

MENÚ DE COL·LECCIONS

Crea un programa modular anomenat `Menú de Col·leccions` que permet triar entre diferents exercicis (1 a 12). Cada exercici és una funció (`static void ExerciciX()` o el nom que vulgueu donar-li a la funció), i el programa torna al menú fins que l'usuari triï 0 per sortir.

```
===== MENÚ DE COL·LECCIONS =====  
1. Afegir i eliminar noms d'una llista (List)  
2. Cercar noms que comencen per una lletra concreta (List)  
3. Comptar paraules repetides (Dictionary)  
4. Mostrar el producte més car d'un catàleg (Dictionary)  
5. Simular una cua d'atenció al client (Queue)  
6. Simular una cua circular amb prioritat (Queue)  
7. Simular una pila de navegació (Stack)  
8. Comprovar si una paraula és palíndrom (Stack)  
9. Crear un conjunt únic de números (HashSet)  
10. Intersecció de conjunts (HashSet)  
11. Ordenar i filtrar llista de nombres (LINQ)  
12. Trobar paraules llargues d'una frase (LINQ)  
0. Sortir
```

EXERCICIS:

1. Demana noms a l'usuari per Consola fins que premi enter. Després mostra tots els noms introduïts i pregunta si vol eliminar algun nom.

```
Introdueix noms (doble enter per acabar):  
Anna  
Pau  
Maria  
  
Llista actual:  
- Anna  
- Pau  
- Maria  
  
Vols eliminar algun nom? Pau  
  
Pau eliminat correctament.  
  
Llista actual:  
- Anna  
- Maria
```

2. Dona una llista predefinida amb alguns noms. Demana una lletra i mostra tots els noms que comencen per aquesta lletra (sense distingir majúscules/minúscules).

```
Noms disponibles: Anna, Pau, Oriol, Esther, Arnau, Marta

Introdueix una lletra: A

Noms que comencen per 'A':
Anna
Arnau
```

3. Demana una frase per Consola i compta quantes vegades apareix cada paraula. Guarda les dades en un `Dictionary<string, int>`.

```
Escriu una frase:
el gat menja el peix i el gos menja pa

Freqüència de paraules:
el: 3
gat: 1
menja: 2
peix: 1
i: 1
gos: 1
pa: 1
```

4. Crea un `Dictionary<string, double>` amb alguns productes i preus. Mostra quin és el més car i el seu preu.

```
Productes: pa=1.2, llet=0.9, formatge=3.5, ous=2.2

El producte més car és formatge (3.5 €)
```

5. Fes servir una `Queue<string>` amb alguns noms de clients. Simula que els clients van sent atesos (surt el primer de la cua amb `Dequeue()`).

```
Clients a la cua: Anna, Joan, Maria, Pau

Atent a Anna
Atent a Joan
Atent a Maria
Atent a Pau

Cua buida!
```

6. Fes servir una `Queue<string>` amb alguns noms de clients. Cada client té un nom i un temps d'atenció (en minuts). El programa ha de treure'ls de la cua un per un i mostrar el temps total acumulat.

```
Cua de clients:
Anna (3 min)
Pau (5 min)
Maria (2 min)
Joan (4 min)

--- Atenció al client ---
Atent a Anna (3 min). Temps total: 3 min
Atent a Pau (5 min). Temps total: 8 min
Atent a Maria (2 min). Temps total: 10 min
Atent a Joan (4 min). Temps total: 14 min

Tots els clients han estat atesos!
```

7. Simula un navegador amb una `Stack` on es guarden les pantalles visitades. Mostra la pàgina actual i fes que l'usuari "tiri enrere" amb un `Pop()`.

```
Historial: Inici → Productes → Detall → Cistella

Pantalla actual: Cistella
Tornes enrere: Detall
```

8. Demana una paraula per Consola i comprova si és igual llegida del dret i del revés, utilitzant una `Stack<char>` per invertir-la.

```
Introdueix una paraula: radar
És un palíndrom!

Introdueix una paraula: casa
No és un palíndrom.
```

9. Demana números a l'usuari per Consola i desa'ls en un `HashSet<int>` (no permet duplicats). Quan acabi, mostra els números únics introduïts.

```
Introdueix números (doble enter per acabar):
3
7
3
5

Números únics:
3
7
5
```

10. Crea dos conjunts (`HashSet<int>`) amb valors diferents. Mostra els elements comuns entre tots dos.

```
Conjunt A: {1, 2, 3, 4, 5}
```

```
Conjunt B: {4, 5, 6, 7}
```

```
Intersecció: 4, 5
```

11. Dona una `List<int>` amb valors desordenats. Fes servir LINQ (`Where`, `OrderBy`) per mostrar només els números parells ordenats de menor a major.

```
Llista original: 5, 1, 8, 3, 9, 2, 7
```

```
Números parells ordenats:
```

```
2
```

```
8
```

12. Demana una frase per Consola, separa les paraules i mostra les que tinguin més de 4 lletres, ordenades de la més llarga a la més curta. No es pot fer servir `Split()` dins un bucle, però sí per dividir una sola vegada la frase.

```
Escriu una frase:
```

```
aquest exercici és força interessant
```

```
Paraules de més de 4 lletres:
```

```
interessant
```

```
exercici
```

```
aquest
```

```
força
```