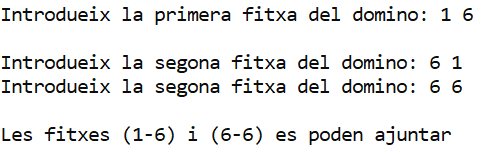
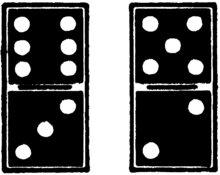
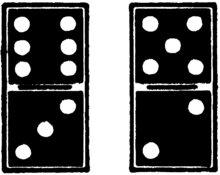
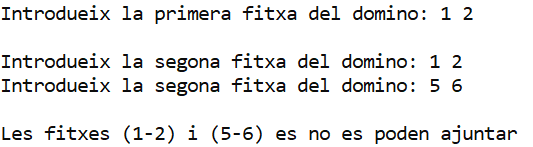
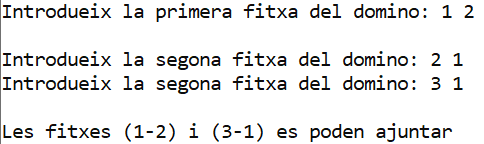
PLA DE XOC!

1. Sí.Demanar dues fitxes del dominó i determinar si es poden ajuntar o no. Una fitxa del dominó consta de dos números que estan compresos entre 1 i 6. Les dues fitxes no poden ser iguals; en cas que ho siguin es demanarà novament la segona fitxa fins que no siguin iguals. Cal resoldre l’exercici amb dues variables per a cada una de les fitxes.

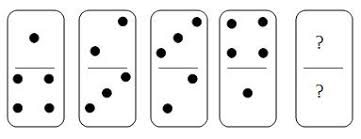




1. Sí. Resolt l’exercici anterior però desant cada una de les fitxes en un vector de dues caselles.



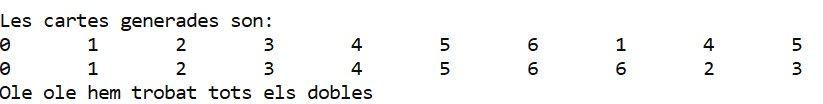
1. Sí. Defineix dos vectors de MAX fitxes. El primer vector representarà la part de dalt de la fitxa del dominó; mentre que l’altre vector representarà la part de sota de la fitxa del dominó. Un cop tinguem tot el vector ple i tenint en compte que es poden repetir les fitxes; fes una funció que ens informi si tenim totes les fitxes dobles. Les fitxes s’han de generar aleatòriament.

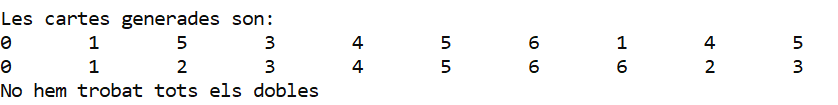
Exemple:

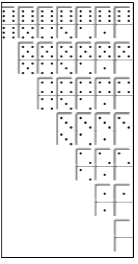
num1[MAX]={1,2,3,4}

num2[MAX]={4,3,2,1}

Recomanació: per a fer les comprovacions omple els dos vectors des de codi.

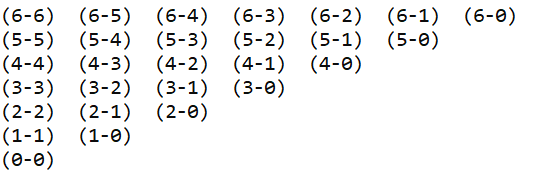


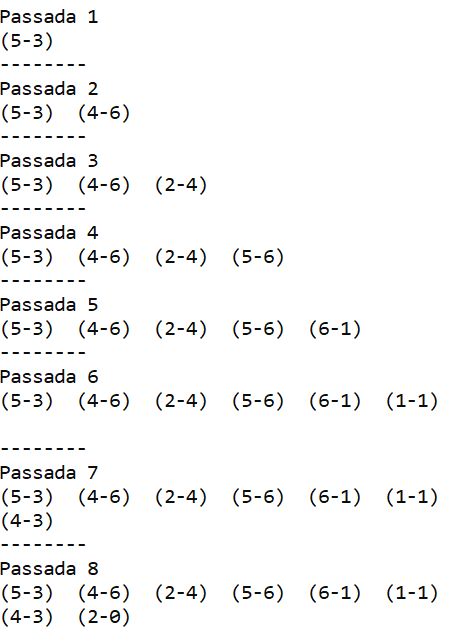


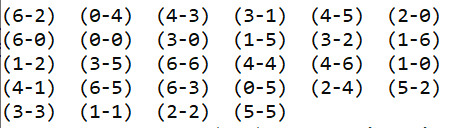
1. Sí. Genera les 28 fitxes del dominó consecutivament utilitzant dos vectors (des de la més alta fins a la més baixa). Les fitxes s’han d’introduir ordenades dins dels vectors. Un cop estiguin generades, les imprimeixes com mostra la imatge.

num1[MAX]={6,6,6,6,6,6,6,5,5,5,5,5,5,4,....}

num2[MAX]={6,5,4,3,2,1,0,5,4,3,2,1,0,4,....}

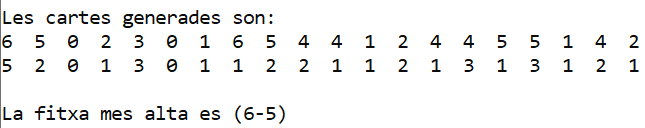


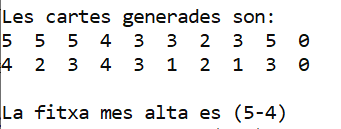
1. Sí. Utilitzant un vector de 56 caselles genera les 28 fitxes del dominó aleatòriament, tenint en compte que cada dues caselles forma una fitxa. El procediment serà:
2. Generar una fitxa utilitzant un vector de dues caselles
3. Comprovar si la fitxa ja existeix dins de les fitxes que ja tenim desades
4. En el cas que la fitxa sigui nova, l’afegirem dins del vector.
5. Un cop s’han desat totes les fitxes, mostrar-les per pantalla.



1. Sí. Genera 20 fitxes del dominó utilitzant dos vectors i mostra quina és la fitxa més alta. Per a generar les fitxes, es demana generar primer la part de dalt i posteriorment la part de sota. Per a que ens sigui més fàcil la resolució, es demana que sempre tinguem en la part superior el número més gran o igual al de sota. Per exemple la fitxa 3,4 no serà una carta vàlida; però la fitxa 4,4 sí

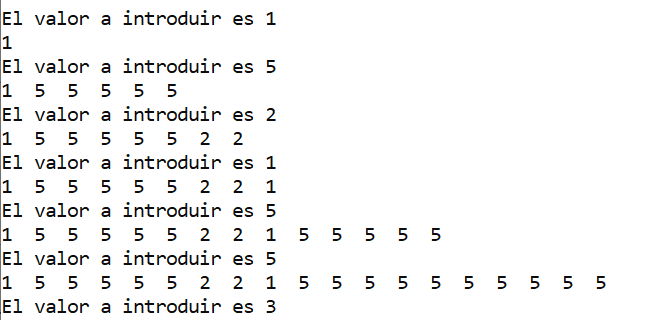
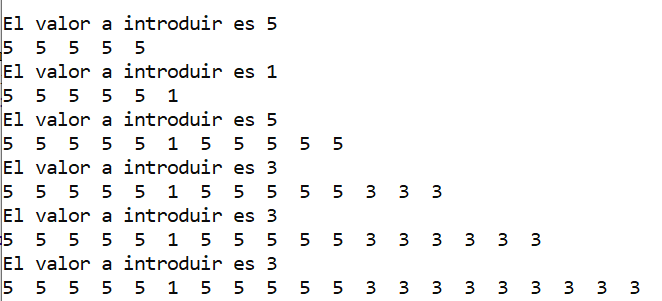
Una fitxa que contingui un 6 és més alta que qualsevol altre fitxa que no contingui el 6. En el cas que dues fitxes tinguin el número 6, serà més alta aquella que el seu segon número sigui més alt.





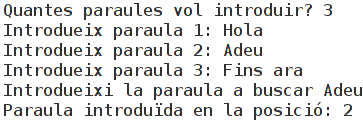
1. Sí. Simularem que estem jugant amb daus. Es demana tirar dos daus alhora i finalitzar quan en dues tirades consecutives surtin els mateixos valors dels daus. En finalitzar la jugada hem de mostrar totes les tirades efectuades. Com a màxim farem 20 tirades.



1. Jugarem amb un dau i un vector de 20 caselles. Tira el dau i omple el vector amb el número que ha sortit tantes caselles com el número ho indiqui. Si ens excedim de les caselles s’atura el programa, i els darrers números no són introduïts.
2. Genera i mostra les taules de multiplicar tot desant el seu valor dins d’una matriu

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 12 | 24 | 36 | 48 | 60 | 72 | 84 | 96 | 108 | 120 | 132 | 144 |

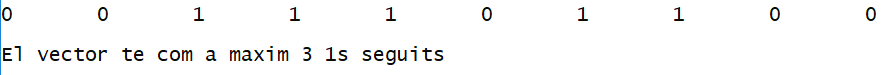
He de prémer una tecla per mostrar la següent getch()

1. Demana a l’usuari quantes paraules es volen introduir. Seguidament introdueix les n paraules i demana a l’usuari quina és la paraula que es vol cercar. En el cas que la paraula existeixi, ens haurà d’indicar en quina posició s’ha introduït.
2. Genera dos vectors plens (sense repeticions) de N caselles i indica si tots els números d’un vectors estan en l’altre.

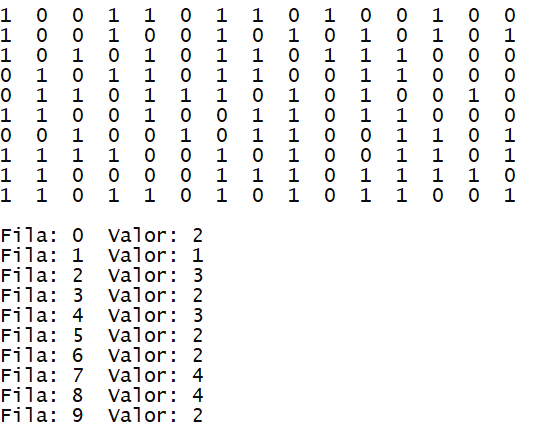
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | 6 | 8 | 10 |
| 4 | 6 | 2 | 10 | 8 |

Evidentment si un dels valors no hi és en l’altre vector el programa no haurà de continuar.

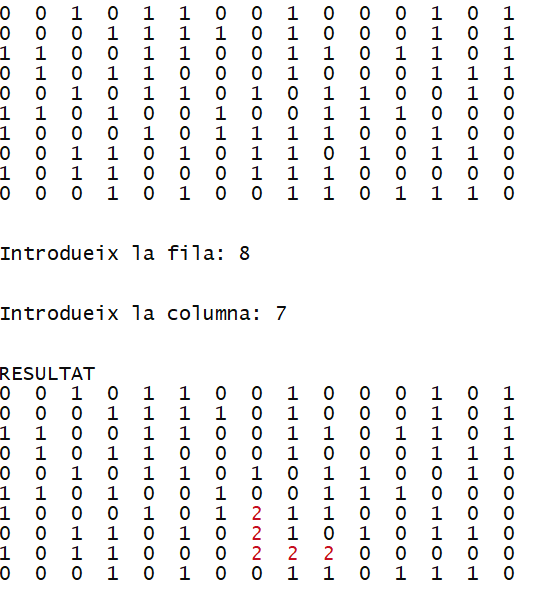
1. Donat un vector omplert amb valors 0 i 1 indica quants 1’s seguits hi ha màxims.

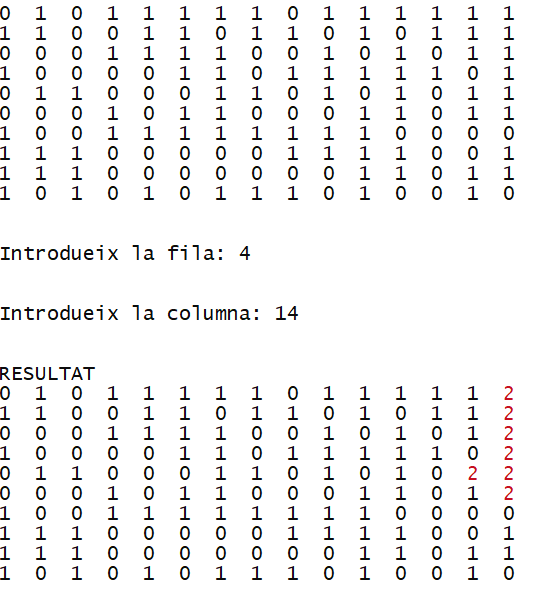


1. Donada una matriu plena amb 0 i 1 . Indica per cada una de les files quants 1’s màxims seguits hi ha.



1. Donada una matriu plena amb 1 i 0, demana fila i columna i si en aquella casella hi ha un 1, canvia tots els 1 que hi ha en la vertical i en l’horitzontal a 2. Mostra posteriorment la matriu, amb els canvis produïts.





1. Donada una matriu de caràcters, i una paraula que es demana a l’usuari comprova si aquesta paraula està dins de la matriu. Per a resoldre l’exercici es demana desar cada subcadena en una cadena auxiliar i comprovar si és igual a la paraula que es cerca.