

## Exercicis IIP Segon Parcial - Bucles amb arrays (Temes 6 i 7)

### P2 - Curs 18/19: 2 punts

### RECORREGUT { RECORREGUT }

**Es demana:** escriure un mètode estàtic que tinga com paràmetres un array d'int anomenat `limits` i un array de `double` anomenat `valors`. El resultat ha de ser un array d'enters amb la mateixa longitud que l'array `limits` i tal que l'element  $i$ -èsim ha de ser igual al número de valors de l'array `valors` que siga inferior a l'element  $i$ -èsim de l'array `limits`.

Per exemple, si els paràmetres són els següents:

- `limits` = {15, 35, 50, 37, 25, 70}
- `valors` = {10.0, 20.0, 50.0, 40.0, 30.0, 80.0}

l'array resultat seria {1, 3, 4, 3, 2, 5}, ja que hi ha:

- 1 valor menor que 15 (10.0)
- 3 valors menors que 35 (10.0, 20.0, 30.0)
- 4 valors menors que 50 (10.0, 20.0, 30.0, 40.0)
- 3 valors menors que 37 (10.0, 20.0, 30.0)
- 2 valors menors que 25 (10.0, 20.0)
- 5 valors menors que 70 (10.0, 20.0, 30.0, 40.0, 50.0)

### RecP2 - Curs 18/19: 2 punts

### RECORREGUT DESCENDENT

**Es demana:** escriure un mètode estàtic que tinga com a paràmetre un array `a` de `char` i que escriga en l'eixida estàndard, línia a línia, tots els *suffixos* de la cadena de caràcters en `a`, des del més curt endavant. S'entén per sufix qualsevol subcadena que comprèn els caràcters des d'un donat fins a l'últim inclusivament. Per exemple, si `a` és {'s','t','a','t','i','c'}, s'ha d'escriure:

```
c
ic
tic
atic
tatic
static
```

### P2 - Curs 17/18: 2 punts

### CERCA

**Es demana:** escriure un mètode estàtic que donat un array de `double` amb almenys una component, retorne `true` quan totes les parelles de components simètriques en l'array sumen el mateix, i `false` en cas contrari. En el cas en què l'array tinga longitud senar s'ha de considerar que l'element central és simètric d'ell mateix (i.e., s'ha de sumar amb ell mateix). Exemples:

Per als arrays {3.0,2.9,2.1,2.0}, {-1.0,2.0,-1.0,-4.0,-1.0}, {3.0,-2.0,-1.0,-6.0}, {1.3,4.5} o {4.5} ha de retornar `true`;

per a {1.0,2.5,5.0}, {1.0,2.5,4.0,2.0,3.0}, {1.0,2.4,4.6,-3.0}, o {-1.0,2.0,-2.0,-4.0,-1.0} ha de retornar `false`.

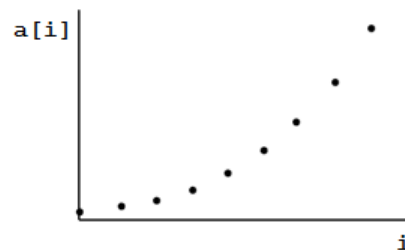
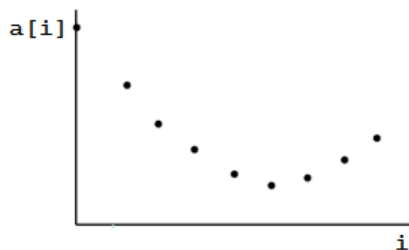
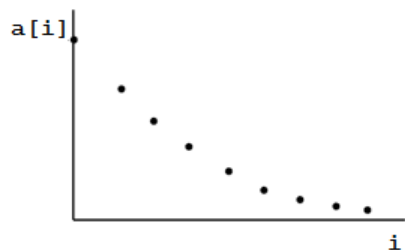
### RecP2 - Curs 17/18: 2 punts

### RECORREGUT + CERCA

**Es demana:** implementar un mètode estàtic què, donat un array de nombres enters, determine si és *Par Dominant*, és a dir, si la suma del valor absolut de cadascun dels elements que ocupen posicions parelles, és major que tots i cadascun dels elements, en valor absolut, que ocupen posicions senars.

**P2 - Curs 16/17: 2 punts****CERCA**

Siga un array  $a$  de reals i longitud  $n \geq 2$ , tal que les seues components s'ajusten al perfil d'una corba còncava, és a dir, hi ha un mínim en una certa posició  $k$ ,  $0 \leq k < n$  (açò és, els valors en  $a[0..k]$  són estrictament decreixents i els valors en  $a[k..n-1]$  són estrictament creixents); el mínim es pot trobar en un dels extrems de l'array. **Es demana:** implementar un mètode estàtic que, donat l'array, torne la posició del mínim. Per exemple, per als arrays de les següents figures, el mètode hauria de tornar 8, 5 i 0, respectivament.

**RecP2 - Curs 16/17: 2 punts****RECORREGUT { CERCA }**

**Es demana:** implementar un mètode estàtic que, donats dos arrays de caràcters  $a$  i  $b$ , tots dos sense repetits, torne el nombre de caràcters comuns. Per exemple, si  $a$  és {'C', 'T', 'A', 'G'} i  $b$  és {'T', 'U', 'C'}, el mètode ha de tornar 2.

**P2 - Curs 15/16: 1.75 punts****CERCA**

**Es demana** escriure un mètode Java estàtic que, donat un array de `double`, comprove si les components d'índex parell apareixen ordenades ascendentment. Exemples:

Per a 

0	1	2	3	4	5
1.5	0.0	3.0	-1.0	3.5	2.0

 i 

0	1	2	3	4	5	6
3.0	0.0	4.5	-1.0	6.5	2.0	8.5

 ha de donar `true`.

Per a 

0	1	2	3	4	5
1.5	0.0	3.0	-1.0	1.5	2.0

 i 

0	1	2	3	4	5	6
3.0	0.0	1.0	-1.0	6.5	2.0	8.5

 ha de donar `false`.

**RecP2 - Curs 15/16: 2 punts****RECORREGUT DESCENDENT**

**Es demana:** escriure un mètode de classe (`static`) que, donat un enter no negatiu, retorne un array de `boolean`, de longitud igual al nombre de dígit de l'enter donat, tal que l'element de la posició  $i$  de l'array és `true` quan el dígit de la posició  $i$  de l'enter (numerant d'esquerra a dreta) és múltiple de 3, i `false` en un altre cas.

Per exemple:

Per a 357 ha de tornar 

0	1	2
true	false	false

 Per a 1609 ha de tornar 

0	1	2	3
false	true	true	true

**P2 - Curs 14/15: 1.75 punts****RECORREGUT + RECORREGUT**

**Es demana:** Implementar un mètode de classe (o estàtic) que reba com paràmetres un array d'enters  $a$  ( $a.length > 0$ ) i un enter  $p$  que representa una posició vàlida dins l'array ( $0 \leq p < a.length$ ). El mètode ha de tornar el valor màxim de les sumes dels elements de l'array en les posicions prèvies i posteriors a la posició donada, sense incloure l'element que ocupa aquesta posició en els càlculs. Per exemple, donat l'array {1, 7, -2, 3, 4, 8, 1, -4} i la posició 2, tornarà el màxim entre  $1 + 7 = 8$  i  $3 + 4 + 8 + 1 - 4 = 12$ , és a dir, 12.

**RecP2 - Curs 14/15: 1.75 punts****RECORREGUT { RECORREGUT }**

**Es demana:** Implementa un mètode de classe (o estàtic) tal que, donats un array  $a$  de `double` ( $a.length > 0$ ) i un enter  $n$  ( $n > 0$ ), modifique tots els elements de l'array de forma que el valor màxim del mateix siga  $n$  i la resta estiguen escalats amb aquest valor. És a dir, si el màxim de  $a$  és  $m$ , la relació entre el valor nou i l'antic d'un component de l'array ha de ser  $n/m$ . Per exemple, si l'array  $a$  és {4.5, 27.0, 18.0, 1.5} i  $n = 9$ ,  $a$  ha de canviar a {1.5, 9.0, 6.0, 0.5}.