# Exercicis complementaris

## Mòdul 3: Programació

## UF1: Programació estructurada

Professorat:Joan Vancells i Flotats

Autoria: Joan Vancells

Bisbe Morgades, 15
08500 Vic (Barcelona)

# Índex

# Exercicis

1. Donada una frase acabada en punt, comptar el nombre d'aparicions de la lletra 'a'.
2. Donada una frase acabada en punt, comptar el nombre de caràcters que hi apareixen:
    a. des del principi
    b. a partir d'un caràcter donat
3. Donada una frase acabada en punt, comptar el nombre de vocals que hi apareixen.
4. Donada una frase acabada en punt, dir quina és la primera vocal que hi apareix.
5. Donada una sèrie d'enters acabada en 0, calcular la seva mitjana aritmètica.
6. Realitzar un algorisme que calculi els divisors d'un nombre donat.
7. Escriure un algorisme que determini si un nombre enter positiu és primer o no.
8. Donada una frase acabada en punt, determinar si té més lletres 'b' que 'c'.
9. Escriure un algorisme que faci la còpia d'un text acabat en punt suprimint tots els espais en blanc.
10. Donada una frase acabada en punt, comptar el nombre de paraules que comencen per 'a'.
11. Donada una sèrie d'enters acabada en 0, fer un algorisme que determini si la seqüència només està formada per valors positius.
12. Donada una sèrie d'enters acabada en 0:
    a. decidir si un valor donat pertany o no a la seqüència
    b. resoldre a) suposant que la sèrie és creixent
13. Donada una frase acabada en punt, comptar el nombre d'aparicions del grup 'la'.
14. Donada una frase acabada en punt, comptar el nombre d'aparicions dels grups 'la' i 'al'.
15. Escriure un algorisme que decideixi si una sèrie d'enters acabada en 0 és creixent.
16. Fer un algorisme que determini si una sèrie d'enters acabada en 0 és monòtona.
17. Donada una sèrie d'enters acabada en 0, comptar el nombre de monotonies creixents.
18. Escriure un algorisme que calculi el nombre de valors diferents que té una sèrie creixent de nombres enters acabada en 0.
19. Donada una sèrie d'enters acabada en 0, calcular la longitud més gran dels segments formats pel mateix valor (un segment és una subsèrie d'elements consecutius).
20. Donada una sèrie d'enters acabada en 0, trobar els valors màxim i mínim i el nombre de canvis de signe que es produeixen.
21. Donada una frase acabada en punt, comptar el nombre de paraules. Una paraula està separada de les altres per un o més blancs.
22. Donada una frase acabada en punt, fer un algorisme que determini si totes les paraules que la formen comencen per la mateixa lletra.

23. Donada una frase acabada en punt, escriure un algorisme que determini el nombre de paraules que acaben en 's'.
24. Donada una frase acabada en punt, eliminar-ne els espais en blanc inútils. Un blanc no és necessari quan va davant d'un altre o bé es troba al principi o al final de la frase.
25. Donada una frase acabada en punt, suprimir les vocals que hi hagi excepte les que sigui inici de paraula. Suposar que existeix un predicat **esvocal** que determina si un caràcter és una vocal.
26. Donada una frase acabada en punt, trobar:
    a. quantes paraules de longitud mínima hi ha
    b. quina és la longitud mitjana de les paraules
    c. quin % de paraules que comencen en 'a' tenen tres o més lletres
    d. quin % de paraules que tenen tres o més lletres comencen en 'a'

27. Donada una sèrie de tirades de ruleta (0 a 36) acabada en -1, fer un algorisme que trobi:
    a. quantes vegades ha sortit 0
    b. quantes vegades ha sortit de 13 a 24
    c. quina és la major quantitat de vegades que s'ha repetit la 2a dotzena (13 a 24)
    d. quants parells han sortit
    e. quants senars han sortit
    f. quin és el nombre anterior a cada 0
    g. comptar el major nombre de vegades seguides que han sortit alternats parells i senars
28. Escriure un algorisme que determini la part entera de l'arrel quadrada d'un nombre natural donat.
29. Donada una sèrie de zeros i uns acabada en -1 que representa un nombre en binari, calcular quin és aquest nombre en decimal.
30. Escriure un algorisme que obtingui tots els nombres perfectes compresos entre 1 i 500. Un nombre **perfecte** és aquell que és igual a la suma de tots els seus divisors excepte ell mateix.

    Per exemple, el 6 és perfecte ja que els seus divisors són 1, 2 i 3, i 1+2+3=6.
31. Escriure un algorisme que obtingui totes les parelles de nombres amics compresos entre 200 i 300. Direm que dos nombres són **amics** quan cadascun d'ells és igual a la suma de tots els divisors de l'altre excepte aquest altre.
32. Escriure un algorisme que ens escrigui tots els nombres cap-i-cua de 2, 3, 4 i 5 xifres.
33. Donada una frase acabada en punt, comptar el nombre de paraules acabades en 'cio'.
34. Realitzar un algorisme que donat un nombre enter en base 10 us el converteixi en l'equivalent en base b (1<b<10), sense necessitat d'utilitzar cap taula.
35. Donada una frase acabada en punt, comptar el nombre d'aparicions de la primera paraula.
36. Trobar el mínim índex d'un component amb un cert valor en una taula (cerca).
37. Donada una frase acabada en punt, fer una llista sense repetició de totes les paraules de longitud 3.

38. Escriure un algorisme que determini si dues sèries d'enters (que venen donades en dues taules) creixents tenen algun element en comú.

39. Donades dues sèries estrictament creixents d'enters (en taules), fer un algorisme que calculi el nombre de valors que apareixen en les dues.

40. Donades dues sèries creixents d'enters F i G (en taules), calcular la distància entre elles.

41. Definim la distància entre dues sèries com:

    min {|f-g| tal que f∈F i g∈G}

42. Donades tres sèries creixents d'enters (en taules) que tenen com a mínim un element en comú, realitzar un algorisme que calculi el valor mínim comú.

43. Donada una taula f d'enters definida sobre [1..n], fer un algorisme que escrigui els elements de f suprimint els redundants. Un element de f és redundant si és igual a un element anterior de la taula. No està permès utilitzar una taula intermitja. Feu-ho en dos casos:

    a. suposant que f no està ordenat

    b. suposant que f està ordenat

44. Direm que un nombre és **irreal** si totes les seves xifres estan en ordre ascendent. Dissenyeu una funció que comprovi si un nombre N és irreal.

45. Sigui t una taula sobre [1..n] que pren només els valors 0 i 1. Un segment de t direm que està balancejat si té el mateix nombre de zeros que de uns. Escriure un algorisme que calculi la longitud màxima dels segments balancejats de t.

46. Donada una sèrie d'enters en una taula, calcular el pes màxim dels seus segments i escriure el segment o segments amb aquest pes màxim. Un segment és una subsèrie d'elements adjacents, i el seu pes ve donat per la suma d'aquests elements.

47. Un segment d'una sèrie de n enters direm que és un **k-segment** si el producte entre tots els valors que el formen presos de dos en dos és positiu o zero. Escriu un algorisme per calcular la màxima longitud de qualsevol k-segment.

48. Donada una frase acabada en punt, codificar-la de la següent manera: a la 'a' fer-li correspondre la 'd', a la 'b' la 'e', ... i així successivament fins a la 'x' la 'a', a la 'y' la 'b' i a la 'z' la 'c'.

49. Donada una frase acabada en punt, codificar-la de la següent manera: agafar grups de 5 caràcters (inclosos els espais en blanc) i invertir-los. Tenir en compte que el darrer grup pot no tenir 5 caràcters.

50. Fer un algorisme que, donada una sèrie de caràcters de longitud arbitrària (encara que especificant un màxim), associï a cada caràcter un nombre enter positiu que representi la distància cap a l'esquerra des de la darrera aparició d'aquest caràcter. La distància màxima que considerarem serà 9 i si és més gran o no es repeteix determinat caràcter li associarem un 0.

51. Per exemple, si la seqüència d'entrada és de la forma a b m c c b f h g t h c m t f li associarem la següent taula de valors 0 0 0 0 1 4 0 0 0 0 3 7 0 4 8.

52. Donada una frase acabada en punt, comptar la freqüència de cada lletra de l'alfabet.

53. Donada una frase acabada en punt, comptar el nombre de paraules que són anagrama de la primera. Una paraula w és un **anagrama** de la paraula v si w s'obté de v canviant l'ordre de les lletres. Per exemple, PARE és anagrama de PERA.
54. Representació de polinomis. Donats dos polinomis efectuar la seva suma, el seu producte i la seva derivada.
55. Donada una xifra entera, representar-la en nombres romans.
56. Donada una xifra en nombres romans, trobar el seu corresponent valor en decimal.
57. Donada una xifra en nombres romans, examinar si és correcte o no sintàcticament.
58. Escriure un algorisme de construcció i explotació d'un diccionari català/castellà, entenent-ho com una correspondència biunívoca de paraula catalana=paraula castellana.
59. Generar totes les configuracions de 8 reines sobre un taulell d'escacs de tal manera que no es matin entre si.
60. Fer un algorisme que generi totes les combinacions possibles de loteria primitiva (combinacions de 50 elements de 6 en 6).
61. Fer un algorisme que generi totes les combinacions possibles de loteria primitiva tret d'aquelles que tinguin dos nombres consecutius o la distància entre el primer i l'últim nombre no sigui major de 10.
62. Fer un algorisme per generar totes les possibles travesses.
63. BANDERA TRICOLOR. Donada una taula de n components en que cadascun representa un de tres colors (B-blau, W-blanc, V-vermell) en una disposició qualsevol, es tracta de reorganitzar-los per tal que formin una bandera tricolor (blau, blanc, vermell). Cal treballar sobre la pròpia taula amb l'operació d'intercanvi i demanar un sol cop pel color de cada component.
64. Quan la rebel·lió de l'any 70 dels jueus contra els romans, 40 jueus foren fets presoners. Per evitar l'esclavitud, elaboraren un algorisme d'autodestrucció. Varen decidir de posar-se en cercle i enumerar-se del 1 al 40. Cada setè jueu havia de ser mort pels seus companys fins que només en quedés un. Aquest darrer s'hauria de suïcidar. El futur historiador Flavius Josephus es va situar de manera que fos el darrer supervivent i ... no es va suïcidar.
65. Aquesta història està a l'origen del problema de Josephus: n persones es posen en cercle i enumerades de 1 a n. Cada m-essim es mort i es demana la successió $J_{m,n}$ de les execucions, successió que és una permutació de (1,2,3, ..., n).

    Per exemple, per n=8 i m=3 s'obté $J_{8,3}$ = 3 6 1 5 2 8 4 7
66. Realitzar un algorisme que escrigui la permutació de Flavius Josephus per qualsevol parella (m,n).

67. Donada una frase acabada en punt, comptar el nombre de paraules que tenen les mateixes vocals que la primera.
68. Siguin t1 i t2 dues taules de nombres enters creixents de longitud diferent. Realitzar un algorisme que calculi el nombre de components i,j tals que t1[i]+t2[j]>0.

69. Escriure algorismes que determinin per una matriu de nombres enters M amb n files i m columnes, la posició dels components nuls. Considerar els dos casos següents:
    a. no coneixem res més de M
    b. M[i,j] és decreixent en funció de i i creixent en funció de j
70. Escriure un algorisme que multipliqui dues matrius.
71. Escriure un algorisme que ens doni la matriu transposada T de la matriu M de nxm.
72. Donada una matriu M de nxm d'enters, calcula:
    a. la suma de cada una de les seves files
    b. la suma de cada una de les seves columnes
73. Escriure un algorisme que donada una matriu de nxm elements, comprovi:
    a. si és simètrica ($a_{ij}=a_{ji}$ per tot i,j)
    b. si és triangular ($a_{ij}=0$ per tot i<j)
    c. si és diagonal ($a_{ij}=0$ per tot i<>j)
    d. escriure la seva matriu inversa (suposant que és de nxn)
74. Definim un **quadrat màgic** d'ordre n com una distribució en quadre dels nombres 1,2, ..., n tal que la suma dels elements de tota fila, tota columna o de les dues diagonals dóna sempre un mateix valor constant.

    Per exemple, el quadrat màgic d'ordre 5 és:

    | | | | | |
    |---|---|---|---|---|
    | 15 | 8 | 1 | 24 | 17 |
    | 16 | 14 | 7 | 5 | 23 |
    | 22 | 20 | 13 | 6 | 4 |
    | 3 | 21 | 19 | 12 | 10 |
    | 9 | 2 | 25 | 18 | 11 |

    i el principi que s'ha seguit per omplir-lo (i que es pot generalitzar a qualsevol quadrat màgic d'ordre n, essent n senar) és el següent:

    Es comença posant l'1 al mig de la primera fila i després s'anirà pujant en diagonal cap a l'esquerra. Si això provoca la sortida del quadrat per dalt o per l'esquerra, el nombre el col·locarem a la darrera fila o a la darrera columna respectivament. En canvi, si arribem a una posició ja ocupada, col·loquem el nombre just a sota del nombre anterior (fixeu-vos que això succeirà cada cop que acabem de col·locar un múltiple de n).

    Construir un algorisme que generi el quadrat màgic d'ordre n (amb n senar).
75. Quan hi ha un alt percentatge d'elements d'una matriu que tenen el valor 0, aquesta s'anomena matriu quasi-nu.la. Una forma senzilla i bastant eficaç per emmagatzemar aquest tipus de matrius sense necessitat de guardar els zeros és amb una matriu de kx3, essent k el nombre d'elements diferents de zero. A la primera i segona columnes hi guardarem la fila i la columna de cada element diferent de zero, i a la tercera columna el seu valor.

Per exemple, donada la matriu

| | | |
|---|---|---|
| 0 0 3 0 0 | | 1 3 3 |
| 4 0 0 0 0 | | 2 1 4 |
| 0 6 1 0 0 | la representarem | 3 2 6 |
| 0 0 0 3 0 | | 3 3 1 |
| 0 0 0 0 0 | | 4 4 3 |

Fer un programa que transformi una matriu quasi nul·la en la corresponent representació reduïda. Al representar una matriu quasi-nul.la en la seva forma reduïda estalviem espai però les manipulacions sobre la matriu es compliquen. Implementar la suma i la multiplicació de matrius en forma reduïda.

76. Considerant un tauler d'escacs com una matriu de 8 files per 8 columnes, feu un algorisme que ens permeti entrar dues posicions (cada posició està formada per una fila i una columna) i analitzar si la primera peça pot matar a la segona en el cas que la primera sigui:
    a. un cavall
    b. una torre
    c. un àlfil
    d. un rei
    e. una dama
77. Omplir un quadre de mots encreuats amb les seves solucions. Representarem el quadre per mitjà d'una matriu de mxn en la qual inicialment només hi ha blancs i '*' per indicar les posicions on no hi va lletra. Les paraules-solució venen donades en la cinta d'entrada separades per blancs. Haurem acabat quan el quadre estigui ple. Per simplificar, suposar que totes les paraules tenen longituds diferents.
78. Donada una matriu M que només conté 0 i 1, realitzar un algorisme que calculi la mida màxima de submatriu quadrada tal que tots els seus elements siguin 0.
79. Donades dues matrius de mxn plenes de paraules finalitzades en blanc, trobar totes les paraules comuns a ambdues.
80. Fer un algorisme que donada una matriu de NxN comprovi que la suma de cada capa és més gran que la suma de totes les seves capes inferiors. Es demana que la funció que suma una capa de la matriu sigui el màxim d'eficient possible.