# Exercicis
# Introducció a la Informàtica
# Grau d'Estadística

Jaume Baixeries
Natàlia Pallarès
Sara Pérez
Enrique Romero

16 d'octubre de 2015

# Índex

# Capítol 1

# Problemes Bàsics

---

**Problema** 1. *Preguntes bàsiques: suma.*

Escriu un programa que demani 3 valors per l'entrada estàndard i n'escrigui la suma per pantalla.

---

**Problema** 2. *Preguntes bàsiques: mitjana.*

Escriu un programa que demani 3 valors per l'entrada estàndard i n'escrigui la mitjana per pantalla.

---

**Problema** 3. *Preguntes bàsiques:programa.*

Què escriu per la sortida estàndard aquest programa?

```
x <- 20
y <- 30
z <- (x + y) / 2
cat ("el resultat entre ",x," i ",y," és ",x+y,"\n")
```

---

**Problema** 4. *Preguntes bàsiques.*

Què escriu per la sortida estàndard aquest programa quan hi entrem 4 per l'entrada estàndard? I si hi entrem 23?

```
x <- scan(n=1,quiet=TRUE)
x <- x %% 2
cat (x,"\n")
```

---

**Problema** 5. *Preguntes bàsiques.*
Què escriu per la sortida estàndard aquest programa quan entrem 20 i 30 per l'entrada estàndard?

```
x <- scan(n=1,quiet=TRUE)
y <- scan(n=1,quiet=TRUE)
cat (x > y && x > 0,"\n")
```

---

**Problema** 6. *Preguntes bàsiques.*
Què escriu per la sortida estàndard aquest programa quan entrem 2 i 20 per l'entrada estàndard?

```
x <- scan(n=1,quiet=TRUE)
y <- scan(n=1,quiet=TRUE)
ex <- x > 0 && x < 21 && y %% 2 == 0
cat (ex,"\n")
```

---

**Problema** 7. *Programa.*
Què escriu per la sortida estàndard aquest programa?

```
x <- 20
y <- 20
cat (x,y,x+y,"\n")
```

---

**Problema** 8. *Programa.*
Què escriu per la sortida estàndard aquest programa quan hi entrem 4 per l'entrada estàndard? I si hi entrem 23?

```
cat("Escriu un valor entre 1 i 20")
x <- scan(n=1,quiet=TRUE)
cat (x %% 20,"\n")
```

---

**Problema** 9. *Instruccions.*
Què escriu per la sortida estàndard aquest programa quan entrem 20 i 30 l'entrada estàndard?

```
x <- scan(n=1,quiet=TRUE)
x <- scan(n=1,quiet=TRUE)
cat (x,"\n")
```

---

**Problema** 10. *Parell més a prop.*

Feu un programa que, donat un enter, escrigui el parell que té més a prop seu. Exemple: si entra el 23, escriu el 24 (o el 22). Si entra el 22, escriu el 22.

---

**Problema** 11. *Seqüència.*

Donat tres enters, escriu `TRUE` per la sortida estàndard si i només si els enters fan una seqüència creixent o decreixent.

---

**Problema** 12. *Doble.*

Donats 2 enters, fes un programa que escrigui `TRUE` per la sortida estàndard si i només si un dels dos enters és més gran que el doble de l'altre.

---

**Problema** 13. *Segon grau.*

Fes un programa que, donats els quoficients d'una equació de segon grau, en calculi les arrels (assumeix que els coeficients no donen problemes).

---

**Problema** 14. *Divisors.*

Donat un enter, fes un programa que escrigui `TRUE` si i només si té almenys un divisor senar més petit que 10.

---

**Problema** 15. *Intercanvi de variables.*

Donats dos enters, fes un programa que intercanviï el valor de les dues variables.

---

**Problema** 16. *Segons.*

Donat un nombre de segons, escriu els nombre d'hores, minuts i segons que representen per la sortida estàndard.

---

**Problema** 17. *Hora.*
Donat un nombre d'hores, un de minuts i un de segons, escriu per la sortida estàndard el nombre de segons que representen.

---

**Problema** 18. *Canvi.*
Donat un import en euros, fes un programa que escrigui la descomposició en bitllets de 100, 50, 20, 10, 5 i monedes d'1 euro.

---

**Problema** 19. *Sobre un nombre.*
Donat un enter, fes un programa que escrigui `TRUE` si i només si l'enter és múltiple de 3 i més gran que 30.

---

**Problema** 20. *Múltiples?.*
Donats dos enters, fes un programa que escrigui `TRUE` si i només si un dels dos enters és múltiple de l'altre.

---

**Problema** 21. *Pesos.*
Donat un enter més gran que 10, fes un programa que escrigui `TRUE` si i només si la segona xifra (començant per la dreta) és el número 7.

---

**Problema** 22. *Dígits múltiples.*
Donat un enter més gran que 100 i menor que 1000, fes un programa que escrigui `TRUE` si i només si la suma de tots 3 dígits és múltiple de 5.

---

**Problema** 23. *Alarmes.*

Siguin $b_1, b_2, b_3$ tres variables booleanes que representen tres alarmes diferents. Se suposa que hi ha alarma perillosa quan **una sola** de les alarmes està activada (és `TRUE`). Fes un programa que, donades les 3 variables, escrigui `TRUE` si i només si hi ha una alarma perillosa.

---

**Problema** 24. *Classificació.*

Per a dos equips de futbol, tenim els punts, els gols a favor i els gols en contra. També tenim el resultat dels dos partits que els han enfrontat a la lliga. Fes un programa que escrigui `TRUE` si i només si el primer equip va abans que el segon a la classificació.

A la classificació tenim en compte els punts. En cas d'empat a punts, qui tingui una diferència de gols més gran, i en cas d'empat, qui tingui més gols en el global de tots dos partits que han jugat els equips.

---

**Problema** 25. *Cap-i-cua.*

Donat un enter més gran que 99 i menor que 1000, feu un programa que escrigui `TRUE` si i només si és un número cap-i-cua.

---

**Problema** 26. *Interval.*

Feu un programa que escrigui `TRUE` si i només si un valor que entreu pel teclat es troba dins de l'interval $\{0, 5\}$

# Capítol 2

# Alternativa

---

**Problema** 27. *Operació.*

Feu un programa que, donats un caràcter (corresponent a un dels caràcters: '+', '-', '*' o '/') i dos enters, calculi el resultat de fer l'operació demanada entre ells.

---

**Problema** 28. *Equació de 2on grau.*

Fes un programa que que avaluï una equació de segon grau. Els paràmetres són els coeficients de l'equació.

---

**Problema** 29. *Valor absolut.*

Feu un programa que calculi el valor absolut d'un enter que entreu pel teclat.

---

**Problema** 30. *Màxim de 3.*

Feu un programa que calculi el màxim de 3 valors que entreu pel teclat.

---

**Problema** 31. *Per sobre la mitjana.*

Feu un programa que, donats 4 valors que entreu pel teclat, digui quants d'aquests valors són més grans que la mitjana de tots quatre.

---

**Problema** 32. *En ordre.*

Feu un programa que, donats 3 valors per teclat, els imprimeixi en ordre per la pantalla.

---

**Problema** 33. *Freqüència.*

Feu un programa que, donades 3 paraules, digui quantes vegades apareix la més freqüent (òbviament, poden estar repetides).

# Capítol 3

# Funcions

---

**Problema** 34. *Funció per arrels.*

Feu una funció que, donats 3 reals (que representen els coeficients d'una equació de segon grau) en torni les 2 arrels (en cas que existeixin).

---

**Problema** 35. *Mcd i mcm.*

Donats dos nombres naturals feu una funció que calculi el mcd i una altra que en calculi el mcm.

---

**Problema** 36. *Funció amb dígits.*

Feu una funció que, donat un enter, torni la suma dels seus dígits i el nombre de dígits que té.

# Capítol 4

# Iteracions

---

**Problema** 37. *3 xifres.*
Escriviu un programa que mostri tots els nombres de tres xifres tals que la suma de les centenes més les desenes sigui igual a les unitats.

---

**Problema** 38. *Parells Petits.*
Escriviu un programa que a partir d'un natural $n$ llegit del canal d'entrada, mostri tots els nombres parells més petits que $n$ pel canal de sortida.

---

**Problema** 39. *Potència de 7.*
Escriviu un programa que calculi la potència de 7 més propera a 1000 per excés (no podeu fer servir l'operació de potència ($x^y$): `x ** y`).

---

**Problema** 40. *Euclides.*
Codifiqueu l'algorisme d'Euclides, el qual permet determinar el màxim comú divisor de dos nombres enters positius $a$ i $b$ de la manera següent:

1. Si $a$ i $b$ són iguals, doneu el valor d'$a$ com a resultat i finalitzeu.
2. Si $a$ i $b$ són diferents, canvieu el valor del més gran dels valors d'$a$ i $b$ per la diferència entre el valor més gran i el més petit, i torneu al pas 1.

---

**Problema** 41. *Successió.*

Escriviu un programa que a partir d'un valor enter donat $N$, que se suposa més gran que 1, i d'un valor real $x$, calculi la suma dels $N$ primers termes de la successió següent:

$$\sum_{i=0}^{N-1} \frac{x^i}{2^i}$$

No podeu fer servir l'operació de potència ($x^y$): `x ** y`. Exemple: $N = 3, x = 3$

$$1 + \frac{x}{2} + \frac{x^2}{4} = 1 + \frac{3}{2} + \frac{9}{4} = 1 + 1.5 + 2.25 = 4.75$$

---

**Problema** 42. *Quadrat perfecte.*

Escriviu un programa que donat un enter, digui si és quadrat perfecte. Es diu que un enter és un quadrat perfecte si existeix un altre enter que elevat al quadrat dóna el primer. Per exemple, 4 és un quadrat perfecte perquè és el resultat d'elevar 2 al quadrat; 5 no ho és perquè no es pot obtenir elevant al quadrat cap nombre enter. No podeu fer servir l'operació de potència ($x^y$): `x ** y`.

---

**Problema** 43. *Nombre de xifres.*

Feu un programa que, donat un enter més gran que zero, calculi el nombre de xifres que conté. Exemple: l'enter 4099 està format per 4 xifres (4, 0, 9 i 9).

---

**Problema** 44. *Suma de xifres.*

Donat un enter, feu un programa que calculi la suma de les seves xifres.

---

**Problema** 45. *Dígits invertits.*

Donat un enter positiu, feu un programa que calculi un enter $x$ resultat d'invertir els dígits que formen el nombre. Exemple: $capgirar(7412) = 2147$ Important: cal calcular una variable que contingui el valor 2147, no simplement treure els dígits en pordre per pantalla.

---

**Problema** 46. *Divisors.*

Feu un programa que mostri per pantalla tots els divisors d'un natural donat.

---

**Problema** 47. *Suma de múltiples.*

Feu un programa que, donat un enter positiu n, torni la suma dels n primers múltiples de 3.

Exemple: n = 4. Valor retornat per la funció = 3 + 6 + 9 + 12 = 30

---

**Problema** 48. *Dígit repetit.*

Dissenyeu un programa que, donat un enter positiu, torni quantes vegades conté el dígit 7. Exemples: per l'enter 1983 tornaria 0, per l'enter 207463727 tornaria 3.

---

**Problema** 49. *Cub perfecte.*

Dissenyeu un programa que, donat un nombre natural n, digui si és un cub perfecte, és a dir, si existeix un altre natural m tal que $3 * m = n$.

---

**Problema** 50. *Nombres triangulars.*

Feu un programa que, donat un enter n, escrigui els primers n nombres triangulars. La sèrie matemàtica dels nombres triangulars es defineix com:

$$x_1 = 1$$

$$x_{k+1} = x_k + k + 1$$

Exemples dels primers termes de la sèrie són:

$$x_1 = 1$$

$$x_2 = x_1 + 1 + 1 = 1 + 1 + 1 = 3$$

$$x_3 = x_2 + 2 + 1 = 3 + 2 + 1 = 6$$

$$x_4 = x_3 + 3 + 1 = 6 + 3 + 1 = 10$$

---

**Problema** 51. *Successió de Fibonacci.*

Donat un enter $n$, calculeu els `n` primers nombres de la successió de Fibonacci, definida com aquella tal que els dos primers nombres són 1, i on cada element a partir del tercer es calcula sumant els dos anteriors: 1, 1, 2, 3, 5, 8, 13, 21, ...

---

**Problema** 52. *Càlcul del Sinus.*

Feu una funció tal que donat, un nombre real $x$, i un enter $n$, aproximi la funció $\sin(x)$ amb $n$ termes ($n$ sumands). Cal que feu l'aproximació de $\sin(x)$ per la sèrie de Taylor. S'ha de tenir present que quan $i$ ($i$=1,2,...,n) és parell el sumant de Taylor pren per valor 0.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \tag{4.1}$$

La funció ha de tenir la següent capçalera:

La funció retorna el $\sin(x)$ avaluat en n termes de sèrie de Taylor. Fixeu-vos que per a un $n$ prou gran, la funció aproxima molt bé el resultat real de $\sin(x)$ que es pot obtenir amb qualsevol calculadora o amb la mateixa funció *sin* que té l'R.

---

**Problema** 53. *22 d'octubre del 2012: Més sèries.*

Una sèrie numèrica es defineix com:

$$x_1 = 1$$

$$x_i = \begin{cases} x_{i-1} + 5, & \text{si } i \text{ és múltiple de 3} \\ 7x_{i-1}, & \text{si } i \text{ és múltiple de 7} \\ x_{i-1} + 1, & \text{si } i \text{ altrament} \end{cases}$$

Feu un programa que, donat un enter $n$, calculi la suma dels primers $n$ termes d'aquesta sèrie. Per exemple, tindríem que:

$$x_2 = x_{i-1} + 1 = x_1 + 1 = 1 + 1 = 2$$

$$x_3 = x_{i-1} + 5 = x_2 + 5 = 2 + 5 = 7$$

$$x_4 = x_{i-1} + 1 = x_3 + 1 = 7 + 1 = 8$$

**Problema** 54. *Sumes de múltiples d'un nombre.*

Donat un enter $x$ (que és el nombre del qual volem calcular múltiples) i un altre enter $n$ (que és el nombre de múltiples que volem calcular), feu un programa que calculi els $n$ primers múltiples de $x$ i retorni dos enters que siguin la suma dels múltiples senars i la suma dels múltiples parells.

El programa només ha de retornar la suma dels múltiples parells i la suma dels múltiples senars.

# Capítol 5

# Seqüències

---

**Problema** 55. *Seqüència d'entrada.*

Feu un programa que llegeixi una seqüència amb valors enters (acabada en 0) llegits de teclat i la mostri per pantalla.

---

**Problema** 56. *Mitjana d'una seqüència.*

Calcula la mitjana d'una seqüència. (la seqüència pot tenir tots dos formats: mida + seq o seq + indicador de final seq).

---

**Problema** 57. *Màxim i mínim d'una seqüència.*

Calcula el màxim i el mínim d'una seqüència que entrem per teclat (la seqüència pot tenir tots dos formats: mida + seq o seq + indicador de final seq).

---

**Problema** 58. *Parells d'una seqüència.*

Calcula el nombre de nombres parells que té una seqüència. (la seqüència pot tenir tots dos formats: mida + seq o seq + indicador de final seq).

---

**Problema** 59. *Positius versus negatius.*
Donada una seqüència d'enters acabada en 0, feu un programa que indiqui si hi ha més valors positius que negatius.

---

**Problema** 60. *Posició dels negatius.*
Donada una seqüència d'enters acabada en 0, escriviu un programa que mostri la posició de tots els valors negatius. En cas que no hi hagi cap valor negatiu, el programa ha de mostrar un missatge indicant-ho.

---

**Problema** 61. *Atletisme.*
Donada una seqüència de nombres reals acabada en 0, el primer dels quals és el rècord del món (en segons) en la prova de 100 metres llisos i els valors següents, que no estan ordenats, són els temps (en segons) enregistrats en la mateixa prova per un grup no buit d'atletes durant una competició, escriviu un programa que calculi:

1. el nombre d'atletes que han participat de la prova,
2. la millor marca enregistrada i
3. decidir si s'ha batut o no el rècord del món.

---

**Problema** 62. *3 xifres.*
Escriviu un programa que donat una seqüència d'entrada de zeros i uns acabada en -1, indiqui si conté més zeros que uns o més uns que zeros. En cas d'igualtat, el missatge a mostrar serà "Empat". Per exemple, si l'entrada fos [0, 1, 0, 1, 0], el programa indicaria: "Hi ha més zeros que uns".

---

**Problema** 63. *Vegades repetit.*
Donats un enter $N$ i una seqüència d'enters positius acabada en zero, escriviu un programa que calculi quantes vegades apareix $N$ a la seqüència d'entrada. Feu-ne dues versions, una suposant que els elements estan ordenats i una altra suposant que no ho estan.

---

**Problema** 64. *Elements iguals en seqüència.*
Feu un programa que, donada una seqüència d'entrada acabada en 0, escrigui `TRUE` si i només si tots els elements de la seqüència són iguals.

---

**Problema** 65. *Elements iguals a un altre en seqüència.*
Escriviu un programa que, donada una seqüència d'entrada acabada en 0 i un enter, escrigui `TRUE` si i només si tots els elements de la seqüència són iguals a l'enter donat.

---

**Problema** 66. *Parell en seqüència.*
Donada una seqüència d'enters positius acabada en 0, feu un programa que retorni `TRUE` si la seqüència conté almenys un nombre parell i `FALSE` altrament.

---

**Problema** 67. *Primer negatiu.*
Donada una seqüència d'enters acabada en 0, feu un programa que digui en quina posició es troba el primer enter negatiu. En cas que no hi hagi cap valor negatiu, el programa ha de mostrar un missatge indicant-ho.

---

**Problema** 68. *Seqüència creixent?.*
Escriviu un programa que donada una seqüència d'enters acabada en 0, comprovi si és creixent.

---

**Problema** 69. *Temperatures.*
Dissenyeu un programa que, a partir de les temperatures mínimes enregistrades durant un període donat de n dies (que se suposa més gran que 2), numerats de 1 a n, mostri quins han estat els primers dies en què s'enregistraren temperatures mínimes negatives durant dos dies consecutius. En cas que no existeixin, cal indicar-ho.

Exemple: Dades: $\mathrm{n} = 7$, temperatures $= 4.2, 1.5, -0.5, -1.8, 2.1, -0.8, -1.3$. Resultat: 3 i 4 són els primers dos dies consecutius amb temperatures negatives.

---

**Problema** 70. *Negatiu en seqüència.*
Escriviu un programa que donada una seqüència d'enters acabada en 0, digui si hi ha cap valor negatiu.

---

**Problema** 71. *Valors en interval.*
Dissenyeu un programa que llegeixi pel canal estàndard d'entrada valors enters (on el primer valor és el nombre d'enters que li passarem), comprovi que no conté cap valor fora de l'interval $[0, 10]$ i mostri per pantalla el missatge que li correspongui: "No conté cap element fora de l'interval [0,10]"o bé "Conté algun element fora de l'interval [0,10]".

---

**Problema** 72. *Seqüència creixent o decreixent.*
Donada una seqüència, feu un programa que digui si una seqüència és estrictament creixent o estrictament decreixent.

---

**Problema** 73. *O parells o senars.*
Donada una seqüència, feu un programa que digui si tots els elements de la seqüència són parells, o bé si són tots senars. No podeu repetir codi.

---

**Problema** 74. *Codificació correcta.*
Sigui una seqüència d'entrada que conté únicament els números 1,2 o 3. El final de seqüència és el 0.

Una seqüència correcta és la que segueix totes les regles següents:

1. Després d'un 1 només hi pot haver un 2.
2. Després d'un 2 només hi pot haver un 1 o un 3.
3. Després d'un 3 només hi pot haver un 1 o un 2.

Per exemple, la seqüència:

```
1 2 1 2 3 1 2 3 2 3 0
```

és correcta, mentre que la seqüència:

```
1 2 1 2 3 3 2 3 2 3 0
```

no ho és. Feu un programa que, donada una seqüència, digui si és correcta.

---

**Problema** 75. *Codificació interessant.*
Sigui una seqüència d'entrada que conté únicament números naturals. El final de seqüència és el -1.
Una seqüència interessant és la que segueix totes les regles següents:

1. Després d'un parell només hi pot haver un senar o un parell més petit.
2. Després d'un 33 només hi pot haver un múltiple de 3.
3. Després d'un múltiple de 5, només hi pot haver un número més gran que 100.

Les condicions són additives. Per exemple, al 50 se li aplicarien les condicions 1 i 3. Si un número no té cap restricció, llavors pot tenir qualsevol altre número.
Per exemple, la seqüència:

```
2 33 30 129 333 5 1000 -1
```

és correcta, mentre que la seqüència:

```
4 2 33 90 92 1 -1
```

no ho és. Feu un programa que, donada una seqüència, digui si és interessant.

---

**Problema** 76. *Pics (Parcial 8 de novembre del 2013).*
Sigui una seqüència d'enters, d'almenys tres elements, que acaba amb un -1 (que no forma part de la seqüència, simplement en marca el final). Un **pic** és un element $x_i$ de la seqüència que compleix la següent condició:

$$x_{i-1} < x_i > x_{i+1}$$

Feu un programa que calculi quants pics té una seqüència d'entrada. Per exemple, amb l'entrada:

```
3 4 5 4 5 1 2 3 1 -1
```

el programa escriuria 3 per pantalla. En aquest cas, els pics són 5, 5 i 3.

---

**Problema** 77. *Creixent-Decreixent (Parcial 8 de novembre del 2013).*

Siguin dues seqüències de la mateixa mida (i no buides, és a dir, almenys hi ha un element) que es donen intercalades. Per exemple, si $X = 3, 4, 7, 2, 4$ i $Y = 5, 9, 6, 5, 1$ llavors l'entrada serà:

`5 3 5 4 9 7 6 2 5 4 1`

on 5 serà la mida de totes dues seqüències, 3 serà el primer element de $X$, 5 el primer de $Y$, 4 el segon element de $X$ i 9 el segon de $Y$, etc.

Feu un programa que escrigui a la pantalla `SÍ` si la seqüència $X$ és creixent i la seqüència $Y$ decreixent, o bé que escrigui `NO` altrament. Per exemple, amb la següent entrada:

`3 3 5 4 4 7 1`

el programa escriuria `SÍ`, mentre que amb la següent entrada:

`3 3 5 4 4 7 5`

el programa escriuria `NO` perquè la seqüència $Y$ (5,4,5) no és decreixent.

---

**Problema** 78. *Saldo (Final 23 de gener del 2015).*

Sigui una **seqüència** (per tant, no feu servir vectors!):

$$N \;\; imp_1 \;\; dia_1 \;\; imp_2 \;\; dia_2 \;\; \ldots \;\; imp_N \;\; dia_N$$

tal que $N$ és el nombre de parells ($imp_i \;\; dia_i$) que té la seqüència. Cada parell ($imp_i \;\; dia_i$) representa un import $imp_i$ que es fa el dia $dia_i$ en un compte corrent. Òbviament, $imp_i$ pot ser positiu (ingrés) o negatiu (reintegrament).

La seqüència **sempre** complirà les següents condicions (per tant, no cal que les comproveu):

1. No hi ha cap dia repetit: com a màxim, hi ha una transacció per dia.
2. La seqüència de transaccions representa que es fa en dies creixents, és a dir, si $i < j$, llavors tenim que $dia_i < dia_j$.
3. $dia_1 = 1$ i $dia_N = 30$.
4. Encara que el saldo a dia 30 sigui negatiu, no cal comptar el dia 30 com a descobert.

Feu un programa que llegeixi una seqüència com la que indica l'enunciat, escrigui per pantalla quants dies ha estat el compte corrent en **descobert**, és a dir, quants dies ha estat el compte corrent amb saldo negatiu. Assumiu que el compte té zero euros al principi, i que es considera descobert quan el compte té saldo negatiu (no pas zero o positiu) a partir de l'endemà de fer l'operació.

Per exemple, amb la seqüència:

$$7 \;\; 40 \;\; 1 \;\; -37 \;\; 5 \;\; 20 \;\; 7 \;\; -40 \;\; 17 \;\; 17 \;\; 28 \;\; -50 \;\; 29 \;\; 10 \;\; 30$$

tindrem que haurem estat amb el saldo en descobert dels dies 17 al 28 (amb -17 euros), és a dir, 11 dies, i del dia 29 al 30 (amb -50 euros), és a dir, 1 dia. Per tant, el programa haurà d'escriure 12.

Amb la seqüència

$$4 \quad -10 \quad 1 \quad 20 \quad 6 \quad -30 \quad 8 \quad 40 \quad 30$$

haurem estat en descobert dels dies 1 al 6 (5 dies) i del 8 al 30 (22 dies), i per tant, el programa haurà d'escriure 27.

---

**Problema** 79. *Pes Compressió (Final 7 de juliol del 2015).*

Feu un **programa** tal que, donada una **seqüència** amb almenys dos elements al canal d'entrada (teclat) composta per nombres naturals estrictament més grans que zero, i que acaba amb un zero (que marca el final de seqüència), **escrigui** pel canal de sortida (pantalla) la **compressió de la seqüència**.

La compressió d'una seqüència consisteix en dir la mida i l'element que composa cada subseqüència d'elements iguals consecutius. Per exemple, donada la seqüència:

$$[5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 1, 2, 4, 4, 4, 0]$$

la compressió d'aquesta seqüència és:

$$(5 : 8)(4 : 2)(1 : 1)(2 : 1)(4 : 3)$$

ja que:

$$[\underbrace{5, 5, 5, 5, 5, 5, 5, 5}_{8}, \underbrace{4, 4}_{2}, \underbrace{1}_{1}, \underbrace{2}_{1}, \underbrace{4, 4, 4}_{3}, 0]$$

**Molt important**: òbviament, per a la resolució d'aquest problema **no podeu fer servir vectors**. Si apareix un vector a la vostra solució, l'exercici tindrà un zero.

# Capítol 6

# Vectors

---

**Problema** 80. *Vector d'entrada.*

Feu un programa que creï un vector amb valors enters llegits de teclat (acabat en zero) i després el mostri per pantalla.

---

**Problema** 81. *Suma de vectors (I).*

Feu una funció que, donats dos vectors amb valors reals retorni el vector resultant de la suma de tots dos.

---

**Problema** 82. *Suma de vectors (II).*

Donats dos vectors amb valors reals, feu una funció que obtingui el vector resultant de la suma de tots dos, element a element.

---

**Problema** 83. *Càlculs sobre vector.*

Donat un vector amb valors enters, feu una funció que calculi quants elements són més grans que la mitjana, quants són més petits i quants són iguals.

---

**Problema** 84. *A l'inrevés.*

Escriu una funció que escrigui per pantalla els elements d'un vector en ordre invers.

---

**Problema** 85. *Element inserit.*
Feu una funció que rebi un vector, un element i una posició dins del vector, i que torni un vector que sigui el resultat d'inserir l'element a la posició en el vector. Per exemple:

---

**Problema** 86. *Posicions parelles.*
Feu una funció que torni els elements d'un vector que tenen els subíndexos parells (no els elements parells, sinó els que són en *posició* parella).

---

**Problema** 87. *Elements iguals en vector.*
Feu una funció que donat un vector, retorni `TRUE` si i només si tots els elements són iguals.

---

**Problema** 88. *Elements iguals a un altre en vector.*
Escriviu una funció que, donats un vector i un enter, digui si tots els elements del vector (seqüència d'entrada) són iguals a l'enter donat.

---

**Problema** 89. *Producte escalar.*
Donats dos vectors amb valors reals, escriviu una funció que en calculi el producte escalar.

---

**Problema** 90. *Vegades repetit.*
Donats un vector i un enter, escriviu una funció que calculi quantes vegades apareix l'enter donat al vector. Feu-ne dues versions, una suposant que els elements estan ordenadats i una altra suposant que no ho estan.

**Problema** 91. *Unió.*

Feu un funció que, donats dos vectors amb valors enters, retorni un nou vector amb la seva unió (operació de conjunts). Suposarem que els vectors d'entrada, individualment, no contenen elements repetits.

Nota: La unió és una operació entre conjunts. Aquesta operació crea un conjunt, anomenat conjunt unió, al qual pertanyen tots els elements que pertanyen a qualsevol dels conjunts que s'uneixen. S'expressa amb el símbol $\cup$. Per exemple, donat $A = 1, 2, 3, 10$ i $B = 1, 2, 6, 8$, si definim $C = A \cup B$, llavors $C = 1, 2, 3, 10, 6, 8$.

---

**Problema** 92. *Intersecció.*

Feu una funció que, donats dos vectors amb valors enters, retorni un nou vector amb la seva intersecció (operació de conjunts). Suposarem que els vectors d'entrada, individualment, no contenen elements repetits.

Nota: La intersecció és una operació entre conjunts. Aquesta operació crea un conjunt, anomenat conjunt intersecció, al qual pertanyen tots els elements que pertanyen a la vegada a tots els conjunts que s'interseguen. S'expressa amb el símbol $\cap$. Per exemple, donat $A = 1, 2, 3, 10$ i $B = 1, 2, 6, 8$, si definim $C = A \cap B$, llavors $C = 1, 2$.

---

**Problema** 93. *Palíndrom (cap-i-cua).*

Donats dues cadenes de caràcters, que representen dues paraules o frases diferents, feu una funció que torni `TRUE` si i només si són bifronts o falsos palíndroms.

Per exemple, `català` i `a l'atac` ho són (si en treiem els espais i símbols de puntuació). També ho són `Roma` i `amor`.

---

**Problema** 94. *Parell en vector.*

Donada un vector d'enters positius, feu una funció que torni `TRUE` si i només si conté almenys un nombre parell.

**Problema** 95. *Màxim en vector.*
Donat un vector d'enters, feu una funció que en torni el valor màxim.

**Problema** 96. *Zeros i uns.*
Feu una funció que, donada una mida, creï i ompli un vector amb zeros i uns. Els valors del vector seran 0 i 1, alternats, començant pel 0. Per exemple, si es demana un vector de longitud de 5, generaria: [0, 1, 0, 1, 0].

**Problema** 97. *Divisor en un vector.*
Feu una funció que, donats un vector i un enter, torni `TRUE` si i només si l'enter donat és divisor d'algun dels elements del vector.

**Problema** 98. *Tots parells?.*
Feu una funció que donat un vector amb valors enters positius, torni `TRUE` si i només si tots els seus valors són parells.

**Problema** 99. *Successió de Fibonacci.*
Donat un enter (`n`), calcula els `n` primers nombres de la successió de Fibonacci, definida com aquella tal que els dos primers nombres són l'1 i el 2, i on cada element a partir del tercer es calcula sumant els dos anteriors: 1, 2, 3, 5, 8, 13, 21, ...

**Problema** 100. *Alfa-mitjana.*
Donat un vector d'enters i un enter $x \in \{0, 100\}$, calcula la mitjana dels valors en el vector, eliminant-ne els valors que es troben (desprès de ser ordenat) en el $x\%$ del principi del vector i en el $x\%$ de la cua del vector.
Per exemple, tenim un vector:

$$v = [3, 6, 2, 3, 1, 3, 5, 4, 1, 8]$$

i tenim que $x = 20$ (és a dir, que cal eliminar el 20% dels primers elements i el 20% dels elements finals prèvia ordenació). Per tant, el vector, un cop ordenat ens queda:

$$v = [1, 1, 2, 3, 3, 3, 4, 5, 6, 8]$$

i retallant-li el 20% al principi i el final, ens queda:

$$v = [2, 3, 3, 3, 4, 5]$$

vector del qual caldrà fer-ne la mitjana.

Entrada: percentatge (en un enter del 0 al 100) que cal retallar i el vector d'elements.

La sortida ha de ser la mitjana del vector ordenat i retallat segons el percentatge de l'entrada. El percentatge serà sempre inferior al 50%.

---

**Problema** 101. *Correlació de Pearson.*

Donat dos vectors $x$ i $y$, fes una funció amb la següent capçalera:

que calculi el coeficient de correlació de Pearson. Cal escriure només la funció. El programa principal serà ignorat. El coeficient de correlació de Pearson es calcula com a:

$$\frac{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

La funció tindrà com a paràmetres dos vectors de reals: $x, y$.

---

**Problema** 102. *Mediana.*

Fes un programa tal que, donat un vector d'enters, en tregui la mediana per pantalla.

---

**Problema** 103. *9 de febrer del 2012: Distància més curta.*

Feu la funció:

```
dist_mes_curta <- function(x,y)
```

Aquesta funció fa el següent: rep un parell de vectors $x$ i $y$ que tenen coordenades en un pla en dues dimensions (la coordenada del punt $p_i$ és $(x_i, y_i)$). Òbviament la mida de tots dos vectors és la mateixa. La funció ha de retornar quina és la distància (euclidiana) més curta que hi ha entre dos punts **diferents** de tot el conjunt de punts. Formalment, torna:

$$min(\{d(p_i, p_j) \mid i \neq j\})$$

La distància euclidiana entre dos punts $p_i = (x_i, y_i)$ i $p_j = (x_j, y_j)$ es defineix com a:

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

---

**Problema** 104. *9 de febrer del 2012: Correlació menys un.*
Feu la següent funció:

```
cor_menys_un <- function(x,y)
```

Sabem que la correlació entre $x$ i $y$ no és tan forta com voldríem, i sabem que a les observacions contingudes en els vectors $x, y$ hi ha una observació atípica que ens fa la guitza. Aquesta funció detecta quina és aquesta observació. Penseu en totes les possibles correlacions

$$correlacio(x - x[i], y - y[i]), \forall i \in \{0, N - 1\}$$

on $x - x[i]$ i $y - y[i]$ vol dir els vectors $x, y$ sense l'observació $x[i], y[i]$. De totes aquestes, volem la més gran, és a dir, volem saber quina és l'observació $i$ tal que $correlacio(x - x[i], y - y[i])$ és la màxima. La funció torna el subíndex $i$. Podeu utilitzar la funció de la pràctica (no cal que la programeu aquí):

---

**Problema** 105. *9 de febrer del 2012: Elements no repetits.*
Feu la funció:

```
uniq <- function(v)
```

Aquesta funció fa el següent: rep un vector $v$ i torna un altre vector amb els elements del vector $v$ sense repetir. Per exemple, si tenim:

$$v = [5, 7, 6, 5, 4, 5, 6, 7, 6, 5]$$

la funció torna:

$$v = [5, 7, 6, 4]$$

L'ordre en què torna els valors no repetits no té importància.

---

**Problema** 106. *9 de febrer del 2012: Segona moda.*

Feu la funció:

```
segona_moda <- function (v)
```

Aquesta funció fa el següent: donat el vector $v$ torna el valor de $v$ que té la segona freqüència més gran. Per exemple, si:

$$v = [5, 7, 6, 5, 4, 5, 6, 7, 6, 5]$$

torna el 6, que té la segona freqüència més gran. Assumiu que tots els valors de $v$ tenen freqüències diferents i que $v$ té, almenys, dos valors diferents.

Si voleu, podeu utilitzar (encara que no l'hagueu respost) la funció **uniq** que hi ha definida al problema anterior.

---

**Problema** 107. *22 d'octubre del 2012: Sèrie.*

Sigui la suma d'una sèrie:

$$\sum_{0 \leq i \leq n} \frac{x + 2i}{i!}$$

Fes un programa que, donat un real $x$ i un enter $n$ calculi la suma d'aquesta sèrie amb $n$ termes. Òbviament, no podeu assumir que teniu una funció que calcula el factorial d'un nombre.

Si cada factorial el calculeu tot sencer a cada pas, llavors la nota serà menor.

---

**Problema** 108. *28 de gener del 2013: Triangle de Pascal.*

El triangle de Pascal serveix per a calcular coeficients binomials (per exemple, fins a $n = 4$):

$n = 0$: $\binom{0}{0}$

$n = 1$: $\binom{1}{0}$ $\binom{1}{1}$

$n = 2$: $\binom{2}{0}$ $\binom{2}{1}$ $\binom{2}{2}$

$n = 3$: $\binom{3}{0}$ $\binom{3}{1}$ $\binom{3}{2}$ $\binom{3}{3}$

$n = 4$: $\binom{4}{0}$ $\binom{4}{1}$ $\binom{4}{2}$ $\binom{4}{3}$ $\binom{4}{4}$

Els valors d'aquest Triangle de Pascal (per a $n = 4$) són:

$$\begin{array}{llllll}
n=0: & 1 & & & & \\
n=1: & 1 & 1 & & & \\
n=2: & 1 & 2 & 1 & & \\
n=3: & 1 & 3 & 3 & 1 & \\
n=4: & 1 & 4 & 6 & 4 & 1 \\
 & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
 & \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4}
\end{array}$$

Es tracta de calcular els coeficients binomials fins a un $n$ determinat. Cal tenir en compte les següents relacions per a calcular el triangle de Pascal:

$$\binom{n}{0} = \binom{n}{n} = 1$$

$$\binom{n}{k-1} + \binom{n}{k} = \binom{n+1}{k}$$

Amb aquestes dues relacions, en teniu prou per a calcular el triangle de Pascal fins a nivell $n$. El que heu de fer és un programa tal que llegeixi un $n$ del teclat, i escrigui per pantalla el triangle de nivell $n$.

Per exemple, si $n = 4$ cal que el programa escrigui per pantalla:

$$\begin{array}{lllll}
1 & & & & \\
1 & 1 & & & \\
1 & 2 & 1 & & \\
1 & 3 & 3 & 1 & \\
1 & 4 & 6 & 4 & 1
\end{array}$$

**Pista**: Penseu que per a calcular els resultats de la fila $n = i$ només cal que tingueu calculats els resultats de la fila $n = i - 1$. Això vol dir que caldrà desar els resultats de la fila $n = i$ per quan volgueu calcular els resultats de la fila $n = i + 1$.

---

**Problema** 109. *28 de gener del 2013: Línies d'una funció.*

Suposem que tenim una funció $f : \mathbb{R} \mapsto \mathbb{R}$ qualsevol. Donat un inverval $[xmin, xmax]$, sigui

$$S_N = [x_1, x_2, \ldots, x_N]$$

$N$ punts *equidistants* tals que $x_1 = xmin$ i $x_N = xmax$. Volem saber, per una banda, els punts $f(x_i)$, i després, per a cada valor $x_i$ el valor $f(x_i)+std(S_N)$ i $f(x_i) - std(S_N)$, on $std$ és la desviació estàndard.

Per tant, donat $xmin, xmax, N$, volem calcular, en una mateixa funció, el següent:

1. $S_N$.
2. $[f(x_i) \mid \forall x_i \in S_N]$.
3. $[f(x_i) + std(S_N) \mid \forall x_i \in S_N]$.
4. $[f(x_i) - std(S_N) \mid \forall x_i \in S_N]$.

Tenim que $1 \leq i \leq N$. Assumiu que les funcions $f$ i $std$ ja les teniu fetes (no cal que les programeu), definides com a:

```
f (x)
std (v)
```

Heu de fer la funció `linies`, que rep (almenys), com a paràmetres

`xmin, xmax, N`.

1. (20%) Escriu la capçalera de la funció `linies`. Recorda que, com a paràmetres, té, almenys, `xmin, xmax, N`.
2. (70%) Escriu el codi de la funció `linies` que calcula i torna les 4 coses que hem dit al principi.
3. (10%) Escriviu una crida a aquesta funció. Només la crida, no tot el `main`.

**Pista:** En aquest problema primer cal que tingueu clar *com* voleu retornar els resultats. Penseu que cal que calculeu els $N$ punts equidistants entre $xmin$ i $xmax$, que també haureu de tornar, després els $N$ punts que són els $f(x_i)$, els $f(x_i) + std(S_N)$ i els $f(x_i) - std(S_N)$. Teniu dues maneres de retornar aquesta informació (diferents vectors o vectors d'estructures). Un cop hagueu decidit quin és el mètode que voleu utilitzar, llavors escriviu la funció.

---

**Problema** 110. *Garbell d'Eratòstenes.*
Donat un enter `N`, feu una funció que retorni un vector amb tots els primers fins a `N`. Per exemple, si `N=5`, el vector ha de contenir $2, 3, 5$.

---

**Problema** 111. *Mitjana Ponderada de Notes (Parcial 8 de novembre del 2013).*
L'expedient d'un estudiant és una seqüència que es composa de (en aquest ordre): nombre total de crèdits (C), nombre (N) de parells de notes i crèdits, i després els parells amb la nota i el nombre de crèdits de l'assignatura (tants com indiqui N).

Per exemple, si un estudiant s'ha examinat de 3 assignatures, en què les notes són 8, 5 i 9, i el nombre de crèdits de cada assignatura són 3, 4.5 i 6 respectivament, la seqüència d'entrada serà:

```
13.5 3 8 3 5 4.5 9 6
```

on 13.5 és en nombre total de crèdits de l'estudiant (C = 13.5), 3 és el nombre total de parells nota-crèdit (N = 3), i 8 3, 5 4.5 i 9 6 són els 3 parells de nota-crèdit.

Feu un programa que calculi la nota mitjana ponderada de les notes de l'expedient. En l'exemple anterior, el programa escriuria per pantalla 7.444.

---

**Problema** 112. *Intercalat (Final 24 de gener del 2014).*

Feu una funció `intercalat`, que rep dos vectors $x, y$, i que retorna un vector que és la intercalació dels elements de tots dos vectors. La intercalació dels vectors $x$ i $y$ es defineix així:

$$x[1], y[1], x[2], y[2], \ldots$$

La mida de tots dos vectors d'entrada no ha de ser necessàriament la mateixa. Si un vector té $n$ elements més que l'altre, aquests $n$ elements hauran d'anar al final del vector resultat. Per exemple, si tenim:

$x = [1, 2, 3, 4]$
$y = [10, 11, 12, 13, 14, 15]$

(fixeu-vos que, en aquest cas, $n = 2$) la funció haurà de tornar:

$r = [1, 10, 2, 11, 3, 12, 4, 13, 14, 15]$

---

**Problema** 113. *Permutació (Final 24 de gener del 2014).*

Feu una funció `permutacio` que, donat dos vectors d'entrada $x, y$, amb nombres naturals positius, torni `TRUE` si i només si el vector $x$ és una permutació del vector $y$. Òbviament, si tots dos vectors no tenen la mateixa mida, un no pot ser la permutació de l'altre. Per exemple, si

$x = [1, 2, 3, 4, 3, 1]$
$y = [1, 1, 3, 2, 3, 4]$

llavors la funció torna `TRUE`. En aquest altre cas:

$x = [1, 2, 3, 4, 3, 1]$
$y = [1, 2, 3, 2, 3, 4]$

la funció tornarà `FALSE`.

---

**Problema** 114. *Seqüència més llarga (Final 24 de gener del 2014).*

Feu una funció `sequencia` que rep un vector que només conté 0's i 1's, i torna la mida del subvector més llarg amb els mateixos nombres. Un subvector d'un vector és una secció d'elements consecutius. Per exemple, si $v = [1, 2, 3, 4, 5, 6]$, un possible subvector de $v$ seria $[3, 4, 5]$, però $[2, 3, 5]$ no ho seria. Per exemple, si:

$v = [0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1]$

la funció `sequencia` torna 5, que és la mida del subvector:

$$v = [0,0,1,1,1,1,0,0,1,\underbrace{0,0,0,0,0}_{5},1,1,1]$$

---

**Problema** 115. *Rangs (Final 24 de gener del 2014).*

Feu una funció `rangs`, que rep un vector d'enters i torna un vector amb les freqüències dels enters que hi ha al vector, ordenades creixentment. Per exemple, si el vector és:

$$v = [1,3,2,1,3,2,1,3,2,4,2,2]$$

llavors la sortida serà:

$$r = [1,3,3,5]$$

perquè la freqüència del 4 és 1, la freqüència de l'1 és 3, la freqüència del 3 és 3 i la freqüència del 2 és 5.

---

**Problema** 116. *Balanceig (Final 23 de gener del 2015).*

Sigui $v$ un vector de mida $N$. Diem que el seu balanceig és la mitjana del valor absolut de la diferència entre els seus elements simètrics. Per exemple, si tenim el vector

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 5 | 1 | 5 | 2 | 3 | 2 |

el seu balanceig serà:

$$\frac{|3-2|+|5-3|+|1-2|}{3} = \frac{1+2+1}{3} = 1.33$$

Fes una funció que rebi dos vectors i que torni `TRUE` si i només si el balanceig del primer vector és estrictament menor que el del segon. Fixeu-vos que quan el nombre d'elements es senar, l'element del mig no es té en compte.

Per exemple, si tenim els vectors

$v_1 =$

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 5 | 1 | 5 | 2 | 3 | 2 |

$v_2 =$

| | | | | |
|---|---|---|---|---|
| 4 | 6 | 2 | 9 | 1 |

tornarà `TRUE` ja que el balanceig de $v_1$ és 1.33 i el de $v_2$ és 3.

En canvi, si tenim que:

$v_1 =$

| | | | | |
|---|---|---|---|---|
| 4 | 6 | 2 | 9 | 1 |

$v_1 =$

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 5 | 1 | 5 | 2 | 3 | 2 |

la funció tornarà `FALSE`.

---

**Problema** 117. *Pes Ponderat (Final 23 de gener del 2015).*

El **pes ponderat** d'un vector és la mitjana dels valors absoluts de les diferències entre un element del vector i el seu consecutiu. Formalment, ho definim com a

$$\frac{\sum_{i=1}^{N-1} |v[i] - v[i+1]|}{N-1}$$

Feu una funció `compara_pes(v)` que rebi un vector, i torni `TRUE` si i només si hi ha alguna permutació de dos elements del vector que faci **incrementar**-ne el pes ponderat. La mida del vector és sempre igual o més gran que 2. Per exemple, si el vector és

| | | |
|---|---|---|
| 1 | 2 | 3 |

la funció haurà de tornar `TRUE`, ja que el el pes ponderat de $v$ és $\frac{|1-2|+|2-3|}{2} = \frac{2}{2} = 1$, però si permutem, per exemple les posicions 2 i 3:

| | | |
|---|---|---|
| 1 | 3 | 2 |

llavors el pes ponderat és $\frac{|1-3|+|3-2|}{2} = \frac{3}{2} > 1$.

En canvi, si el vector és

| | | |
|---|---|---|
| 1 | 3 | 2 |

la funció tornarà `FALSE` ja que qualsevol permutació de dos elements donarà un pes ponderat igual o menor.

Us podria ajudar el fet de tenir les funcions:

1. `pes_ponderat(v)`. Aquesta funció retorna el pes ponderat del vector `v`.
2. `pes_ponderat_amb_permutacio(v,i,j)`. Aquesta funció retorna el pes ponderat del vector `v` amb la permutació dels valors de les posicions `i,j`. Aquesta funció pot òbviament cridar la funció `pes_ponderat(v)`.

---

**Problema** 118. *Pes Màxim (Final 23 de gener del 2015).*

Fes una funció que rebi un enter $M > 0$ i un vector $v$ de mida $N > 1$ (sempre passarà que $M < N$) i que torni la suma dels $M$ valors més grans del vector $v$. Els valors del vector $v$ poden estar repetits, i no han d'estar necessàriament ordenats.

Per exemple, si tenim:

$M = 3 \qquad v_1 =$

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 5 | 1 | 5 | 2 | 3 | 2 |

llavors tenim que la funció ha de tornar 13, ja que és la suma dels 3 valors més grans del vector: $5 + 5 + 3 = 13$.

Si $M = 5$ llavors hauria de tornar $5 + 5 + 3 + 3 + 2 = 18$.

Aquest problema es pot resoldre de diferents maneres. Per exemple, podeu tenir un vector on aneu desant els màxims del vector $v$ i anar-los eliminant del vector original. Una altra manera pot ser calcular el màxim del vector, eliminar-lo del vector (o *marcar-lo* de manera que deixi de ser un màxim) i acumular-lo en una variable que faci de sumatori. Si feu una cosa així, penseu que si el màxim és el valor 5 (posem per cas) llavors no heu d'eliminar tots els cincs del vector, sinó només un d'ells (mireu l'exemple anterior).

---

**Problema** 119. *Semiordenat (Final 7 de juliol del 2015).*

Feu una funció que, donat un vector `v`, que conté nombres naturals estrictament més grans que zero, torni `TRUE` si i només sí el vector `v` està **semiordenat**.

Un vector `v` està semiordenat si, per separat, els nombres parells del vector estan ordenats, i els nombres senars també estan ordenats, però no necessàriament tots junts. Per exemple, el vector:

$$v \quad = \quad [2, 1, 4, 3, 6, 5, 8, 7]$$

està semiordenat, perquè els senars ho estan, i els parells també. Però el vector:

$$v \quad = \quad [1, 2, 3, 5, 3, 34, 36, 38]$$

no ho està. Òbviament, si un vector està ordenat, també està semiordenat, però a l'inrevés no és necessàriament cert.

**Molt important**: només podeu fer **un sol** bucle per a aquest exercici. Dit altrament: si comproveu l'ordre dels parells en un bucle i dels senars en un altre bucle (cosa trivial), l'exercici tindrà un zero. No s'hi val fer cada bucle en dues funcions diferents. Si no apliqueu **correctament** l'esquema de cerca o recorregut, el vostre exercici tindrà una penalització del 50%

---

**Problema** 120. *Entropia (Final 7 de juliol del 2015).*

Feu una funció que, donat un vector `v`, que conté nombres naturals entre 1 i `N`, en torni l'entropia de Shannon.

L'entropia de Shannon per a un **conjunt** $\mathcal{X}$ es calcula amb la fórmula:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_b p(x). \tag{6.1}$$

En el nostre cas, substituïm la probabilitat $p(x)$ d'un element $x \in \mathcal{X}$ per la **freqüència** dins del vector `v`. D'aquesta manera, la fórmula que cal calcular és:

$$H(X) = -\sum_{x \in 1:N} freq(x, v) \log_2((freq(x, v))). \tag{6.2}$$

on $freq(x, v)$ és la freqüència del nombre `x` al vector `v`.

**Molt important**: si feu servir les funcions auxiliars `uniq`, `sort` o `freq`, la nota tindrà un **descompte** de 5 punts. Podeu fer servir la funció `max` de l'`R`.

# Capítol 7

# Algoritmes d'Estadística

---

**Problema** 121. *Tirant monedes(III).*

Donat un enter $n$ que és el número de vegades que es tira la moneda, i $p$ la probabilitat que surti cara en una tirada, feu un programa que escrigui per pantalla les probabilitats de cada possible esdeveniment i l'esperança matemàtica de la variable aleatòria: número de cares.

Heu de calcular les probabilitats d'obtenir 0, 1, 2, ..., n cares (recordeu que les probabilitats es poden obtenir aplicant la llei de probabilitats de la distribució binomial).

L'entrada consisteix en un enter $n$ i un real entre 0 i 1 $p$. Si $p$ és menor a 0 o més gran a 1, doneu el missatge: "no hi ha solucio".

La sortida ha de ser: en la primera línea i separades per espais, les probabilitats que pren la variable aleatòria (des de 0 a n cares). En la segona, l'esperança matemàtica del número de cares.

Indiqueu en el programa principal que volem una precisió de dos dígits decimals.

---

**Problema** 122. *Esperança matemàtica.*

Donat un enter $n$ que és la mida de la mostra i $prob[i]$ les probabilitats de obtenir el valor $i$ ($i = 0, 1, ..., n$), feu un programa que escrigui per pantalla l'esperança matemàtica de la sèrie de probabilitats.

En el cas que la suma de probabilitats sigui diferent a 1, retorna el següent missatge d'error "la suma de probabilitats ha de ser estrictament 1".

Indiqueu en el programa principal que es vol una precisió de dos dígits decimals.

Recordeu que :

$$E[x] = \sum_{i=0}^{n} x_i prob[x_i] \tag{7.1}$$

L'entrada consisteix en un valor enter positiu $n$ (mida mostral) i una seqüència de probabilitats (en tant per 1) de mida $n$.

El programa ha de retornar com a sortida l'esperança matemàtica de la seqüència.

---

**Problema** 123. *Model lineal simple (I).*

Feu un programa tal que, donat un enter $N$ que és la mida mostral, i una seqüència de mida $N$ de parells $Y$ (variable resposta) i $X$ (variable explicativa), calculi la $\beta_0$ (intercepció) i $\beta_1$ (pendent) de la recta de regressió que millor explica les dades.

Recordeu que:

$$\beta_1 = \frac{covar(x,y)}{var(x)} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}; \beta_0 = \overline{Y} - \beta_1\overline{X} \tag{7.2}$$

L'entrada consisteix en la mida de la seqüència $N$, i una seqüència no buida de parelles de nombres reals $X$ i $Y$.

El programa ha de retornar els valors de $\beta_0$ i la $\beta_1$.

---

**Problema** 124. *Model lineal simple (II).*

Donat $N$ mida mostral, i introduint les seqüències de mida $N$, $Y$ (variable resposta) i $X$ (variable explicativa), programeu una funció que calculi i retorni la $\beta_0$ (intercepció) i $\beta_1$ (pendent) de la recta de regressió que millor explica les dades.

La capçalera de la funció seria:

i cal comprovar, primer de tot, que la mida de $X$ i $Y$ és la mateixa (i igual a $N$).

La funció ha de retornar un vector amb dos components, $\beta_0$ i $\beta_1$.

---

**Problema** 125. *Nombres aleatoris.*

Donada una seqüència de probabilitats $p_1, \ldots, p_n$ (reals entre 0 i 1, tal que la seva suma fa 1) i un nombre $N$, feu un programa que torni $N$ nombres $x_1, \ldots, x_N$ tal que cada $x_i \in \{0, n-1\}$ i cada $x_i$ aparegui en el vector amb probabilitat $p_{x_i}$.

Per exemple, si les probabilitats són:

$$0.1, 0.2, 0.4, 0.1, 0.2$$

i $N = 10$, el programa ha de treure un vector de $N$ enters entre el 0 i el 4, tal que el 0 hi aparegui amb una probabilitat del 10%, l'1 amb una probabilitat del 20%, el 2 amb una probabilitat del 40%, el 3 amb una probabilitat del 10%, i el 4 amb una probabilitat del 20%.

Cal tenir en compte que el programa també rebrà, com a paràmetre, una llavor per a la funció *srand*. Per tant, per a generar el vector de sortida caldrà fer servir la funció *srand* (que inicialitzareu amb la llavor que es donarà com a entrada) i la funció *rand* per a generar un nombre $x \in \{0, n-1\}$ amb una probabilitat $p_{x_i}$.

L'entrada serà un enter que serà la llavor per a la funció *srand*, el nombre $N$ d'enters que caldrà treure per la sortida, la mida del vector de probabilitats i el vector de probabilitats.

La sortida serà un vector de $N$ enters en què cada enter apareix amb probabilitat $p_{x_i}$.

---

**Problema** 126. *Significació Estadística.*

Partim d'un parell de vectors $X, Y$, on tant $X$ com $Y$ són vectors i tenen la mateixa mida. Volem saber si la correlació que hi pugui haver entre les dades de tots dos vectors és significativa des d'un punt de vista estadístic. Ja sabem que, en alguns casos, dos vectors de dades poden estar fortament correlacionats i que, fins i tot, el p-value pot ser significativament baix com perquè puguem considerar la correlació com a significativa.

Per a saber si la correlació no és fruit de l'atzar, es pot utilitzar un mètode que consisteix a permutar aleatòriament les dades moltes vegades, i en cada vegada, en calculem la correlació. De cada correlació, mirem si és significativa i més gran o igual que la de les dades originals, i anem comptant, d'aquesta manera, quantes correlacions significatives més grans o iguals a l'original hi ha. Si aquest nombre és molt alt, llavors el que ens indica és que la correlació original és significativa simplement perquè independentemnt de com són les dades, ho seria igualment. És a dir, no és significativa.

L'entrada del programa és un nombre de proves $P$, una mida mostral $N$ i un vector de $N$ parells $x_i, y_i$.

---

**Problema** 127. *Variància (II).*

Hi ha dues maneres de calcular la variància d'una seqüència d'enters en què s'aplica la definició "clàssica"de variança:

$$Var(x) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{7.3}$$

on $\overline{x}$ és la mitjana dels elements de $x$ i $N$ és la mida d'$N$. Aquest algoritme necessita fer dos recorreguts sobre la seqüència d'entrada.

Hi ha una altra manera, però, de calcular-la, sense haver de fer dos recorreguts sobre la seqüència d'entrada. Aquesta segona manera s'obté desenvolupant el quadrat de la diferència.

Feu un programa que apliqui aquesta segona manera de calcular la variança.

L'entrada és un vector d'enters.

---

**Problema** 128. *Variància (I).*

Donat un vector d'enters $x$, calculeu-ne la variància segons aquesta equació:

$$Var(x) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{7.4}$$

on $\overline{x}$ és la mitjana dels elements de $x$ i $N$ és la mida d'$N$.

L'entrada és un vector d'enters.

---

**Problema** 129. *Nombres aleatoris(II).*

Feu dues funcions:

- Una funció que donat un vector de nombres reals en retorni la mitjana aritmètica.
- Una funció que donat un vector de nombres reals en retorni la variança.

Feu un programa que donat, una enter *llavor*, i un enter $n$ mida mostral, calculi la mitjana i la variància d'una mostra de nombres aleatoris. Per tal que els resultats puguin ser comparables, feu que els nombres aleatoris que entren a la mostra tinguin llavors: $llavor, llavor + 1, ..., llavor + n$. Realitzeu el mòdul a 100, i dividiu el valor resultant per 100 per tal d'obtenir valors entre 0 i 1.

Les funcions han de tenir la següents capçaleres:

L'entrada consisteix en un enter *llavor* i un enter $n$ major a 0. La sortida és la mitjana i la variància de la mostra en dos línies diferents.

---

**Problema** 130. *Nombres aleatoris(III).*

Feu tres funcions:

- Una funció que donat un vector de nombres reals et retorni la mitjana aritmètica.
- Una funció que donat un vector de nombres reals et retorni la variança.
- Una funció que donat un enter *llavor*, i 2 enters més $n$ i $m$ (mides mostrals) et retorni un vector de mida $m$ amb les mitjanes de mostres de números aleatoris de mida $n$.

Feu un programa que donat, una enter *llavor*, i les dos mides mostrals $n$ i $m$, calculi i imprimeixi per pantalla la mitjana i la variança d'una mostra de mitjanes de nombres aleatoris. Per tal que els programes siguin comparables genereu els nombres aleatoris amb llavors: $llavor$, $llavor$+1,...,$llavor$+n,...,$llavor$+2n,...$llavor$+mn. Realitzeu el mòdul a 100 a cada random, i dividiu el número resultant per 100 per tal d'obtenir valors entre 0 i 1.

Indiqueu en el programa principal que es vol una precisió de dos dígits decimals.

Les funcions han de tenir la següents capçaleres:

L'entrada consisteix en un enter *llavor* i dos enters $n$ i $m$ majors a 0. La sortida és la mitjana i la variança de la mostra de mitjanes en la mateixa línea separada per un espai.

---

**Problema** 131. *Passos aleatoris en 1D.*

Donat un nombre $N$ de passos i una probabilitat $p$ d'avançar (i per tant, de $1-p$ de recular), simuleu (amb la funció *rand*) $N$ passos amb probabilitat $p$. La probabilitat $p$ és un real entre 0 i 1

El programa ha de dir a cada pas a quina distància a devant del punt de partida (nombre positiu) o a quina distància a darrere del punt de partida (nombre negatiu) ens hem quedat. Cada distància ha d'anar separada per un espai.

Per exemple, si tenim que $N = 5$ amb una probabilitat d'avançar del 90% llavors tindrem, per exemple, que si les 5 probabilitats són sempre positives menys l'última:

$$11110$$

Això és simplement una suposició, perquè els valors dependran de la funció *rand*. En tot cas, amb aquestes probabilitats, la seqüència de sortida serà:

12343

Això vol dir que en el primer pas ens hem allunyat una unitat del centre, en el segon una altra, igual que al tercer i quart pas, ja que la funció rand ens ha donat "endevant". Això ens ha situat a distància 4 del punt d'origen. Com que la darrera probabilitat ens diu que cal recular, llavors ens trobem al punt 3.

L'entrada és una llavors (enter), un nombre de passos (enter positiu) i una probabilitat (enter entre 0.0 i 1.0). La sortida és la seqüència de distàncies, a cada pas, del punt de partida.

---

**Problema** 132. *Passos aleatoris en 2D.*

Donat un nombre $N$ de passos i una probabilitat $p_x$ d'avançar (i per tant, de $1 - p_x$ de recular) en l'eix de $y$, i una probabilitat $p_y$ d'anar a l'esquerra (i per tant, de $1-p_y$ d'anar a la dreta) en l'eix de $x$, simuleu (amb la funció $rand$) $N$ passos amb probabilitats $p_x, p_y$. Les probabilitats són reals entre 0 i 1

El programa ha de dir a cada pas a quina distància a devant del punt de partida (nombre positiu) o a quina distància a darrere del punt de partida (nombre negatiu) ens hem quedat en totes dues direccions. Imagineu-vos que teniu els eixos de les x's i les y's, i després de tirar dos daus, us diu si heu d'anar esquerra o dreta (eix x) i endevant o enrera (eix y).

Cada punt ha de tenir dues coordenades separades per coma i entre parèntesi, i els punts han d'anar separades per espais.

Per exemple, assumint que al principi ens trobem en el punt $(0, 0)$, si tenim que $N = 2$ amb una probabilitat d'avançar del 50% i d'anar a l'esquerra del 50% llavors tindrem, per exemple, que el càlcul de la primera probabilitat de bellugar-nos en tots dos eixos ens diu que hem d'avançar i anar a la dreta, llavors el primer punt serà:

$$(1, 1)$$

(Això és simplement una suposició, perquè els valors dependran de la funció $rand$). En tot cas, si el segon càlcul de les probabilitats ens diu que cal recular i anar a l'esquerra, llavors ens trobarem una altra vegada al punt d'inici:

$$(0, 0)$$

Per tant, el que hauria de treure el programa per pantalla seria:

$$(1, 1)(0, 0)$$

L'entrada és una llavor (enter), un nombre de passos (enter positiu) i dues probabilitats (reals entre 0 i 1). La sortida és la seqüència de corrdenades, a cada pas, del punt de partida.

---

**Problema** 133. *Permutació d'un vector.*

Donat un vector de reals $p$, fes un procediment que calculi una permutació del vector $p$. Cal fer-ho de la següent manera:

1. Calcularem un nombre aleatori amb la funció $rand$. Aquest nombre ha d'estar entre 0 i $N-2$, on $N$ és la mida de $p$. Sigui $i = rand()$.
2. Intercanviem l'element $p_i$ amb l'element $p_{N-1}$.
3. Procedim a fer el mateix que en el primer pas, però ara limitem més el nombre aleatori, que ha d'estar entre 0 i $N-3$, i intercanviarem l'element que doni la funció $rand()$ amb l'element $p_{N-2}$.
4. Farem això mentre hi hagi elements a intercanviar.

Cal escriure només la funció. El programa principal serà ignorat. Assumiu que en el programa principal s'ha iniciat la funció $srand$ amb paràmetre 2.

La funció tindrà com a paràmetres un vector de reals: $p$. Com a sortida, una permutació d'$p$.