i una altra versió sense usar aquest mètode.
14. (occurrencesOfString, Program, occurrencesOfString) Donades dues cadenes, determina quantes vegades apareix la segona dins la primera.
15. (withoutString, Program, withoutString) Donades dues cadenes, comprova si la segona és subcadena de la primera i si és així, crea una tercera cadena igual a la primera sense l'aparició de la segona. Si apareix més d'un cop, només es suprimirà la primera aparició.
16. (positionDiffers, Program, positionDiffers) Donades dues cadenes, determina en quina posició discrepen per primer cop. Si no discrepen, retornarem -1.
17. (toUpper, Program, toUpper) Donada una cadena, crea una segona cadena amb els caràcters que estaven en minúscula passats a majúscula. Fes dues versions del programa: una usant el mètode toUpperCase() i una altra versió sense usar aquest mètode.
18. (moreRepeatedChar, Program, moreRepeatedChar) Donada una cadena, determina quin és el caràcter que apareix més cops. Retornarem un array de dues cel·les on la primera cel·la contindrà el codi ASCII del caràcter més repetit i en la segona cel·la quants cops apareix.
19. (dni2nif, Program, dni2nif) Donada una cadena amb el número de un DNI, genera una cadena amb el NIF. El NIF s'obté a partir del DNI afegint-li la lletra que s'obté calculant el residu de la divisió entera del DNI entre 23 . Les lletres són: TRWAGMYFPDXBNJZSQVHLCKE i la transformació de codi és: 0 ->T, 1->R, 2->W, etc. Exemple: Al DNI 37721039 li correspon el NIF 37721039G
20. (insertChars, Program, insertChars) Donades dues cadenes, genera una altra cadena amb els caràcters de les anteriors intercalats començant pel primer caràcter de la primera cadena. Si els caràcters d'una de les cadenes s'esgoten, s'afegeixen tots els caràcters de l'altra.
21. (decimal2binary, Program, decimal2binary) Donat un nombre enter positiu, construeix una cadena que representi el mateix nombre en el sistema de numeració binari.
22. (binary2decimal, Program, binary2decimal) Donada una cadena formada pels caràcters 0 i 1 que codifica un nombre enter no negatiu en el sistema de numeració binari, calcula un nombre enter que sigui la seva representació en el sistema de numeració decimal.
23. (numberOfWords, Program, numberOfWords) Donada una cadena, calcula el nombre de

paraules que conté. Considerarem una paraula com una seqüència de caràcters separada per espais en blanc.

24. (sameEndingChar, Program, sameEndingChar) Donada una cadena, determina si totes les paraules acaben amb la mateixa lletra. Fes dues versions del programa: una usant el mètode endsWith(s) i una altra versió sense usar aquest mètode.

25. (encryptCaesar, Program, encryptCaesar) Donada una cadena de lletres majúscules i un nombre enter en l'interval [1,9], genera la cadena codificada segons l'algorisme de Julius Cesar (http://ca.wikipedia.org/wiki/Xifratge_de_C%C3%A8sar).

26. Per als exercicis que no disposin de TUI, crea la classe ProgramaTUI, en la qual hi hagi el mètode que ja havíeu codificat i un nou mètode main en el qual es crearà una interfície d'usuari per terminal.