A dark blue vertical bar runs down the left side of the page. A green arrow points from the right edge of this bar towards the text.

Dossier Exercicis M3 UF1 : Programació estructurada

Claudina Rianza Español

ÍNDIX

1	ALGORISMES SIMPLES.....	2
1.1	ESTRUCTURA SEQÜENCIAL	2
1.2	FUNCIÓ CONDICIONAL.....	2
1.3	ITERACIONS.....	10
2	FUNCIONS I PROCEDIMENTS	26
2.1	PAS DE PARÀMETRES: PER VALOR I PER REFERÈNCIA.....	32
3	VECTORS.....	39
4	EXERCICIS EXTRES DE VECTORS	45

1 ALGORISMES SIMPLES

1.1 ESTRUCTURA SEQÜENCIAL

1. Fer un programa que demani dos nombres enters a l'usuari i mostri per pantalla la seva suma.
2. Fer un programa que demani dos nombres enters a l'usuari i mostri per pantalla la seva suma, resta, multiplicació i divisió sencera
3. Dissenya un algorisme que intercanviï el valor de dues variables introduïdes pel teclat. No ha d'imprimir les dues al revés, sino que ha d'intercanviar el seu valor

Exemple si `a=10` i `b=5`, quan finalitzi el nostre programa `a=5` i `b=10`

1.2 FUNCIO CONDICIONAL

1. (**Màxim**) Fer un algoritme que calculi el màxim de dos números enters entrats per l'usuari. En cas que siguin el mateix nombre, mostrarà qualsevol dels dos nombres.

Exemple: `num1=10 num2=20`

Resultat: El màxim dels dos nombres es 20

2. (**Compara_enters**) Fer un algoritme que donats dos nombres enters, indiqui si el primer és més gran, igual o menor que el segon.

Exemple: `num1=10 num2=20`

Resultat: El primer numero es mes petit que el segon

3. (**Abs**) Fer un algoritme que mostri el valor absolut d'un número enter entrat per l'usuari. El valor absolut d'un número és el mateix número amb signe positiu

Exemple: num=-10

Resultat: El valor absolut del número és 10

Exemple: num=10

Resultat: El valor absolut del número és 10

4. **(Nat_Senar)** Fer un algoritme que donat un nombre natural, indiqui si és o no un nombre natural senar.

Exemple: num=1201

Resultat: El número 1201 és senar

5. **(Num_Vocal)** Fer un algoritme que donada una lletra vocal (majúscula o minúscula) mostri quin nombre de lletra és. Suposem valors correctes.

Per exemple, donada la 'o' (o la 'O'), l'algoritme ha de mostrar un 4.

6. **(Notes)** Fer un algoritme que donada una nota numèrica (número natural), calculi la nota textual, segons la següent taula:

<i>Nota numérica</i>	<i>Nota Textual</i>
1,2,3	Molt deficient
4	Insuficient
5	Suficient
6	Be
7,8	Notable
9,10	Excel·lent

Exemple

Nota introduïda per l'usuari: 9

Resultat: Excel·lent

Nota introduïda per l'usuari: 15

Resultat: No es una nota correcta

7. **(Dies del mes)** Fer un algoritme que donat un nombre natural de l'u al dotze, que representa el mes, indiqui si el mes corresponent té 28 (no es té en compte si és any de traspàs), 30 o 31 dies.

Per exemple: mes: 10 ; Resultat 31 dies

8. **(Estacions)** Fer un algoritme que demani a l'usuari un mes i un dia, i mostri per pantalla l'estació de l'any corresponent.

Estació	Inici	Final
Tardor	21 de setembre	20 de desembre.
Hivern	21 de desembre	20 de març.
Primavera	21 de març	20 de juny.
Estiu	21 de juny	20 de setembre.

9. **(Restaurant)** Un restaurant cobra els menús dels dies laborables a 10 € i dels festius a 14 €. Fer un algoritme que demani a l'usuari un número de l'1 al 7 que indica el dia de la setmana, i la quantitat d'euros que ha pagat el client. Llavors l'algoritme ha de donar error si el client ha pagat menys del que correspon segons el dia, o bé el canvi que se li ha de tornar, si ha pagat més.

Per exemple: numero dia=5; import=20,
mostrarà per pantalla: import a retornar 10 €

10. **(Convertidor_Temperatures)** Realitza un programa que demani la temperatura en graus Celsius i la converteixi en graus Fahrenheit (i viceversa) mostrant per pantalla un missatge del tipus:

xxx.xx graus Celsius son yyy.yy graus Fahrenheit
o bé
xxx.xx graus Fahrenheit son yyy.yy graus Celsius

La formula per a convertir graus Celsius a Fahrenheit és la següent:

$$\frac{F - 32}{9} = \frac{C}{5}$$

Per a fer la resolució de l'exercici, és demanarà a l'usuari quina de les dues conversions vol fer:

1: de graus Celsius a Fahrenheit

2: de graus Fahrenheit a Celsius

L'usuari introduirà un 1 si vol fer la conversió de graus Celsius a Fahrenheit (i per tant introduirà la quantitat de graus Celsius que vol convertir a Fahrenheit) i 2 en cas que vulgui realitzar la conversió de graus Fahrenheit a Celsius (i per tant introduirà els graus Fahrenheit que vol convertir a Celsius).

Per exemple: usuari introdueix 1 i 25 graus Celsius

Resultat: 25 graus Celsius son 77 graus Fahrenheit

Per exemple: usuari introdueix 2 i 56 graus Fahrenheit

Resultat: 56 graus Fahrenheit son 13.33 graus Celsius

11. **(Canvi_Minim)** Escriure un programa que calculi el nombre de bitllets de 500, 200, 100, 50, 20, 10 i 5, així com de monedes de 2, 1, 0,5, 0,2, 0,1 euros per a desglossar una quantitat C d'euros, de manera que es necessitin la menor quantitat de monedes i bitllets de cada tipus.

Per exemple: Import: 1647,58

Resultat:

bitllets 500: 3

bitllets 200: 0

bitllets 100: 1

bitllets 50: 0

bitllets 20: 2

bitllets 10: 0

bitllets 5: 1

monedes 2: 0

monedes 1: 1

monedes 0.50: 1

monedes 0.20: 0

monedes 0.10: 0

monedes 0.05: 1

monedes 0.02: 1

monedes 0.01: 0

12. **(CreixentDecreixent)**: Donats tres nombre a,b i c, determina si la seqüència és creixent, decreixent o cap d'elles. Es determina que una seqüència és

creixent si el valor actual és igual o superior al valor anterior. Una seqüència on els tres nombres tenen el mateix valor, la considerarem seqüència creixent

Per exemple:

si a=4, b=4 i c=11, es mostrarà el missatge "seqüència creixent"

Si a=6, b=-6 i c=11 es mostrarà el missatge "no és ni creixent ni decreixent"

Si a=6, b=6 i c=6 es mostrarà el missatge "seqüència creixent"

Si a=6, b=6 i c=-7 es mostrarà el missatge "seqüència decreixent"

13. **(Orderna_3)** Fer un algoritme que demani a l'usuari tres números naturals. El programa ha de mostrar els tres números ordenats de menor a major.

Exemple: a=4 , b=2, c=1 Resposta: 1,2,4

Abans de realitzar l'exercici omple la taula que s'adjunta

a	b	c	max	mig	min
10	10	10			
10	10	8			
10	6	10			
10	7	6			
10	6	7			
10	7	7			
8	10	7			
7	10	8			
7	10	7			
4	10	10			
5	6	10			
6	5	10			
5	5	10			

14. **(Any_Traspas)** Escriu un programa que demani un any i digui si és de traspàs o no.

Seran de traspàs quan...

- l'any sigui divisible per 4, excepte si és un final de segle (acaba en 00).

- l'any sigui un final de segle i les dos primeres xifres de l'any també siguin divisibles per 4.

Abans de resoldre analitza la taula i omple els anys que falten

Any	Divisible per 4	És final de segle?	Dues primeres xifres	Les dues primeres xifres són divisibles entre 4?	ÉS DE TRASPAS
1600	SI	SI	16	SI	SI
1601	NO				NO
1604	SI	NO			SI
1608	SI	NO			SI
1700	SI	SI	17	NO	NO
2000					
2004					
2100					
2400					

15. (**Calcul_Salari**) Calcula el salari net dels empleats d'una empresa pagats per hores de la següent manera (if anidats):

- les hores inferiors a 35 hores es paguen a una tarifa determinada que s'ha d'introduir pel teclat al igual que el número d'hores
- Les hores iguals o superiors a 35 es pagaran com extres a un preu de 1.5 hores normals.
- L'import a deduir als treballadors depèn del sou:
 - sou ≤ 120 €, no impostos
 - les següents 900 € al 20%
 - La resta al 30%.

Nota: Les hores treballades per l'empleat seran sempre hores senceres, mentre que el preu per hora contindrà decimals

Per exemple: hores=11; preuhora=12, mostrarà per pantalla 105,6

Per exemple: hores=38; preuhora=20, mostrarà per pantalla 640

Per exemple: hores=40; preuhora=25, mostrarà per pantalla 752,50

Abans de resoldre l'exercici omple la taula:

N Hores	Preu Hora	Hores A Preu Normal	nHores Extres	Import Normal	Import Hores Extres	Import Total	Deduc cions	Import a Cobrar
11	12							
38	20							
40	25							

16. (**Benzinera**) Fes un programa que mostri quan haurà que pagar un client en una benzinera atenent a les següent condicions:

1. Si el client te la `targeta client`, tindrà un descompte del 10% sobre el preu per litre.
2. La benzina té un cost de 1.30€/litre.
3. El gasoil té un cost de 1.50€/litre.

El programa haurà de demanar tipus de combustible; quantitat de litres i si el client disposa de la `targeta client`

17. (**Hotel**) Crea un algoritme que calculi el preu a pagar en un hotel; cal tenir en compte que durant els dies laborals el preu de l'habitació és de 80 € mentre que el cap de setmana és de 100. Se li aplicarà un descompte del 5% en cas que sigui soci Premium (aquesta variable serà tipus bool). El client només estarà en l'hotel 1 dia.

18. (**Públic cinema**) Un cinema ha demanat si podem fer un programa que calculi l'import que ha de pagar un client. Aquest cinema compta amb unes ofertes segons l'edat dels usuaris, juntament amb una diferència de preu segons si són dies laborables o cap de setmana. Suposem que només entra al cinema una única persona.

Ofertes disponibles

- Oferta Infantil (de 0 a 15 anys) --> 70%
- Oferta Junior (de 16 a 59 anys) --> 30%
- Oferta Senior (més de 60 anys) --> 15%

Preus cap de setmana: 12 €

Preus dies laborables: 8€

19. (**Públic cinema2**) Un altre cinema ens ha demanat un altre programa. Aquest cinema vol calcular quin serà l'import total d'un grup de persones. Per a desenvolupar el programa es demanarà quantes persones volen entrar al cinema de menys de 10 anys i quantes persones majors o iguals a 10 anys. Seguidament cal tenir present que el preu per als adults és de 10.50, i de cada menor és de 8.50. El preu es manté igual per a tots els dies de la setmana; però dimarts és el dia de l'espectador i el seu preu final és redueix en un 30%.

Ull per què s'aplica un 20% de descompte adicional sobre el preu final; en cas del pack familiar. El pack familiar té com a mínim 2 adults i 2 menors.

1.3 ITERACIONS

1. **(Numeros1al100)** Escriure els números de 1 a 100. Fer l'exercici amb les tres estructures MENTRE, PER i REPETIR
2. **(Numeros1alNumero)** Escriure els números des de l'1 fins al número introduït per l'usuari. En cas de negatiu Per exemple: num=-3 ha d'escriure 1,0,-1,-2,-3

Introdueix el num:	Introdueix el num:
> 5	> -4
1	1
2	0
3	-1
4	-2
5	-3
	-4

3. **(Taula del 5)** Fer un algoritme que mostri per pantalla la taula del 5.

```
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

4. **(Taula de multiplicar)** Fer un algoritme que mostri per pantalla la taula del número que indiqui l'usuari. El resultat que mostri el vostre programa ha d'ésser igual al que s'adjunta com a la imatge.

Introdueix quina taula vols:

```
> 7
7 * 0 = 0
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
```

5. **(Parells)** Fer un algoritme que mostri tots els nombres parells del 0 al número natural que indiqui l'usuari. Fer l'exercici amb les tres estructures MENTRE, PER i REPETIR

Introdueix el darrer numero:

```
> 7
0
2
4
6
```

6. **(><)** Fer un algoritme que demani a l'usuari dos nombres naturals i un caràcter, que pot ser '>' o '<'. Llavors, l'algoritme ha de mostrar per pantalla tots els nombres naturals entre els dos introduïts, ordenats de menor a major si el caràcter és '>' o de major a menor si el caràcter és '<'

Per exemple:

```
Caràcter: '>'
Primer número: 10      Segon Número: 2
Resultat: 2 3 4 5 6 7 8 9 10
Caràcter: '>'
Primer número: 2       Segon Número: 10
Resultat: 2 3 4 5 6 7 8 9 10
Caràcter: '<'
Primer número: 10      Segon Número: 2
Resultat: 10 9 8 7 6 5 4 3 2
```

Caràcter: '<'

Primer número: 2 Segon Número: 10

Resultat: 10 9 8 7 6 5 4 3 2

Introdueix el signe:

> >

Introdueix el primer numero:

> 7

Introdueix el segon numero:

> 3

3

4

5

6

7

Introdueix el signe:

> <

Introdueix el primer numero:

> 10

Introdueix el segon numero:

> 13

13

12

11

10

7. **(Suma n)** Fer un algoritme que demani a l'usuari un nombre natural. L'algoritme ha de mostrar la suma de tots els nombres entre el 1 i el número introduït per l'usuari. Cal controlar que el valor introduït per l'usuari sigui positiu

Procediment:

acum = 0

i=1 acum = 0 + 1 = 1;

i=2 acum = 1 + 2 = 3;

i=3 acum = 3 + 3 = 6;

i=4 acum = 6 + 4 = 10;

i=5 acum = 10 + 5 = 15;

i=6 acum = 15 + 6 = 21;

Introduir n:

> -3

Introduir n:

> -6

Introduir n:

> 4

La suma total es: 10

8. **(Comptador-Suma parell-senar)** Fer un algoritme que demani 10 números enters a l'usuari. El programa ha de mostrar per pantalla la quantitat i la suma

de tots els números parells i la quantitat i la suma de tots els números senars introduïts per l'usuari

```
Introdueix el numero:
> 2
> 4
> 2
> 4
> 2
> 4
> 1
> 1
> 1
> 3
Total de numeros parells: 6
El total de la seva suma es: 18
Total de numeros senars: 4
El total de la seva suma es: 3
```

9. **(Suma2al100)** : Sumar els números parells compresos entre el 2 i el 100.

```
El suma total del numeros parells es : 2550
```

10. **(MitjaAritmetica)**: Calcula la mitjana aritmètica de n valors introduïts per l'usuari. Finalitzarem d'introduir nombres quan l'usuari introdueixi un -1. Considerarem que tots els números introduïts són enters.

Exemple:

Nombres introduïts per l'usuari: 1 5 7 4 2 4 5 8 4 6 -1

Resultat: Mitja aritmetica del (10) numeros itnroduits: 4.6

```
Introdueix num:
> 2
Introdueix num:
> 5
Introdueix num:
> 7
Introdueix num:
> -1
El promig es: 4.6666666667
```

11. **(MesGranMesPetit):** Calcula el número más gran i més petit dels números introduïts per teclats. Acabem d'introduir números quan l'usuari ens introdueixi el 0. EL 0 no s'ha de tenir en compte, és una marca final.

Exemple:

Nombres introduïts per l'usuari: 1 5 7 4 2 4 5 8 4 6 0

Resultat: Maxim: 8; Minim: 1

Nombres introduïts per l'usuari: 0

Resultat: No s'ha introduït cap valor correcta

Introdueix numero:

> 4

> 5

> 8

> 2

> 0

min: 2 max: 8

Introdueix numero:

> 4

> 5

> 8

> 2

> 0

min: 2 max: 8

12. **(Factorial):** Calcula el factorial del número natural introduït per teclat.

Exemple:

Número: 6

Resultat: 720

Aclariment:

Res=1

Res=1*1=1

Res=1*2=2

Res=2*3=6

Res=6*4=24

Res=24*5=120

Res=120*6=720

13. **(Serie):** Calcular les 10 primeres potències de 4.

0	1
1	4
2	16
3	64
4	256

5	1024
6	4096
7	16384
8	65536
9	262144

14. **(Lletres)** Fer un algoritme que conti quantes lletres vàlides ha introduït l'usuari fins que introdueixi una 'z' o 'Z'.

```

Escriu una lletra
> S
> x
> p
> A
> .
>
> !
> z
Shan introduït 4 caràcters (sense comptar la darrera z)

```

15. **(Comptar 'a' s)** Fer un algoritme que donat una sèrie de caràcters introduïts per l'usuari, calculi el nombre de lletres 'a' introduïdes. L'usuari introduirà 10 caràcters.

```

Escriu una lletra
> e
> e
> e
> w
> w
> w
> w
> w
> a
> a
Shan introduït 2 as

```

16. **(Comptar 'a's limitat)** Fer un algoritme que demani un número natural entre 5 i 10 a l'usuari. L'algoritme ha de demanar després tantes lletres com indiqui el número demanat anteriorment i en finalitzar, ha de mostrar per pantalla quantes lletres 'a' ha introduït l'usuari.

Exemple:

Numero: 6 //cal controlar que el numero estigui entre 5 i 10

Usuari introdueix: edtsfa

Resposta: L'usuari ha introduït 1 cop el caracter 'a'

```
Quantes lletres vol introduir?
> 20
Quantes lletres vol introduir?
> 3
Quantes lletres vol introduir?
> 5
Escriu una lletra
> r
> t
> c
> a
> e
Shan introduït 1 as
```

17. Fes un programa que ens calculi quants dígits té un número positiu.

```
Introduïx el numero
> -50
Introduïx el numero
> -30
Introduïx el numero
> 3040
El numero te: 4 digits
```

18. **(Divisor)** Fer un programa que demani números naturals a l'usuari, fins que introdueixi un 0, i ens indiqui per a cadascun dels números introduïts, si el número que hem introduït prèviament és divisor del número que acabem d'introduir.

```
introdueix numero:
> 6
introdueix numero:
> 3
Si, ja que 6 es divisible entre 3
introdueix numero:
> 8
No, ja que 3 no es divisible entre 8
introdueix numero:
> 4
Si, ja que 8 es divisible entre 4
introdueix numero:
> 0
```

```
introdueix numero:
> 0
No has escrit cap valor valid
```

```
introdueix numero:
> 3
introdueix numero:
> 0
Has introduït un 0 en el segon numero
```

19. **(Fibonacci)** Fer un algoritme que calculi el terme n de la sèrie de Fibonacci. Aquesta sèrie està composta de la següent manera:

```
f0 = 0
f1 = 1
fn = fn-1 + fn-2;
Això produeix els següents valors:
f2 = 1
f3 = 2
f4 = 3
f5 = 5
f6 = 8
f7 = 13
f8 = 21
I així succesivament
```

```
introdueix quin terme vols de la serie Fibonacci:
> 6
fn2: 0 fn1: 1 fn: 1
fn2: 1 fn1: 1 fn: 2
fn2: 1 fn1: 2 fn: 3
fn2: 2 fn1: 3 fn: 5
fn2: 3 fn1: 5 fn: 8
Et terme 6 es 8
```

20. **(Comptar "la"s)** Fer un algoritme que donat una sèrie de caràcters introduïts per l'usuari, calculi en nombre de "la" introduïdes. Finalitzarem d'introduir caràcters quan l'usuari posi un '.'

Exemple:.

Usuari introdueix: LArrLalartla.

Resposta: L'usuari ha introduït 4 cops la sequencia 'la'

```
Introdueix caracter:  
> p  
Introdueix caracter:  
> l  
Introdueix caracter:  
> a  
Introdueix caracter:  
> p  
Introdueix caracter:  
> l  
Introdueix caracter:  
> a  
Introdueix caracter:  
> .  
Has introduit 2 vegades la
```

21. **(CompleixenCondicio)**: Dissenyar un algorisme que sumi tots els naturals menors de 1000 que compleixin la següent condició:

La suma de les unitats més les centenes és la meitat de les desenes. Exemple: 183 compleix la condició. Indica a més a més de la suma total, quins nombres compleixen aquesta condició.

Recordeu que per obtenir el quocient amb el programa PSEINT cal aplicar la funció trunc (que retorna la part sencera de la divisió)

```
*** Ejecución Iniciada. ***  
0  
21  
42  
63  
84  
120  
141  
162  
183  
240  
261  
282  
360  
381  
480  
La suma total es: 2820
```

22. **(N_vocals)** Fer un algoritme que demani un cert número natural (n) a l'usuari. Després ha de demanar una sèrie de lletres acabades en '.' i mostrar per pantalla si hi ha en aquesta sèrie, al menys, 'n' lletres vocals.

Fixeu-vos que quan s'obtenen les 4 vocals que es demanen ja no cal continuar demanant més caràcters

```

Quantes vocals vols trobar ?
> 4
Introdueix caracter:
> a
Introdueix caracter:
> e
Introdueix caracter:
> y
Introdueix caracter:
> i
Introdueix caracter:
> a
Hem trobat les n vocals

```

```

Quantes vocals vols trobar ?
> 4
Introdueix caracter:
> a
Introdueix caracter:
> f
Introdueix caracter:
> g
Introdueix caracter:
> t
Introdueix caracter:
> y
Introdueix caracter:
> e
Introdueix caracter:
> i
Introdueix caracter:
> .
No hem trobat les n vocals

```

23. **(Totes_les_vocals)** Fer un algoritme que demani a l'usuari una sèrie de lletres acabades en '.'. Llavors l'algoritme ha de mostrar si en aquesta sèrie de lletres hi ha totes les vocals, almenys un cop. Fixeu-vos que si troba les 5 vocals, no ha de continuar el programa.

```

*** Ejecución Iniciada. ***
Introdueix caracter:
> a
Introdueix caracter:
> e
Introdueix caracter:
> a
Introdueix caracter:
> p
Introdueix caracter:
> i
Introdueix caracter:
> o
Introdueix caracter:
> u
Hem trobat les n vocals

```

```

*** Ejecución Iniciada. ***
Introdueix caracter:
> p
Introdueix caracter:
> a
Introdueix caracter:
> a
Introdueix caracter:
> .
No hem trobat les vocals

```

24. **(QuantesVocal_QuantesConsonants)** Fer un algoritme que demani lletres a l'usuari fins que l'usuari introdueixi una 'z', amb un màxim de 10 caràcters. Llavors ha de mostrar per pantalla quantes vocals i quantes consonants ha introduït

Exemple

Caracters introduïts per l'usuari: aaa,,,tt4f

Resultat: Total de vocals:3 I Total de consonants: 3

Exemple

Caracters introduïts per l'usuari: afgz

Resultat: Total de vocals:1 i Total de consonants: 2

25. **(Compta_Paraules)** Donada una frase acabada en '.'. Calcula quantes paraules s'han introduït. Cal tenir en compte que els caràcters que introduïrem seran tots correctes i que entre paraula i paraula hi poden haver més d'un espai. Entre paraula i paraula només hi haurà espais.

```

Caracter:
>
>
> h
> o
> l
> a
>
>
> n
> o
> i
>
>
>
> .
Hem escrit 2 paraules

```

26. **(Serie_Creixent_Res_Decreixent)** Fer un algoritme que donat una sèrie de nombres naturals introduïts per l'usuari, indiqui si aquesta sèrie és creixent (cada nombre és major o igual que l'anterior), no creixent (hi ha un nombre que no és igual o major que l'anterior) o decreixent (tots els nombres són

iguals o més petits que l'anterior). La sèrie de nombres acaba quan l'usuari introdueix un 0 (zero)

Exemples:

Números introduïts: 3 3 6 7 8 8 0

Resposta: Sèrie creixent

Números introduïts: 3 3 6 7 6 8 0

Resposta: Sèrie no creixent

Números introduïts: 3 3 2 2 1 0

Resposta: Sèrie decreixent

Números introduïts: 3 3 3 0

Resposta: La sèrie no creix ni decreix

Procediment: demana els dos primers números abans d'entrar en el bucle i determina si la seqüència amb aquests dos números està pujant o baixant.

El cas que siguin iguals l'hauràs de tractar com a cas especial

Seguidament dins del bucle cerca quines són les condicions que fan que tant creixent com decreixent canviïn d'estat

```
Introdueix numero:
> 3
Introdueix numero:
> 3
Introdueix numero:
> 6
Introdueix numero:
> 7
Introdueix numero:
> 8
Introdueix numero:
> 9
Introdueix numero:
> 0
creix
```

```
Introdueix numero:
> 3
Introdueix numero:
> 3
Introdueix numero:
> 6
Introdueix numero:
> 7
Introdueix numero:
> 6
Introdueix numero:
> 8
Introdueix numero:
> 0
Ni creix ni decreix
```

```
Introdueix numero:
> 3
Introdueix numero:
> 2
Introdueix numero:
> 2
Introdueix numero:
> 1
Introdueix numero:
> 0
decreix
```

```
Introdueix numero:
> 3
Introdueix numero:
> 3
Introdueix numero:
> 3
Introdueix numero:
> 0
La sèrie no creix ni decreix
```

27. (**Divisible_NoDivisible**) Fer un algoritme que demani un nombre a l'usuari i mostri per pantalla el missatge "DIVISIBLE" si es pot dividir entre algun nombre del 2 al 10. Si no, ha de mostrar el missatge "NO DIVISIBLE". Fixa't que en el moment que sigui divisible per un número ja no cal que continuem fent la resta de divisions.

Exemples:

Número: 27

Resultat: Divisible

Número: 17

Resultat: No Divisible

28. (**N_Suma_de_Numeros**) Fer un algoritme que demani un número natural (n) a l'usuari. Aquest algoritme ha d'indicar si la suma d'una sèrie de com a molt cinc números demanats també a l'usuari, és més gran que el primer número natural (n) demanat.

```
Quin és el numero maxim?
> 10
Introdueix numero:
> 1
Introdueix numero:
> 1
Introdueix numero:
> 1
Introdueix numero:
> 1
Introdueix numero:
> 1
La suma no sha trobat abans de demanar els n numeros
```

```
Quin és el numero maxim?
> 10
Introdueix numero:
> 2
Introdueix numero:
> 200
La suma sha trobat abans de demanar els n numeros
```

29. (**Mastermind**) Fer un algoritme que generi un nombre aleatori de 4 xifres. L'usuari llavors ha d'introduir nombres de 4 xifres fins que l'encerti o be hagi fet n intents. El nombre d'intents també els ha d'introduir l'usuari. A cada nombre de 4 xifres que introdueixi l'usuari, el programa li ha de dir si el nombre que ha d'endevinar és més gran o més petit. Si l'usuari l'encerta en un nombre de vegades menor que n, llavors ha guanyat. Si no, ha perdut.

Per a crear números aleatoris en PESINT

```

Algoritmo Aleatoris
Definir aleatori Como Entero;

    aleatori=Azar(10); //numero entre 0 i 9
    Escribir aleatori;

    aleatori=Aleatorio(10,14); // numero entre 10 i 14
    Escribir aleatori;
FinAlgoritmo

```

```

El numero a cercar es: 2748
Quants intents vols fer?
> 5
Introdueix un numero:
> 2000
Ha desser mes gran
Introdueix un numero:
> 3000
Ha desser mes petit
Introdueix un numero:
> 2500
Ha desser mes gran
Introdueix un numero:
> 2800
Ha desser mes petit
Introdueix un numero:
> 2748
Moltes felicitats

```

```

El numero a cercar es: 5191
Quants intents vols fer?
> 5
Introdueix un numero:
> 5000
Ha desser mes gran
Introdueix un numero:
> 5191
Moltes felicitats

```

```

El numero a cercar es: 3518
Quants intents vols fer?
> 2
Introdueix un numero:
> 3000
Ha desser mes gran
Introdueix un numero:
> 5000
Ha desser mes petit
Un altre cop sera

```

Per a crear números aleatoris en C utilitzarem el següent codi

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i;
    srand(time(NULL)); //SEMPRE COM A PRIMERA SENTÈNCIA DEL MAIN
    for (i=0;i<50;i++)
        printf("%d ",rand()%50); //donarà 50 numeros entre 0 i
49
}

```


30. **(Hotel2)** Crea un algoritme que calculi el preu a pagar en un hotel; cal tenir en compte que durant els dies laborals el preu de l'habitació és de 80 € mentre que el cap de setmana és de 100. Se li aplicarà un descompte del 5% en cas que sigui soci Premium (aquesta variable serà tipus bool). El client només estarà en l'hotel durant un període de N dies, que haurà també d'introduir com a paràmetre

```
Quants dies vol estar a l'hotel?  
> 5  
En quin dia de la setmana entres a l'hotel?  
> 6  
Total a pagar: 440
```

31. **(Taules de multiplicar)** Fer un algoritme que mostri per pantalla les taules de multiplicar de l'u al deu, fent una pausa a cada taula.

```
TAULA 1  
1 X 1 = 1  
1 X 2 = 2  
1 X 3 = 3  
1 X 4 = 4  
1 X 5 = 5  
1 X 6 = 6  
1 X 7 = 7  
1 X 8 = 8  
1 X 9 = 9  
1 X 10 = 10
```

```
TAULA 2  
2 X 1 = 2  
2 X 2 = 4  
2 X 3 = 6  
2 X 4 = 8  
2 X 5 = 10  
2 X 6 = 12  
2 X 7 = 14  
2 X 8 = 16  
2 X 9 = 18  
2 X 10 = 20
```

32. **(Taules de multiplicar Columnes)** Fer un algoritme que mostri per pantalla les taules de multiplicar de l'u al deu, fent una pausa a cada taula.

1 X 1 = 1	2 X 1 = 2	3 X 1 = 3	4 X 1 = 4	5 X 1 = 5
1 X 2 = 2	2 X 2 = 4	3 X 2 = 6	4 X 2 = 8	5 X 2 = 10
1 X 3 = 3	2 X 3 = 6	3 X 3 = 9	4 X 3 = 12	5 X 3 = 15
1 X 4 = 4	2 X 4 = 8	3 X 4 = 12	4 X 4 = 16	5 X 4 = 20
1 X 5 = 5	2 X 5 = 10	3 X 5 = 15	4 X 5 = 20	5 X 5 = 25
1 X 6 = 6	2 X 6 = 12	3 X 6 = 18	4 X 6 = 24	5 X 6 = 30
1 X 7 = 7	2 X 7 = 14	3 X 7 = 21	4 X 7 = 28	5 X 7 = 35
1 X 8 = 8	2 X 8 = 16	3 X 8 = 24	4 X 8 = 32	5 X 8 = 40
1 X 9 = 9	2 X 9 = 18	3 X 9 = 27	4 X 9 = 36	5 X 9 = 45
1 X 10 = 10	2 X 10 = 20	3 X 10 = 30	4 X 10 = 40	5 X 10 = 50

6 X 1 = 6	7 X 1 = 7	8 X 1 = 8	9 X 1 = 9	10 X 1 = 10
6 X 2 = 12	7 X 2 = 14	8 X 2 = 16	9 X 2 = 18	10 X 2 = 20
6 X 3 = 18	7 X 3 = 21	8 X 3 = 24	9 X 3 = 27	10 X 3 = 30
6 X 4 = 24	7 X 4 = 28	8 X 4 = 32	9 X 4 = 36	10 X 4 = 40
6 X 5 = 30	7 X 5 = 35	8 X 5 = 40	9 X 5 = 45	10 X 5 = 50
6 X 6 = 36	7 X 6 = 42	8 X 6 = 48	9 X 6 = 54	10 X 6 = 60
6 X 7 = 42	7 X 7 = 49	8 X 7 = 56	9 X 7 = 63	10 X 7 = 70
6 X 8 = 48	7 X 8 = 56	8 X 8 = 64	9 X 8 = 72	10 X 8 = 80
6 X 9 = 54	7 X 9 = 63	8 X 9 = 72	9 X 9 = 81	10 X 9 = 90
6 X 10 = 60	7 X 10 = 70	8 X 10 = 80	9 X 10 = 90	10 X 10 = 100

Cal utilitzar la funció gotoxy, a través del següent codi:

```
#include <stdio.h>
#include <stdlib.h>
#include "rlutil.h" //cal afegir la llibreria al projecte

int main (void)
{
    int x, y;

    x=10; y=20;
    gotoxy(x,y);

    printf("x");
}
```

33. **(Rectangle)** Fer un algoritme que demani a l'usuari dos naturals entre 2 i 20. L'algoritme ha de dibuixar per pantalla un quadrat que tingui d'ample el primer dels dos naturals introduïts i d'alt el segon. Per exemple:

```
Introdueix l'alçada del rectangle: 4
Introdueix l'amplada del rectangle: 10
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

34. (**Rectangle_Buit**) Fer una modificació de l'algoritme anterior, que dibuixi un rectangle buit.

Exemple amplada=5 i alçada=4

	col=1	col=2	col=3	col=4	col=5
fila=1	*	*	*	*	*
fila=2	*				*
fila=3	*				*
fila=4	*	*	*	*	*

```
Introdueix l'alçada del rectangle: 5
Introdueix l'amplada del rectangle: 10
XXXXXXXXXX
X        X
X        X
X        X
XXXXXXXXXX
```

2 FUNCIONS I PROCEDIMENTS

1. **(Maxim)** Fer una funció que passant-li dos nombres, ens retorni el màxim dels dos nombres. En el cas que els dos nombres siguin iguals, la funció ha de retornar qualsevol dels dos nombres

```
int maxim(int num1, int num2);  
Exemple: num1=1, num2=5  
Resultat: 5
```

2. **(NotaNumericaaLletres)** Fer un procediment que passant-li una nota numèrica mostri la seva qualificació en lletres

```
void imprimirNota(int nota);  
Exemple: nota=2  
Resultat: Insuficient
```

3. **(NotaNumericaaLletres2)** Fer un programa que, a partir d'una funció demani a l'usuari una nota compresa entre 0 i 10. Posteriorment aquesta nota, i a través del procediment que s'ha creat a l'exercici número 2, cal transformar-la en lletres.

```
Funció: int demanarNota() // cal controlar que el número sigui  
correcte  
Procediment: void notaEnLletres(int nota)
```

4. **(NotaNumericaaLletres5)** Realitzar l'exercici número 3 (demaneu nota i visualitzeu nota en lletres), però amb 5 alumnes. Ha de mostrar:

```
Alumne 1 ha tret Insuficient  
Alumne 2 ha tret Excel·lent  
...  
Alumne 5 ha tret Notable
```

5. **(TaulaMultiplicar)** Fer un procediment que passant-li un número, ens mostri la taula de multiplicar d'aquell número. Per tal de demanar el número, que haurà d'estar comprès entre 0 i 10, utilitza la funció que de l'exercici 3.

```
Funció: int demanarNumero() // cal controlar que el número sigui
correcte
Procediment: void imprimirTaula (int taula)
```

6. **(TotesTaulesMultiplicar)** Fer un programa que ens mostri les taules de multiplicar des de l'1 fins al 10, utilitzant el procediment de l'exercici 5.

```
Procediment: void imprimirTaula (int taula)
```

7. **(SumaNumerosAnterior)** Fer una funció que passant-li un número positiu ens retorni la suma de tots els nombres des del 0 fins al número introduït

```
int sumaNumerosAnteriors(int numero)
```

Exemple: número: 6

La suma és 21

8. **(QuantitatDigits)** Fer una funció que li passem un nombre positiu i ens retorni la quantitat de dígit que té.

9. **(MostraDigits)** Fer un procediment que li passem un nombre i ens mostri per pantalla els seus dígit

```
Exemple: 456      6,5,4
           4,5,6
```

10. **(Estacio)** Fer una funció que li passem un mes i un dia i ens retorni en quina estació estem

11. **(CanviMoneda)** Fer una funció que li passem un import expressat en euros i una moneda i ens retorni l'import en la moneda seleccionada. Les monedes serán: USD, GBP,JPY,CHF,AUD,CNY. Per a trobar el canvi actual utilitzeu la pàgina web <https://www.infobolsa.es/divisas>

12. **(TriangleCharacter)** Demana un caràcter i un número i imprimeix el següent:

Exemple: 'c', 5

```
Solució: ccccc
         cccc
         ccc
```

CC
C

13. **(Base exponent)** Fer una funció que li passem dos nombre (base i exponent) i ens retorni el valor de la base ^{exponent} (No es pot utilitzar la funció pow)

Exemple: 2 elevat 5: $2*2*2*2*2$

14. **(Factorial)** Fer una funció que ens retorni el factorial del nombre que li passem com a paràmetre.

Fac(4) = $4*3*2*1$

15. **(DiesMesAnyQuantitatDies)** Donada una data determinada per un dia, un mes i un any en format numèric, i una quantitat de dies que ens proporciona l'usuari. Es demana, a partir del darrer dia del mes en curs, escriure els n dies que es demanen. S'haurà de tenir en compte els anys de trapàs.

Seguir els següents passos:

- Escriure una funció per tal de validar l'any introduït per l'usuari. L'any ha d'estar comprès entre 0 i 2020 (ambdós inclosos)

```
int demanaAny()
```

- Escriure una funció que ens retorni un mes correcte (entre 1 i 12)

```
int demanaMes()
```

- Escriure una funció que passant-li el mes ens retorni la quantitat de dies que té el mes

```
int quantitatDiesMes(int mes, int any)
```

- Escriu una funció que ens demani quants dies volem visualitzar. La quantitat de dies com a mínim serà 1 i com a màxim 2000

```
int demanaQuantitatDies()
```

Exemple:

entrada:

Introdueix l'any: 2016

Introdueix el mes: 12

Introdueix el dia: 15

Introdueix quantitat de dies: 40

sortida:

```
1 31/12/2016
2 01/01/2017
3 02/01/2017
4 03/01/2017
5 04/01/2017
6 05/01/2017
7 06/01/2017
8 07/01/2017
9 08/01/2017
10 09/01/2017
11 10/01/2017
12 11/01/2017
13 12/01/2017
14 13/01/2017
15 14/01/2017
16 15/01/2017
17 16/01/2017
18 17/01/2017
19 18/01/2017
20 19/01/2017
21 20/01/2017
22 21/01/2017
23 22/01/2017
24 23/01/2017
25 24/01/2017
26 25/01/2017
27 26/01/2017
28 27/01/2017
29 28/01/2017
30 29/01/2017
31 30/01/2017
32 31/01/2017
33 01/02/2017
34 02/02/2017
35 03/02/2017
36 04/02/2017
37 05/02/2017
38 06/02/2017
39 07/02/2017
40 08/02/2017
```

16. (**Rectangle_Dreta_Esquerra_Centrat**) Fer un programa que demani a l'usuari un nombre natural entre 1 i 10 i un caràcter que pot ser 'C', 'E' o 'D'. El programa ha de dibuixar llavors un triangle depenent de la lletra proporcionada i amb l'alçada que digui el número. Les lletres signifiquen si el triangle ha d'estar

Centrat, a l'Esquerra o a la Dreta. El programa finalitzarà quan l'usuari hagi fet les tres opcions.

Passos que cal seguir:

- a) Fer una funció que passant-li com a paràmetres el mínim i el màxim (1, 10), ens retorni un número introduït per l'usuari que complirà aquests requeriments.

```
int demanarNumero(int min,int max);
```

- b) Fer una funció que ens demani a l'usuari quin triangle vol fer. Caldrà controlar que l'usuari només pugui inserir 'C', 'E' o 'D'.

```
char demanarTriangle();
```

- c) Fer un procediment per a l'opció 'E' que ens mostrarà un triangle a l'esquerra

```
void triangleEsquerra(int num);
```

```
Introduïx el tamany del triangle
5
C:Centrat E:Esquerra D:Dreta
Escull opció: E

*
**
***
****
*****
```

- d) Fer un procediment per a l'opció 'D' que ens mostrarà un triangle a la dreta

```
void triangleDreta(int num);
```



```

Introduïx el tamany del triangle
5
C:Centrat E:Esquerra D:Dreta
Escull opció: D

  *
 **
***
****
*****

```

e) Fer un procediment per a l'opció 'C' que ens mostrarà un triangle centrat

```
void triangleCentrat(int num);
```

```

Introduïx el tamany del triangle
5
C:Centrat E:Esquerra D:Dreta
Escull opció: c

  *
 ***
*****
*****
*****
*****

```

2.1 PAS DE PARÀMETRES: PER VALOR I PER REFERÈNCIA

1. (**IntercanviVariables**) Crea un procediment que intercanviï el valor de dues variables que passarem com a paràmetres i que prèviament haurem introduït per teclat. S'han d'intercanviar els valors entre les variables i no mostrar les variables al revés)

```
void intercanviVariables (int *num1, int *num2)
```

2. (**Trío**) Fer un algoritme per poder jugar al TRIO. Aquest joc tracta de triar tres números del 0 al 9. Llavors el sistema genera la combinació guanyadora de tres números. Cal mostrar per pantalla quants encerts ha tingut l'usuari. S'ha

de fer servir únic procediment per tal de generar la combinació guanyadora i un altre per a obtenir els tres nombres de l'usuari. Els nombres tant de l'usuari com de la combinació guanyadora no es poden repetir. Per a calcular el nombre d'encerts, no cal que els nombres estiguin en la mateixa posició, simplement cal que siguin iguals.

```
void combinacioGuanyadora(int *ord1,int *ord2,int *ord3)
void combinacioUsuari(int *usu1,int *usu2,int *usu3)
int calculaEncerts(int ord1,int ord2,int ord3, int usu1,int usu2,int usu3)
```

3. **(RectanglePle_RectangleBuit)** Fer un algoritme que demani a l'usuari dos nombres naturals entre 2 i 20 (que representent l'alçada i l'amplada respectivament). Posteriorment, demanarà també a l'usuari quina de les dues opcions vol fer 'P' o 'B'. En el cas que l'usuari ens indiqui una 'P', ens haurà de mostrar per pantalla un requadre ple: mentre que si ens indica l'opció 'B', ens haurà de mostrar per pantalla un quadre buit amb l'alçada i l'amplada que hem introduït prèviament.

Cal utilitzar les següents funcions o procediments:

```
void demanarParametres(int *alcada, int *amplada);
char opcio();
void imprimirRequadrePle(...);
void imprimirRequadreBuit(...);
```

4. **(Categoria equip)** Fer un algoritme per portar la puntuació d'un equip de futbol a la lliga. Aquest algoritme demanarà deu resultats de deu partits, seguint el format de local – visitant, on el nostre equip començarà jugant com a local i anirà alternant a visitant. Si l'equip guanya, anotarem tres punts. Si empata, un i si perd, cap punt. Al final s'ha de mostrar el número de punts assolits per l'equip. Depenent del número de punts que l'equip assoleixi a aquesta minilliga, l'equip pot pujar de categoria, mantenir la categoria o baixar de categoria, segons la següent taula:

Punts	Resultat
De 0 a 10	Baixa
De 11 a 25	Es manté
Més de 25	Puja

JORNADA	LOCAL	VISITANT	PUNTS
1	2	2	1
2	3	5	3
3	5	3	3
4	5	0	0
5	0	5	0
6			
7			
8			
9			
10			

5. JOC DEL CUC: S'ha de programar el joc del cuc.

El nostre cuc, és un cuc molt simpàtic que té el cap a la posicio 40,10 de la pantalla. A partir d'aquí, anirà creixent cap a d'alt, baix, dreta o esquerra. Sempre que creix ho fa amb longitud d'1. Només té un inconvenient, si toca un dels marges o bé si es trepitja el seu propi cap mor. Per a fer el nostre cuc més simpàtic, cada cop que creix, li canviarem el color.

INDICACIONS

El cuc s'haurà de moure dins del requadre 5,1 i 80,20

El cap del cuc comença a la posició 40,10

FUNCIONS:

La funció `getkey()` retorna un número, depenent de la tecla que ha premut l'usuari. Aquesta funció pertany a la llibreria `rlutil.h`

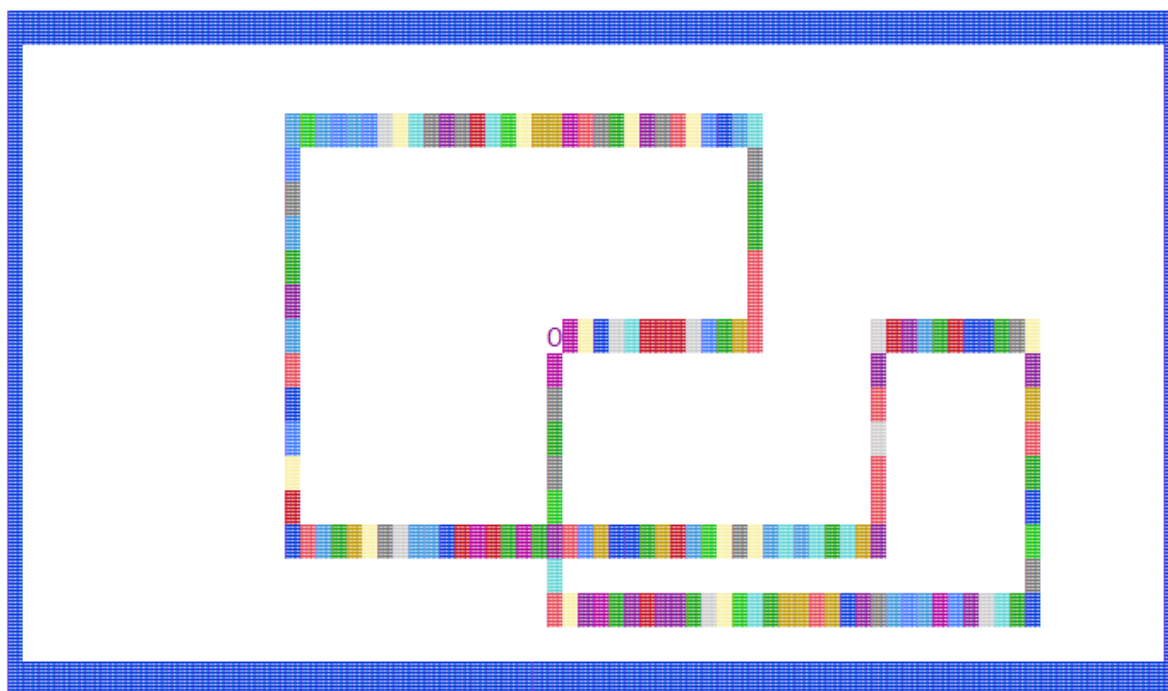
```
KEY_UP      = 14
KEY_DOWN    = 15
KEY_LEFT    = 16
KEY_RIGHT   = 17
```

Per a fer el requadre utilitzarem el caràcter 175, que será el mateix per al cos.

Estructura del programa

```
x=40,y=10;
pintaRectangle();
while (....)
{
    pintaCaracter(x,y,caracter);
    tecla=afagarTecla();
    actualitzaPosicioCuc(tecla,&x,&y);
    limit=controlDeLimits(tecla,x,y);
    cap=controlCap(x,y);
}
```

Hi podeu afegir les modificacions que us agradin



6. **(GPS)** Es tracta de fer un programa que donada la posició d'un GPS, introduïda per l'usuari, ens mostri els punts que tenim al seu voltant.

Per a demanar la posició inicial del GPS, l'usuari haurà d'introduir una posició x i una posició y. La posició x haurà d'estar controlada entre els marges (0 i 79), mentre que la posició y ha d'estar controlada entre (0 i 20). En la posició del nostre GPS mostrarem un 'X' de color verd.

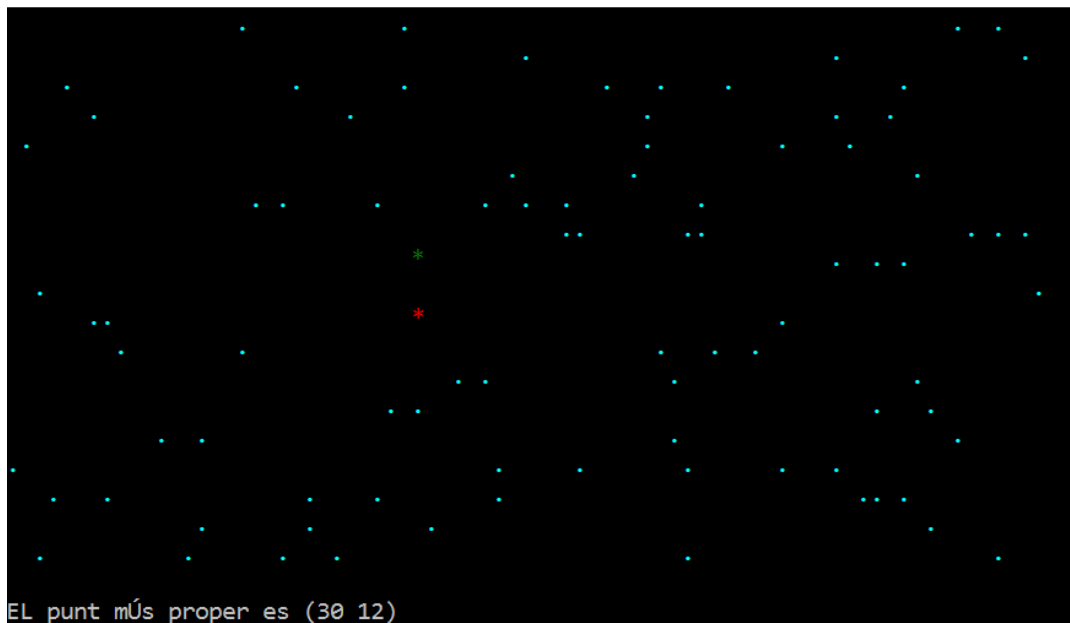
Posteriorment, cal generar 100 punts aleatoris, que es mostraran a la pantalla en color blau cel i el caràcter serà un `.`.

Finalment, volem conèixer quin és el punt que està més a prop del nostre GPS, mostrant aquest punt en color vermell i amb el format de `*`. Per a calcular la distància entre dos punts aplicarem pitàgoras, fent servir la formula sqrt que ens retorna l'arrel quadrada d'un número. A continuació es mostra un exemple de com es calcula:

$$\begin{aligned}
 P1 = (3,6) \quad P2 = (7,9) \quad DIST &= \text{SQRT}((7-3)^2 + (9-6)^2) \\
 &= \text{SQRT}((4)^2 + (3)^2) \\
 &= \text{SQRT}(16 + 9) \\
 &= \text{SQRT}(25) \\
 &= 5
 \end{aligned}$$

Es recomana posar el fons de la pantalla en color negre, per tal de simular millor l'espai. Per acabar i al final de la pantalla, escriu quines són les coordenades del punt més proper al nostre GPS.

Exemple:



Cal utilitzar les següents funcions o procediments:

```
void demanarGPS(int *gpsx, int *gpsy);  
void generarPunt(int *, int *y);
```

```
void mostrarPunt (int x, int y, int color);  
float calcularDistancia(int x, int y, int qpsx, int qpsy);
```

Per poder utilitzar la funció sqrt, es necessita la llibreria
`#include <math.h>`

Per a pintar en color, cal utilitza

```
#include <stdio.h>  
#include <stdlib.h>  
#include "rlutil.h"  
  
int main (void)  
{  
    int i;  
    for (i=0;i<=15;i++)  
    {  
        setColor(i);  
        printf("%4d: Hola\n",i);  
    }  
    setColor(RED);  
    printf("\n\nTambe es pot posar directament el nom del color");  
    getch();  
}
```

```
1: Hola  
2: Hola  
3: Hola  
4: Hola  
5: Hola  
6: Hola  
7: Hola  
8: Hola  
9: Hola  
10: Hola  
11: Hola  
12: Hola  
13: Hola  
14: Hola  
15: Hola
```

Tambe es pot posar directament el nom del color

7. 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.

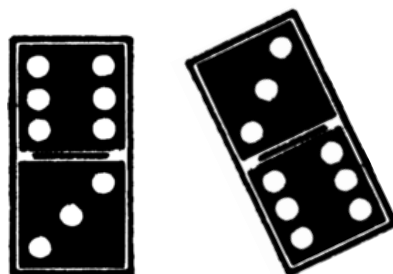


Introdueix la primera fitxa del domino: 1 6

Introdueix la segona fitxa del domino: 6 1

Introdueix la segona fitxa del domino: 6 6

Les fitxes (1-6) i (6-6) es poden ajuntar



Introdueix la primera fitxa del domino: 1 2

Introdueix la segona fitxa del domino: 1 2

Introdueix la segona fitxa del domino: 5 6

Les fitxes (1-2) i (5-6) es no es poden ajuntar

8. 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.



(6-3) (1-2) (5-1) (3-5) (2-2) (5-4) (4-5)

3 VECTORS

1. (**Vectors**) Definir un vector de 6 caselles buides i fer els següents exercicis:

1.1. Omplir el vector amb els valors 4,3,5,6,2 i 10

1.2. Mostrar per pantalla el vector

```
v[0]=4  
v[1]=3  
v[2]=5  
etc
```

1.3. Indicar quants números són parells.

1.4. La suma de tots els valors del vector.

1.5. La suma dels números parells

1.6. Modifica el valor de cadascuna de les caselles, de tal manera que cadascuna tingui el seu valor més el valor de la casella anterior. Fixa't que la primera casella val 4.

```
v[0]=4  
v[1]=7  
v[2]=12  
v[3]=18  
etc...
```

1.7. Intercanviar el valor de la primera casella amb la darrera casella

```
v[0]=10  
v[1]=2  
v[2]=6  
v[3]=5  
v[4]=3  
v[5]=4
```

2. (**Promig**) A partir d'un vector de 6 caselles, demanar a l'usuari un nombre comprés entre 0 i 6 i omplir tantes caselles com aquest número indiqui. El valor de les caselles seran valors aleatoris compresos entre 0 i 10. Es demana fer un programa que calculi el valor promig dels números aleatoris generats.


```
Introduix quant numeros vols introduir: 5
Contingut del vector:
0: 7
1: 1
2: 4
3: 6
4: 8
El valor promig es 5.20
```

```
Introduix quant numeros vols introduir: 7
0
No s'ha introduit cap valor
```

3. (**Maxim**) A partir de l'exercici anterior, el programa ens ha d'indicar quin és el valor màxim dels dels números aleatoris generats.

```
Introduix quant numeros vols introduir: 5
Contingut del vector:
0: 3
1: 9
2: 5
3: 6
4: 9
El valor maxim dels numeros generats es 9
```

4. (**Cerca**) A partir d'un vector de 10 caselles, demanar a l'usuari un nombre comprés entre 0 i 10 i omplir tantes caselles com aquest número indiqui. El valor de les caselles seran valors aleatoris compresos entre 1 i 100. Posteriorment demanar un número a l'usuari, comprés entre 1 i 100 i cercar si aquest valor existeix dins del vector.

```
Introduix quant numeros vols introduir: 10
Contingut del vector:
0: 24
1: 13
2: 4
3: 41
4: 23
5: 93
6: 35
7: 17
8: 76
9: 86
Introduix quin numero vols cercar: 90
EL numero 90 no esta dins del vector
```

```
Introduix quant numeros vols introduir: 10
Contingut del vector:
0: 39
1: 12
2: 91
3: 88
4: 25
5: 13
6: 49
7: 25
8: 97
9: 29
Introduix quin numero vols cercar: 13
EL numero 13 esta dins del vector
```

5. (**CercaSentinella**) Refer l'exercici anterior utilitzant l'opció del sentinella. Per a fer l'opció amb sentinella, el vector es defineix amb una nova posició, on hi col·loquem el valor a cercar. D'aquesta manera sempre el programa el trobarà i podem simplificar la cerca.

```
Introduix quant numeros vols introduir: 10
Contingut del vector:
0: 29
1: 24
2: 9
3: 90
4: 48
5: 15
6: 7
7: 35
8: 25
9: 93
Introduix quin numero vols cercar: 94
EL numero 94 no esta dins del vector
```

```
Introduix quant numeros vols introduir: 6
Contingut del vector:
0: 8
1: 62
2: 99
3: 85
4: 94
5: 10
Introduix quin numero vols cercar: 10
EL numero 10 esta dins del vector
```

6. (**VectorSenseRepetits**) Defineix un vector de 10 caselles i demana números a l'usuari fins que el vector estigui ple.

Introduïrem el nou números dins del vector, sempre i quan, aquest no estigui ja introduït prèviament. Al finalitzar caldrà mostrar el vector amb els 10 números diferents.

```
Introdueix numero de la pos 0: 2
Introdueix numero de la pos 1: 3
Introdueix numero de la pos 2: 2
Introdueix numero de la pos 2: 5
Introdueix numero de la pos 3: 23
Introdueix numero de la pos 4: 25
Introdueix numero de la pos 5: 6
Introdueix numero de la pos 6: 7
Introdueix numero de la pos 7: 89
Introdueix numero de la pos 8: 25
Introdueix numero de la pos 8: 34
Introdueix numero de la pos 9: 47

Contingut del vector:
0: 2
1: 3
2: 5
3: 23
4: 25
5: 6
6: 7
7: 89
8: 34
9: 47
```

7. **(ComptaQuantesVegades)** Definir un vector de 100 caselles. Posteriorment, es demanarà números a l'usuari mentre que aquest no introdueixi un -1. Els números vàlids estaran compresos entre el 0 i el 99. Al finalitzar el programa, ens ha d'indicar quantes vegades s'han introduït cadascun dels números vàlids.

```
Introdueix numero: 6
Introdueix numero: 8
Introdueix numero: 2
Introdueix numero: 45
Introdueix numero: 8
Introdueix numero: 8
Introdueix numero: 90
Introdueix numero: 104
56
Introdueix numero: 4
Introdueix numero: 0
Introdueix numero: -1

Contingut del vector:
0: 1
2: 1
4: 1
6: 1
8: 3
45: 1
56: 1
90: 1
```

8. **(InserirOrdenadament)** Definir un vector de n caselles. Posteriorment, l'usuari introduirà nombres i aquests s'introduiran ordenadament dins del vector. Finalitza el programa quan tot el vector estigui ple. En cada passada mostrarem el vector amb els seus valors

```
n=5
Numero introduït per l'usuari:6
Vector: 6
Numero introduït per l'usuari:1
Vector: 1 6
Numero introduït per l'usuari:2
Vector: 1 2 6
Numero introduït per l'usuari:6
Vector: 1 2 6 6
Numero introduït per l'usuari:3
Vector: 1 2 3 6 6
```

9. **(SerieFibonacciNTermes)** Fer un algoritme que calculi el terme n de la sèrie de Fibonacci. Aquesta sèrie està composta de la següent manera:

```
n:11
F1 = 0;
F2 = 1;
Fn = Fn-1 + Fn-2;
Aquesta sèrie és: 0,1,1,2,3,5,8,13,21,34 i 55
```

Com a màxim ens demanarà el terme 30. Al final del programa ens haurà de mostrar tots els termes fins al terme número n introduït per l'usuari.

```
Introdueix quants termes vols de la serie: 20

Serie Fibonacci de 20 termes :
  0      1      1      2      3      5      8      13      21      34
55      89     144     233     377     610     987     1597     2584     4181
```

```
Introdueix quants termes vols de la serie: 1

Serie Fibonacci de 1 termes :
  0
```

10. Defineix dos vectors i cal comprovar si tots els valors del primer vector estan dins del segon vector. Els dos vectors tindran una capacitat màxima de 10 elements. Es demanarà a l'usuari quants elements vol omplir tant del primer com del segon vector.

```
Exemple 3,6  
a:2,4,5  
b:2,2,4,4,7,5  
Resposta: Tots els valors d'a están en b
```

```
Exemple 3,6  
a:2,4,5  
b:2,2,4,4,7,2  
Resposta: Tots els valors d'a no están en b
```

11. Definir un vector de 8 caselles, i omplir només les 6 primeres. Posteriorment eliminar el contingut de la casella 2; tot desplaçant el contingut de la resta de caselles cap a l'esquerra. Cal controlar el cas que el vector estigui buit.
12. Definir un vector de 8 caselles, i omplir només les 6 primeres. Posteriorment insereix a la posició 3 el valor de 1; tot desplaçant el contingut de la resta de les caselles cap a l'esquerra. Recorda comprovar què passa si el vector està ple.

4 EXERCICIS EXTRES DE VECTORS

1. 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.



Resolt l'exercici desant cada una de les fitxes en un vector de dues caselles

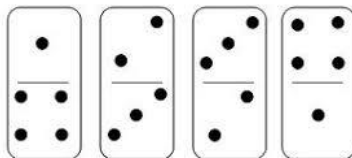
```
Introdueix la primera fitxa del domino: 1 2
Introdueix la segona fitxa del domino: 2 1
Introdueix la segona fitxa del domino: 3 1
Les fitxes (1-2) i (3-1) es poden ajuntar
```

2. 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

```
Les cartes generades son:
0      1      2      3      4      5      6      1      4      5
0      1      2      3      4      5      6      6      2      3
Ole ole hem trobat tots els dobles
```

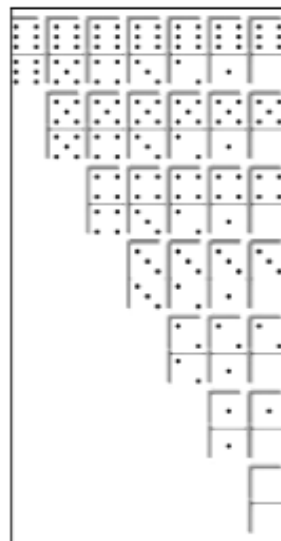
```
Les cartes generades son:
0      1      5      3      4      5      6      1      4      5
0      1      2      3      4      5      6      6      2      3
No hem trobat tots els dobles
```

3. 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]={6,6,6,6,6,6,6,5,5,5,5,5,4,...}
```

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

(6-6)	(6-5)	(6-4)	(6-3)	(6-2)	(6-1)	(6-0)
(5-5)	(5-4)	(5-3)	(5-2)	(5-1)	(5-0)	
(4-4)	(4-3)	(4-2)	(4-1)	(4-0)		
(3-3)	(3-2)	(3-1)	(3-0)			
(2-2)	(2-1)	(2-0)				
(1-1)	(1-0)					
(0-0)						



4. 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:

1. Generar una fitxa utilitzant un vector de dues caselles
2. Comprovar si la fitxa ja existeix dins de les fitxes que ja tenim desades
3. En el cas que la fitxa sigui nova, l'afegirem dins del vector.
4. Un cop s'han desat totes les fitxes, mostrar-les per pantalla.

(6-2)	(0-4)	(4-3)	(3-1)	(4-5)	(2-0)
(6-0)	(0-0)	(3-0)	(1-5)	(3-2)	(1-6)
(1-2)	(3-5)	(6-6)	(4-4)	(4-6)	(1-0)
(4-1)	(6-5)	(6-3)	(0-5)	(2-4)	(5-2)
(3-3)	(1-1)	(2-2)	(5-5)		

Passada 1	(5-3)				
Passada 2	(5-3)	(4-6)			
Passada 3	(5-3)	(4-6)	(2-4)		
Passada 4	(5-3)	(4-6)	(2-4)	(5-6)	
Passada 5	(5-3)	(4-6)	(2-4)	(5-6)	(6-1)
Passada 6	(5-3)	(4-6)	(2-4)	(5-6)	(6-1)
Passada 7	(5-3)	(4-6)	(2-4)	(5-6)	(6-1)
Passada 8	(5-3)	(4-6)	(2-4)	(5-6)	(6-1)

5. 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.

Les cartes generades son:

6 5 0 2 3 0 1 6 5 4 4 1 2 4 4 5 5 1 4 2
5 2 0 1 3 0 1 1 2 2 1 1 2 1 3 1 3 1 2 1

La fitxa mes alta es (6-5)

Les cartes generades son:

5 5 5 4 3 3 2 3 5 0
4 2 3 4 3 1 2 1 3 0

La fitxa mes alta es (5-4)

6. 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.



El valor a introduir es 1

1

El valor a introduir es 5

1 5 5 5 5 5

El valor a introduir es 2

1 5 5 5 5 5 2 2

El valor a introduir es 1

1 5 5 5 5 5 2 2 1

El valor a introduir es 5

1 5 5 5 5 5 2 2 1 5 5 5 5 5

El valor a introduir es 5

1 5 5 5 5 5 2 2 1 5 5 5 5 5 5 5 5 5 5

El valor a introduir es 3

El valor a introduir es 5

5 5 5 5 5

El valor a introduir es 1

5 5 5 5 5 1

El valor a introduir es 5

5 5 5 5 5 1 5 5 5 5 5

El valor a introduir es 3

5 5 5 5 5 1 5 5 5 5 5 3 3 3

El valor a introduir es 3

5 5 5 5 5 1 5 5 5 5 5 3 3 3 3 3 3

El valor a introduir es 3

5 5 5 5 5 1 5 5 5 5 5 3 3 3 3 3 3 3 3 3