

Exercici Opcional 2

- a) Implementeu una funció que donats dos vectors d'enters, retorni al primer vector la intersecció dels dos vectors, és a dir ha d'eliminar del primer vector tots els elements que no estan també al segon. Per exemple si el primer vector té els elements 1 6 5 3 9 i el segon té els elements 4 2 3 7 6 el contingut final del primer vector ha de ser 6 3. La mida final del vector s'ha de modificar també per ajustar-se al nou contingut del vector.

La capçalera de la funció ha de ser la següent:

```
void interseccioVectors (int vector1[], int& mida1, int vector2[], int mida2);
```

Us proposem la següent descomposició de la funció en funcions més simples:

1. Creeu una funció que donat un vector i un número, retorni si el número està dins del vector o no.

```
bool cercaElement (int vector[], int mida, int valor);
```

2. Creeu una funció que donat un vector i un índex de posició dins del vector, elimini el valor que ocupa la posició de l'índex, desplaçant cap a l'esquerra tots els elements que hi ha a partir d'aquella posició.

```
int eliminaElement (int vector[], int& mida, int index);
```

Per exemple, si tenim el vector 1 5 6 2 i l'índex 1 després d'executar la funció el vector ha de contenir 1 6 2 (recordeu que la primera posició del vector és 0) i la mida del vector s'ha disminuir en 1.

- b) Implementeu una funció que faci també la intersecció de dos vectors, però ara podeu suposar que els dos vectors inicials ja estan ordenats, i la intersecció s'ha de retornar amb els valors ordenats en un tercer vector que es passa com a paràmetre. Penseu bé com podeu optimitzar i simplificar el codi.

La capçalera de la funció ha de ser la següent:

```
void interseccioVectorsOrdenats (int vector1[], int mida1, int vector2[], int mida2,  
                                int vectorInterseccio[], int& midaInterseccio);
```