

Llista 3: Variables dimensionades i cadenes de caràcters

37. Llegiu un enter n ($n \leq 15$), dos vectors $u, v \in \mathbb{R}^n$ i $\lambda \in \mathbb{R}$ i calculeu $w = u + v$ i $z = \lambda u$. Podeu fer primer el programa fixant n a 3. (**espVect.c**)

38. Donats un nombre natural n ($n \leq \text{DIM}$) i una seqüència de n notes (nombres reals no negatius), calculeu el nombre de notes per sota la nota mitjana. (**mitjanaNotes.c**)

39. Feu un programa que llegeixi un nombre natural n i escrigui els nombres primers menors o iguals que n , usant el garbell d'Eratòstenes.

El garbell d'Eratòstenes consisteix en tenir d'entrada un vector amb els naturals. Llavors s'eliminen els múltiples de 2, els de 3 i successivament els múltiples de tots els que no s'han esborrat. (**eratostenes.c**)

40. Donats un nombre natural n i dues seqüències de com a màxim n nombres enters acabades en 0, indiqueu si les dues contenen els mateixos números encara que potser en ordre diferent i amb repeticions. (**mateixos.c**)

41. Sigui $I = \{2, 3, \dots, 1000\}$. Se sap que qualsevol $n \in I$ té, com a màxim 32 divisors. Donat $n \in I$, li associem un vector de 33 components $v = (v_0, v_1, \dots, v_{32})$ de la manera següent:

- la component v_0 conté la quantitat de divisors de n ;
- les següents components v_1, v_2 , etc., contenen els divisors de n , ordenats creixentment, fins a n mateix;
- la resta de components són 0.

Per exemple, si $n = 30$ llavors $v = (8, 1, 2, 3, 5, 6, 10, 15, 30, 0, \dots, 0)$.

Diem que dos elements diferents de I són *parents* quan tenen, com a mínim, 4 divisors comuns.

Feu un programa que llegeixi dos enters, comprovi que són de I i diferents entre si, generi els dos vectors associats i digui si són, o no, parents. (**vectorsParents.c**)

42. Donat un nombre natural n ($n \leq 30$) i un vector d'enters u de dimensió n , ompliu els primers $n/2$ elements del vector amb números enters generats en ordre creixent aleatòriament. Per exemple, es pot generar la primera component del vector u mitjançant $u[0] = \text{rand}() \% 10 - 5$; i la resta de components mitjançant $u[i] = u[i-1] + \text{rand}() \% 10$;

Després s'aniran demanant per pantalla números, i cada número introduït s'afegirà al vector en el lloc que li pertoca, fent córrer una posició cap a la dreta tots els valors més grans que ell. Cada cop que s'afegeixi un número s'escriurà el contingut del vector. (**ordenats.c**)

Per exemple, si $n = 12$

```

vector = [-2, 14, 45, 56, 63, 112]
Nou numero -> 5
vector = [-2, 5, 14, 45, 56, 63, 112]
Nou numero -> 76
vector = [-2, 5, 14, 45, 56, 63, 76, 112]
Nou numero -> 13
vector = [-2, 5, 13, 14, 45, 56, 63, 76, 112]
etc ...

```

Compteu també la quantitat de vegades que moveu algun valor cap a la dreta.

43. Llegir un nombre natural n ($n \leq \text{DIM}$) i un vector v real de n components, i calcular les tres normes (**normesVectorials.c**):

$$\|v\|_1 = \sum_{i=1}^n |v_i|, \quad \|v\|_2 = \sqrt{\sum_{i=1}^n v_i^2}, \quad \|v\|_\infty = \max_{i=1,2,\dots,n} |v_i|.$$

44. Feu un programa que llegeixi un enter n i escrigui n files del triangle de Tartaglia, centrades i en forma de triangle. Per a generar les files, només es pot usar un vector. (**tartaglia.c**)

La sortida haurà de tenir el següent aspecte:

```

          1
        1 1
      1 2 1
    1 3 3 1
  1 4 6 4 1

```

45. Les targetes de crèdit contenen un codi numèric de 16 díigits decimals que escriurem

$$a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15},$$

(amb $a_0 \neq 0$), tot i que en les targetes apareixen en grups de quatre.

La darrera xifra a_{15} és el dígit de control i es calcula a partir dels altres 15 de la següent manera:

- es calcula la suma ponderada dels primers 15 díigits, amb pes 2 els que estan en les posicions parelles (a_0, a_2, \dots, a_{14}) i amb pes 1 els que estan en les posicions senars (a_1, a_3, \dots, a_{13});
- s'afegeix a la suma ponderada anterior la quantitat de díigits en posicions parelles que són més grans que 4, i s'anomena r la darrera xifra del resultat;
- el dígit de control a_{15} val 0, quan r és 0, i $10 - r$, quan no ho és.

El programa anirà llegint codis, dígit a dígit, fins que llegeixi un a_0 que valgui 0, i de cadascun informará si és o no correcte i, si no ho és, escriurà també el codi correcte. (**codabar.c**)

Exemple: si llegim 1234567890123452 9876543210987654 0987654321098765 el programa escriurà

| codi | correcte | correccio |
|------------------|----------|------------------|
| 1234567890123452 | Si | |
| 9876543210987654 | No | 9876543210987658 |

46. Feu un programa que llegeixi un valor n ($0 \leq n \leq 20$), i els coeficients d'un polinomi de grau n , $p(x) = \sum_{i=0}^n p_i x^i$, i després un valor m ($0 \leq m \leq 20$) i els coeficients d'un polinomi de grau m , $q(x) = \sum_{i=0}^m q_i x^i$, i calculi la suma $p(x) + q(x)$. Escriurà els polinomis llegits i el polinomi suma. (**sumaPolin.c**)

Per exemple, si llegim

```
3  1.2  0  -4.4  1.
3  -2.  1.  3.3  -1.
```

el programa escriurà

```
p(x)  1.2000e+00 +0.0000e+00 x -4.4000e+00 x^2 +1.0000e+00 x^3
q(x)  -2.0000e+00 +1.0000e+00 x +3.3000e+00 x^2 -1.0000e+00 x^3
suma: -8.0000e-01 +1.0000e+00 x -1.1000e+00 x^2
```

47. Volem sumar dos números enters de 25 dígit. Com que són massa llargs per guardar-los en una variable simple, farem servir vectors de 25 posicions. Feu un programa que llegeixi dos números dígit a dígit, els guardi en vectors, guardi la suma en un vector i l'escrigui. S'ha de guardar un dígit en cadascuna de les components dels vectors. (**sumaLlarga.c**)
48. Feu un programa que ompli dues matrius reals $n \times m$, $A = (a_{ij})$ i $B = (b_{ij})$, amb $a_{ij} = i(j+1)$, $b_{ij} = \frac{1}{i+j}$, $i = 1 \dots, n$, $j = 1, \dots, m$, i després les escrigui correctament encolumnades. (**omplirMatrius.c**) Podeu fer primer el programa fixant $n = 5$ i $m = 4$.
49. Donada una matriu A de dimensió $n \times n$ i un vector de dimensió n ($n \leq 15$) calculeu $v^T A v$. (**prodvTAvc.c**)
50. Donades dues matrius A i B (A de $m \times n$, B de $n \times p$), calculeu AB . (**producteMatrius.c**)
51. Llegir un nombre natural n ($n \leq \text{MIDA}$) i una matriu real A de $n \times n$ components, i calcular les dues normes usuals (**normesMatricials.c**):

$$\|A\|_1 = \max_{j=1,2,\dots,n} \sum_{i=1}^n |a_{ij}|, \quad \|A\|_\infty = \max_{i=1,2,\dots,n} \sum_{j=1}^n |a_{ij}|.$$

52. Donada una matriu quadrada A , de dimensió $n \times n$ ($n \leq \text{DIM}$), que té molts elements iguals a 0 (matriu escassa), la volem emmagatzemar utilitzant només dos vectors `elem` i `index`. Suposarem que el màxim nombre d'elements no nuls és $3n$. El vector `elem` contindrà els elements no nuls de la matriu A , i el vector `index` contindrà la posició dels elements en la matriu. La manera de guardar la posició que ocupa en la matriu és la següent: a l'element a_{ij} de la matriu A li correspon l'índex $i \cdot n + j$. Per exemple, si tenim la matriu

$$A = \begin{pmatrix} 0 & 0 & 3 \\ 1.3 & 2 & 0 \\ -4.1 & 0 & 1.8 \end{pmatrix},$$

els vectors corresponents són:

$$\text{elem} = (3, 1.3, 2, -4.1, 1.8) \quad \text{index} = (2, 3, 4, 6, 8)$$

Escriviu un programa (**matriuEscasa.c**) que ens permeti triar entre dues opcions. Mostrarà el següent menú:

- 1 donada la matriu, escriu els vectors corresponents
- 2 donats els vectors, escriu la matriu correctament encolumnada

L'usuari haurà de triar l'opció. Per l'opció 1 s'hauran de llegir la dimensió n i els elements de la matriu per files; per l'opció 2 s'hauran de llegir la dimensió n , els elements del vector `elem` i després els de `index`, ordenadament com es veu en l'exemple.

NOTES: 1) no s'ha d'usar cap matriu, 2) cal comprovar que els índexos dels diferents vectors són correctes.

53. Generació d'un **quadrat màgic** d'ordre senar n : cal omplir una matriu $n \times n$ amb els valors $1, 2, 3, \dots, n^2$, de manera que la suma de cadascuna de les files, de cadascuna de les columnes i de cada diagonal gran, doni la mateixa quantitat $(n + n^3)/2$ anomenada constant màgica. Cal pensar que el quadrat és *cíclic*, és a dir, la fila anterior a la primera fila és l'última fila, i la columna posterior a l'última columna és la primera columna.

L'algorisme per generar el quadrat és: el número 1 va en el centre de la primera fila, el 2 en la casella situada una fila abans (en aquest cas, serà l'última fila) i una columna després de la del 1, el 3 en la casella situada una fila abans i una columna després de la del 2, etc. Si, en voler posar un número per aquest sistema, la casella ja està ocupada, llavors es posa en la casella de sota de l'últim nombre posat. (**quadratMagic.c**)

Exemple:

| | | |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

Podeu modificar el programa i construir quadrats màgics amb nombres de qualsevol progressió aritmètica. Llavors la constant màgica serà $n(2a_0 + (n^2 - 1)d)/2$ on a_0 és el primer terme de la successió i d la raó.

54. Volem guardar les temperatures de cada dia durant els mesos de juny, juliol i agost de cinc poblacions (Barcelona, Manresa, Puigcerdà, Reus i Olot). Per fer això, definiu una taula de tres dimensions, on la primera indica la població, la segona el mes i la tercera el dia del mes. Llegiu per a cada població les temperatures dels mesos de juny, juliol i agost (en aquest ordre) i calculeu la temperatura màxima de totes les poblacions en el mes de juny, la menor en un cert dia de juliol, que demanareu a l'usuari, i la mitjana de les d'Olot en el mes d'agost. (**tempeEstiu.c**)
55. Donada una cadena de caràcters, `tira[100]`, i un vector `vocal[11] = "aeiouAEIOU"`, determineu quines vocals apareixen en `tira` i quants cops apareix cadascuna. (**comptarVocals.c**)
56. Concateneu dues cadenes caràcter en una tercera cadena. Podeu suposar que teniu prou espai per fer-ho. (**concatenar.c**)
57. Escriviu un programa que llegeixi un text d'un màxim de 100 caràcters i l'escrigui posant cada paraula en una línia. Supposeu que les paraules només estan separades per un espai. (**separarParaules.c**)
58. Donat un nombre natural n i una seqüència de n paraules en minúscules, indiqueu quina és la paraula més freqüent. (**paraulaFrequent.c**)

Exemple: si llegim

```
17
```

```
pa amb oli i pa amb xocolata i pa amb vi i pa amb tomaquet i pa
```

el programa escriurà

```
La paraula més freqüent és: pa
```

```
apareix 5 vegades
```