Dossier Exercicis M3 UF1 : Programació estructurada

Claudina Riaza Español

[1 ALGEBRA 2](#_Toc523402756)

[2 ALGORISMES SIMPLES 1](#_Toc523402757)

[2.1 ESTRUCTURA SEQÜENCIAL 1](#_Toc523402758)

[2.2 FUNCIÓ CONDICIONAL 1](#_Toc523402759)

[2.3 ITERACIONS 7](#_Toc523402760)

[3 FUNCIONS I PROCEDIMENTS 20](#_Toc523402761)

[3.1 PAS DE PARÀMETRES: PER VALOR I PER REFERÈNCIA 24](#_Toc523402762)

[4 VECTORS (UF1) 30](#_Toc523402763)

# ALGEBRA

Fer una expressió que ens digui si estem en cap de setmana. Disposem d’una variable dia, que valdrà entre 1 i 7.

Fer una expressió que ens indiqui si és dia laborable.

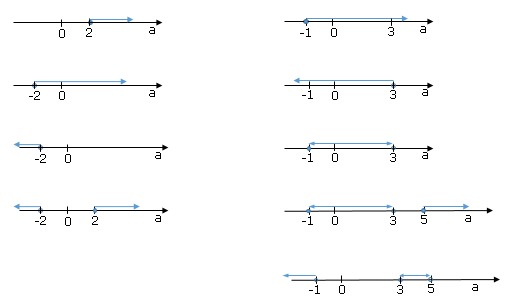
Fer una expressió que ens indiqui quan hem d’anar a judo, tenint en compte que hi anirem si és dijous i no plou. Disposarem de les variables dia i plou.La variable plou val Cert o Fals.

Fer una expressió que ens indiqui quan hem d’anar a gimnàs. Anirem al gimnàs si és dilluns o dimarts; sempre i quan no plogui

Fer una expressió que ens indiqui quan no anirem a judo. Omple la taula amb l’expressió que has obtingut

|  |  |  |  |
| --- | --- | --- | --- |
| **dia** | **plou** |  |  |
| 1 | Fals |  |  |
| 1 | Cert |  |  |
| 4 | Fals |  |  |
| 4 | Cert |  |  |
| 7 | Fals |  |  |
| 7 | Cert |  |  |

Cerca les següents expressions



Fer la taula de veritat de la següent expressió. Verifica amb l’esquema que ens ha donat. En la taula de veritat, els valors que li donarem a la ‘a’ són: -4,-1,0,3,4,5 i 7

a<-1 o (a>=3 i a<5)

-1

a

0

3

5

Fes un dibuix indicant quan es compleix cadascuna de les següents expressions

1. a>=2
2. no (a>=2)
3. a>=10 i a<=20
4. a>=10 o a<=20
5. no(a>=10 i a<=20)
6. no (a>=0) o no (a<=10)
7. no (a>=0) i no (a<=10)

# ALGORISMES SIMPLES

## 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

## FUNCIÓ 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

(**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

(**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

(**Nat\_Senar**) Fer un algoritme que donat un nombre natural, indiqui si és o no un nombre natural senar.

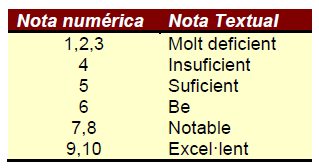
Exemple: num=1201

Resultat: El número 1201 és senar

(**Equació\_segon\_grau**) Fer un algoritme que calculi el resultat d’una equació de segon grau. L’usuari ha d’introduir els valors per a a, b i c. S’ha de fer servir la fórmula de resolució de les equacions de segon grau:



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



Exemple

Nota introduïda per l’usuari: 9

Resultat: Excel·lent

Nota introduïda per l’usuari: 15

Resultat: No es una nota correcta

(**Num\_Vocal**) Fer un algoritme que donada una lletra vocal (majúscula o minúscula) mostri quin nombre de lletra és.

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

**(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

**(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. |

(**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 €

(**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:



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

(**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

**(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”

**(Ordena\_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 realizar l’exercici omple la taula que s’adjuunta

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |  |  |  |

(**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 |  |  |  |  |  |

(**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 |  |  |  |  |  |  |  |

## 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.

Exemple:

Número: 8

Resposta: 1, 2 , 3, 4, 5, 6, 7, 8

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

Resposta:

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

**(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’esser igual al que s’adjunta com a resposta.

Exemple:

Número: 3

Resposta:

3 x 0 = 0

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

3 x 10 = 30

**(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

Per exemple: numero=11 Resposta: 0, 2, 4, 6, 8 i 10

**(><)** 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

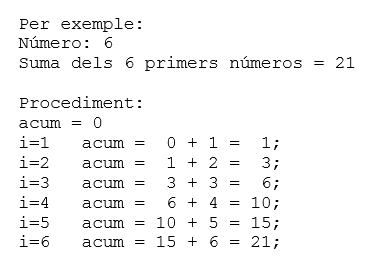
Resultat: 10 9 8 7 6 5 4 3 2

**(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.

Per exemple:

Número: 6

Suma dels 6 primers números = 21



**(Suma parell-senar)** Fer un algoritme que demani 10 números enters a l’usuari. El programa ha de mostrar per pantalla la suma de tots els números parells i la suma de tots els números senars introduïts per l’usuari.

Exemple:

Nombres introduits per l’usuari: 1 5 7 8 2 4 5 8 4 6

Resultat: Total parells: 32; Total senars: 18

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

Resultat: 2250

**(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 introduits per l’usuari: 1 5 7 4 2 4 5 8 4 6 -1

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

**(MesGranMesPetit)**: Calcula el numero més 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 introduits per l’usuari: 1 5 7 4 2 4 5 8 4 6 0

Resultat: Maxim: 8; Minim: 1

Nombres introduits per l’usuari: 0

Resultat: No s’ha introduït cap valor correcta

**(Factorial):** Calcula el factorial del número natural introduit 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

**(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 |

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

Exemple:

Usuari introdueix: afgw34.8rz

Resposta: L’usuari ha introduit 9 lletres

**(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.

Exemple:

Usuari introdueix: arcftadrfq

Resposta: L’usuari ha introduit 2 cops el caracter ‘a’

**(Comptar ‘a’s limitat)** Fer un algoritme que demani un número natural entre 5 i 10 a l’usuari. L’algoritme ha de demanar desprès 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 introduit 1 cop el caracter ‘a’

**(Lletra n vegades)** Fer un algoritme que demani a l’usuari un nombre n i un caràcter c. L’algoritme ha de indicar si la lletra c apareix, al menys, n vegades a una sèrie de lletres (acabada per ‘z’) introduïda per l’usuari.

Exemple:

Usuari introdueix: n=3, c=’d’

Llistat de caràcters que introdueix l’usuari: derftd3a5ygx.z

Resposta: No ha introduit 3 cops el carácter ‘d’ dins de la cadena

Usuari introdueix: n=3, c=’d’

Llistat de caràcters que introdueix l’usuari: derdtd3d5ygx.z

Resposta: Si que s’han introduit com a minim 3 cops el carácter ‘d’ dins de la cadena

**(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.

Exemple

Número: 6

Número: 3

Resposta ordinador: Si, ja que 6 és divible entre 3

Número: 8

Resposta ordinador: No, ja que 3 no es divible entre 8

Número: 4

Resposta ordinador: Si, ja que 8 es divisible entre 4

Número:0

Resposta ordinador: Acabem el joc

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

F1 = 0;

F2 = 1;

Fn = Fn-1 + Fn-2;

Aquesta s\_erie \_es: 0,1,1,2,3,5,8,13,21,34,55.....

Exemple:

Número: 6

Resultat: 5

**(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 caracters quan l’usuari posi un ‘.’

Exemple:.

Usuari introdueix: LArrLalartla.

Resposta: L’usuari a introduit 4 cops la sequencia ‘la’

**(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.

Resultat: 2820

Nombres: 21,42,63,84,120,141,162,183,240,261,282,360,381,480

**(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.

Exemple

Número: 4

Caracters introduïts per l’usuari: afgtyei.

Resultat: No hi ha les 4 vocals que es demanen

Número: 4

Caracters introduïts per l’usuari: afgtyeio

Resultat: Hi ha les 4 vocals que es demanen.

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

**(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.

Exemple

Caracters introduïts per l’usuari: afgtyeaaai.

Resultat: No hi ha totes les vocals

Caracters introduïts per l’usuari:afgrioue

Resultat: S’han trobat com a minim un cop cadascuna de les vocals.

**(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

**(Compta\_Paraules)** Donada una frase acabada en ‘.’. Calcula quantes paraules s’han introduït. Cal tenir en compte que els caràcters que introduirem seran tots correctes i que entre paraula i paraula hi poden haver més d’un espai. Recordeu el l’espai en blanc equival al codi ascii 10

Exemple;

Caracters introduïts per l’usuari:

H

o

l

a

a

d

e

u

a

.

Resultat: Total de paraules: 3

**(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 canvïin d’estat

(**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

(**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 deu números demanats també a l'usuari, és més gran que el primer número natural (n) demanat.

Exemple

Número: 15

Sèrie de números que introduiex l’usuari: 2, 8, 5, 1

Resultat: La suma és més gran // fixa’t que ja no cal demanar els deu números a l’usuari

Número: 150

Sèrie de números que introduiex l’usuari: 2, 8, 5, 1, 3, 5, 6, 7, 1, 2

Resultat: La suma dels números introduïts no és més gran que el número

(**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.

Exemple:

Número aleatori a esbrinar: 1456

Número d’intents (introduït per l’usuari): 5

Intent usuari: 5000

Resposta Ordinador: Més petit

Intent usuari: 2500

Resposta Ordinador: Més petit

Intent Usuari: 1250

Resposta Ordinador: Més gran

Intent Usuari: 1400

Resposta Ordinador: Més gran

Intent Usuari: 1450

Resposta Ordinador: Has perdut!, Has esgotat tots els intents

Per a crear números aleatoris 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

}

Cerca quin és el resultat d’aquest programa

#include <stdio.h>

main()

{

int i,j,x=0;

for (i=0;i<5;++i)

for (j=0;j<i;++j)

{

x+=(i+j-1);

printf(“%d”,x);

}

printf(“\nx=%d”,x);

}

**(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.

Resposta:

1 x 0 = 0

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

2 x 0 = 0

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

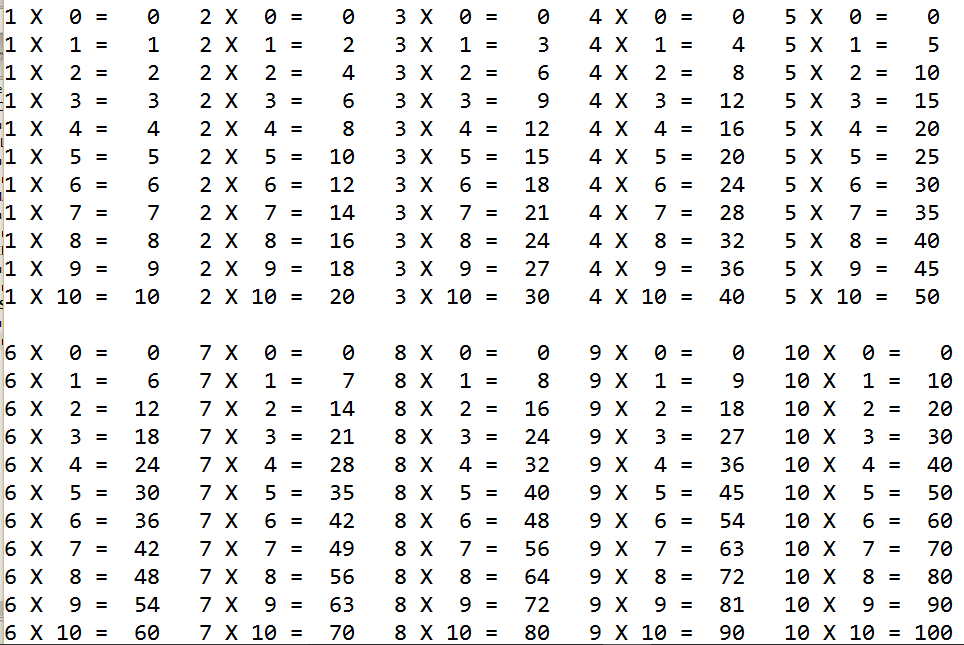
2 x 9 = 18

2 x 10 = 20

etc..

**(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.

Resposta



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");

}

(**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:

Exemple:

A = 4

B = 3

Resultat:

\*\*\*\*

\*\*\*\*

\*\*\*\*

(**Rectangle\_Buit**) Fer una modificació de l'algoritme anterior, que dibuixi un rectangle buit.

Exemple:

A = 4

B = 3

Resultat:

\*\*\*\*

\* \*

\*\*\*\*

# 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

1. (**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

1. (**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)

1. **(NotaNumericaaLLetres5)** Realitzar l’exercici número 3 (demanar nota i visualitzar 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

1. (**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)

1. (**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)

1. (**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

1. (**QuantitatDigits**) Fer una funció que li passem un nombre positiu i ens retorni la quantitat de dígits que té.
2. (**MostraDigits**) Fer un procediment que li passem un nombre i ens mostri per pantalla els seus digits

Exemple: 456 6,5,4

4,5,6

1. (**Estacio**) Fer una funció que li passem un mes i un dia i ens retorni en quina estació estem
2. (**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 página web https://www.infobolsa.es/divisas
3. (**TriangleCaracter**) Demana un carácter i un número i imprimeix el següent:

Exemple: ‘c’, 5

Solució: ccccc

cccc

ccc

cc

c

1. (**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

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

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

1. (**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

1. (**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 finalitarà quan l’usuari hagi fet les tres opcions.

Passos que cal seguir:

1. 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.

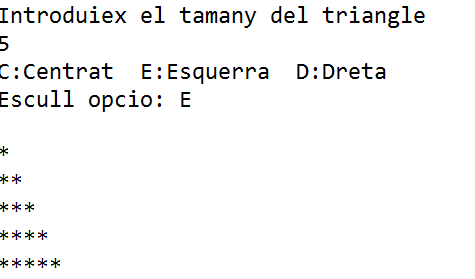
int demanarNumero(int min,int max);

1. 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();

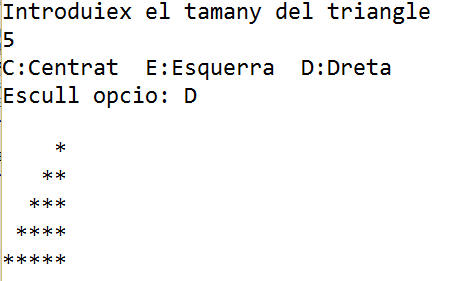
1. Fer un procediment per a l’opció ‘E’ que ens mostrarà un triangle a l’esquerra

void triangleEsquerra(int num);



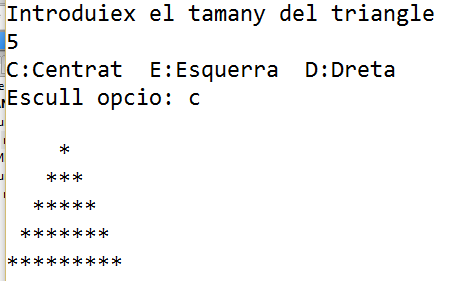
1. Fer un procediment per a l’opció ‘D’ que ens mostrarà un triangle a la dreta

void triangleDreta(int num);



1. Fer un procediment per a l’opció ‘C’ que ens mostrarà un triangle centrat

void triangleCentrat(int num);



## 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)

1. (**Trio**) 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)

1. (**RectanglePle\_RectangleBuit**) Fer un algoritme que demani a l'usuari dos nombre 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.

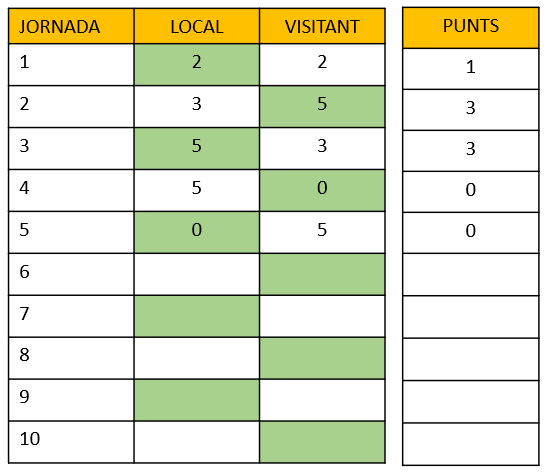
Cal utilitzar les següents funcions o procediments:

void demanarParametres(int \*alcada, int \*amplada);

char opcio();

void imprimirRequadrePle(...);

void imprimirRequadreBuit(...);

1. (**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 |

1. **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 posición 40,10

FUNCIONS:

La funció getkey() retorna un número, depenent de la tecla que ha premut l’usuari. Aquesta funió 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();

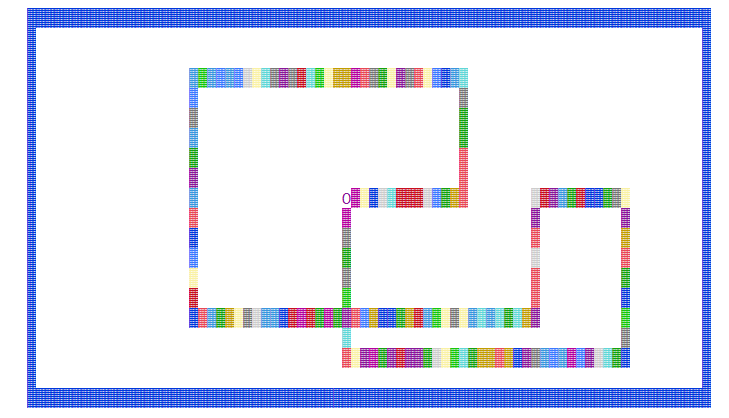
actualitzaPosicioCuc(tecla,&x,&y);

limit=controlDeLimits(tecla,x,y);

cap=controlCap(x,y);

}

Hi podeu afegir les modificacions que us agradin

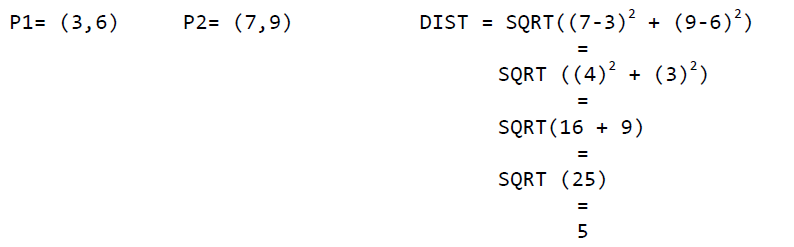


1. **(GPS**) Es tracta de fer un programa que donada la posició d’un GPS, introduïda per l’usuari, ens mostri els punts que tenim al seu voltant.

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 carácter serà un ‘.’.

Finalment, volem conèixer quin és el punt que està més a prop del ostre 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:



Es recomana posar el fons de la pantalla en color negre, per tal de simular millor l’espai. Per acabar i al final del 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. (**Fraccions**) Realitza un programa que ens permeti operar amb fraccions. Per a realitzar el programa cal demana a l’usuari les dues fraccions. Una fracció ve determinada a través d’un numerador i d’un denominador (num i den). Alhora de demanar les fraccions cal tenir en compte que el denominador mai pot ser 0.

Un cop l’usuari ha introduït les dues fraccios, el programa ens ha de demanar quina de les següents opcions vol realitzar ‘+’, ‘-’, ‘\*’ o ‘/’ i mostrar per pantalla el seu resultat.

Finalment, l’usuari podrà executar novament tot el programa en el cas que ho desitji.

Exemple:

|  |  |
| --- | --- |
|  |  |
|  |  |

Cal utilitzar les següents funcions o procediments:

void demanarFraccio(int \*num, int \*den);

char opcio();

void sumaFraccio(int num1, int den1, int num2, int den2, int \*numres, int \*denres);

etc.

void imprimirResultat (int num1, int den1, int num2, int den2, int numres, int denres, int opcio);

# VECTORS

1. (**Vectors**) Definir un vector de 6 caselles buides i fer els següents exercicis:
   1. Omplir el vector amb els valors 4,3,5,6,2 i 10
   2. Mostrar per pantalla el vector

v[0]=4

v[1]=3

v[2]=5

etc

* 1. Indicar quants números són parells.
  2. La suma de tots els valors del vector.
  3. La suma dels números parells
  4. 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. Intercanviar el valor de la primera casella amb la darrera casella

v[0]=10

v[1]=2

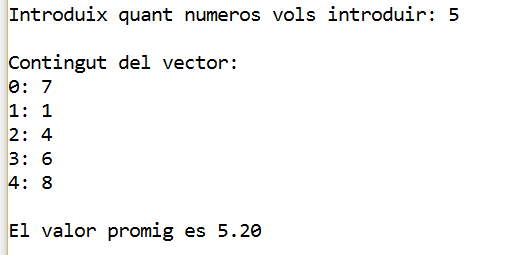
v[2]=6

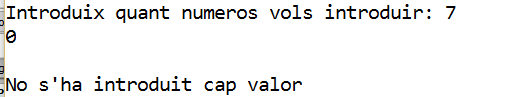
v[3]=5

v[4]=3

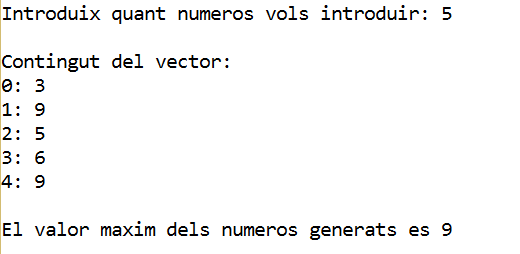
v[5]=4

1. (**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.

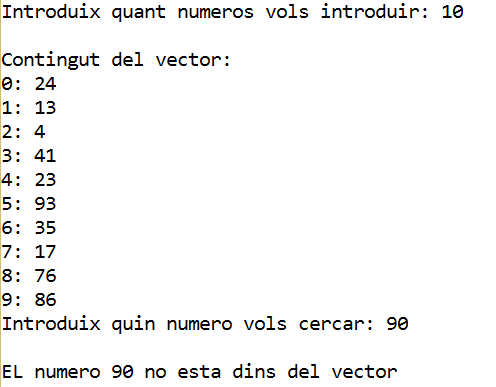
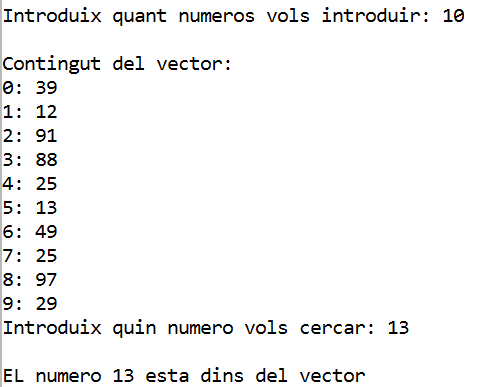




1. (**Maxim**) A partir de l’exercici anterior, el programa ens ha d’indicar quin és el valor màxim dels dels números aleatoris generats.



1. (**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.

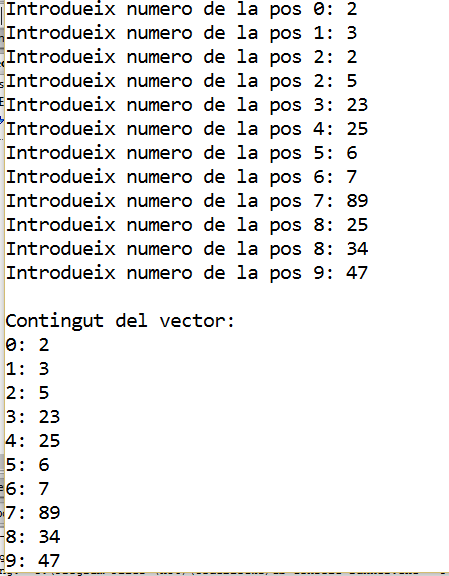
 

1. (**CercaSentinella**) Refer l’exercici anteior utilitzant l’opció del sentinella. Per a fer l’opció amb sentinella, el vector es defineix amb una nova posició, on hi col·loquem el valor a cercar. D’aquesta manera sempre el programa el trobarà i podem simplificar la cerca.

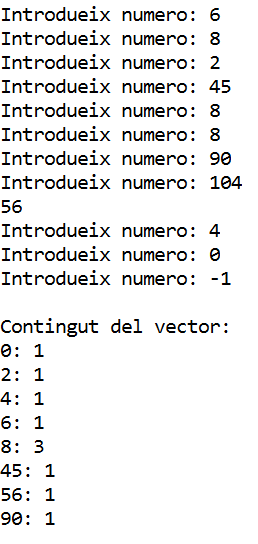
|  |  |
| --- | --- |
|  |  |

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

Introduirem 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.



1. (**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.



1. (**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

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

n:11

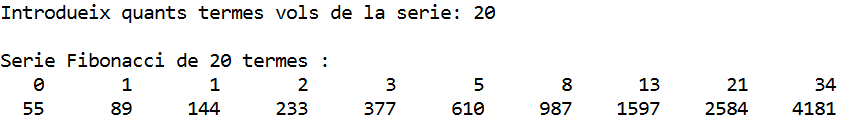
F1 = 0;

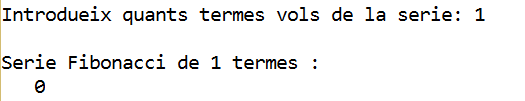
F2 = 1;

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.





1. 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

1. 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.
2. 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.