



## Instruccions:

- Llegeix atentament cada exercici abans de començar.
- Completa cada exercici utilitzant programació modular amb l'ús de paquets i prenent en consideració els conceptes apresos durant el curs.
- Assegura't de modular correctament els programes i fer servir crides a funcions adequadament parametritzades.
- Recorda tenir en compte l'àmbit de les variables en les crides a les funcions.

## Exercicis:

1. Crea un paquet anomenat "stats" que contingui una classe anomenada "Statistics" amb una funció estàtica que calculi la desviació estàndard d'una llista de nombres. Utilitza l'algorisme correctament i gestiona els casos especials. (RA: 1.1, 1.2, 1.4)
2. Crea un paquet anomenat "sorting" que contingui una classe anomenada "Sorting" amb una funció que ordeni una llista de nombres de menor a major utilitzant un algorisme d'ordenació eficient com ara QuickSort o MergeSort. (RA: 1.3, 1.5)
3. Escriu un programa modular que faci servir el paquet "sorting" per ordenar una llista de nombres enters introduïts per l'usuari i mostrar-los per pantalla. Assegura't de proporcionar una interfície clara i amigable per a l'usuari. (RA: 1.6)
4. Crea un paquet anomenat "strings" que contingui una classe anomenada "StringUtils" amb una funció que ordeni una llista de cadenes de text alfabèticament utilitzant un algorisme d'ordenació eficient com ara QuickSort o MergeSort. (RA: 1.3, 1.4)
5. Crea un paquet anomenat "search" que contingui una classe anomenada "SearchUtils" amb una funció que trobi el nombre més petit d'una llista de nombres enters utilitzant una tècnica d'optimització com ara la cerca binària. (RA: 1.5)
6. Escriu un programa modular que faci servir el paquet "search" per trobar el nombre més petit d'una llista de nombres enters introduïts per l'usuari i el mostri per pantalla. (RA: 1.6)
7. Crea un paquet anomenat "wordcount" que contingui una classe anomenada "WordCounter" amb una funció que compti el nombre de paraules en una cadena de text. Assegura't de gestionar correctament els casos especials com ara els caràcters de puntuació. (RA: 1.3, 1.4)



### Rúbrica:

Criteri	Excel·lent	Molt bé	Acceptable	Insuficient	Puntuació
Correcta implementació de les funcions modulars requerides. Ús adequat de paquets per organitzar el codi.	Implementació impecable de les funcions amb un ús exemplar de paquets.	Implementació correcta de les funcions amb un bon ús de paquets.	Implementació amb alguns errors menors o mancances en l'ús de paquets.	Implementació inadequada de les funcions amb un ús deficient o inexistent de paquets.	20%
Ús de l'algorisme adequat per a cada tasca (per exemple, BubbleSort per a la tasca de ordenació).	Selecció i implementació excel·lents de l'algorisme, demostrant un alt nivell de comprensió i aplicació.	Utilització correcta d'algorismes amb una comprensió adequada dels principis subjacents.	Ús d'algorismes amb algunes deficiències o mancances en la comprensió.	Selecció inadequada o implementació incorrecta d'algorismes.	15%
Interfície clara i amigable per a l'usuari en la lectura d'entrades i la presentació de resultats.	Interfície molt clara i intuïtiva, facilitant la comprensió i l'ús per a l'usuari.	Interfície clara amb algunes opcions d'optimització possibles.	Interfície funcional, però amb algunes mancances en la claredat o facilitat d'ús.	Interfície confusa o poc amigable, dificultant la comprensió i l'ús per a l'usuari.	15%
Maneig correcte de	Maneig impecable	Maneig adequat de	Maneig correcte en	Maneig inadequat o	15%



casos especials i errors (per exemple, quan la llista està buida).	de casos especials i errors amb un comportame nt esperat en totes les situacions.	la majoria dels casos especials i errors, amb només algunes petites deficiències.	algunes situacions, però amb mancances notables en altres casos.	inexistent de casos especials i errors, amb resultats inesperats o errors.	
Claredat del codi	L'ús de paquets és coherent i millora la comprensió del codi.	El codi està ben estructurat i la seva funcionalitat és clara.	El codi és comprensibl e, però hi ha algunes àrees on la claredat pot ser millorada.	Hi ha mancances significatives en la claredat del codi, dificultant la comprensió i la lectura.	15%
Disseny Top-Down	El codi està dissenyat amb una estructura clara i modular utilitzant el mètode top-down.	El disseny top-down és evident en l'organització i la jerarquia del codi.	El disseny top-down és present, però hi ha algunes incoherèncie s o mancances en la seva implementac ió.	Hi ha una manca significativa d'ús del disseny top-down, dificultant la comprensió i el manteniment del codi.	20%