



TEORIA

PROGRAMACIÓ CIENTÍFICA

T2

Variables i dades
(part 2: operadors)

INTERACCIÓ AMB L'USUARI

ENTRADA/SORTIDA ESTÀNDARD

Interacció amb l'usuari

Entrada i sortida de teclat i pantalla

- Sovint trobem programes que necessiten la interacció de l'usuari:
 - Introduir dades
 - Emplenar formularis
 - Escollir opcions d'un menú
 - Prémer botons per processar la informació, sortir del programa, etc.

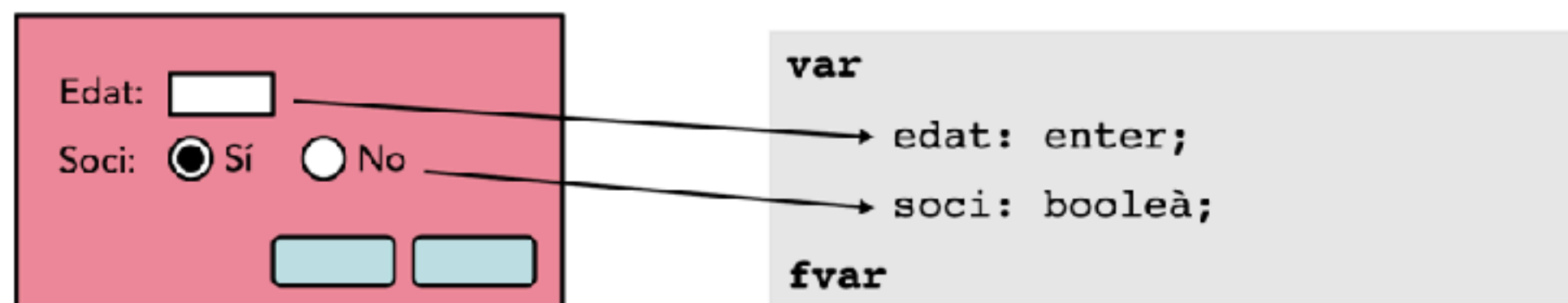
Interacció amb l'usuari

Entrada i sortida de teclat i pantalla

- Sovint trobem programes que necessiten la interacció de l'usuari:
 - Introduir dades
 - Emplenar formularis
 - Escollir opcions d'un menú
 - Prémer botons per processar la informació, sortir del programa, etc.
- La majoria de programes actuals que han d'interactuar amb l'usuari presenten una interfície gràfica d'usuari (**GUI**, Graphic User Interface))

– Una interfície gràfica d'usuari conté elements per a què l'usuari introdueixi dades o bé interaccioni amb el programa:

- Les accions sobre els botons governen el flux d'execució del programa, activen processos, etc.
- Els elements de formulari permeten emplenar els valors de les variables que farà servir el programa.

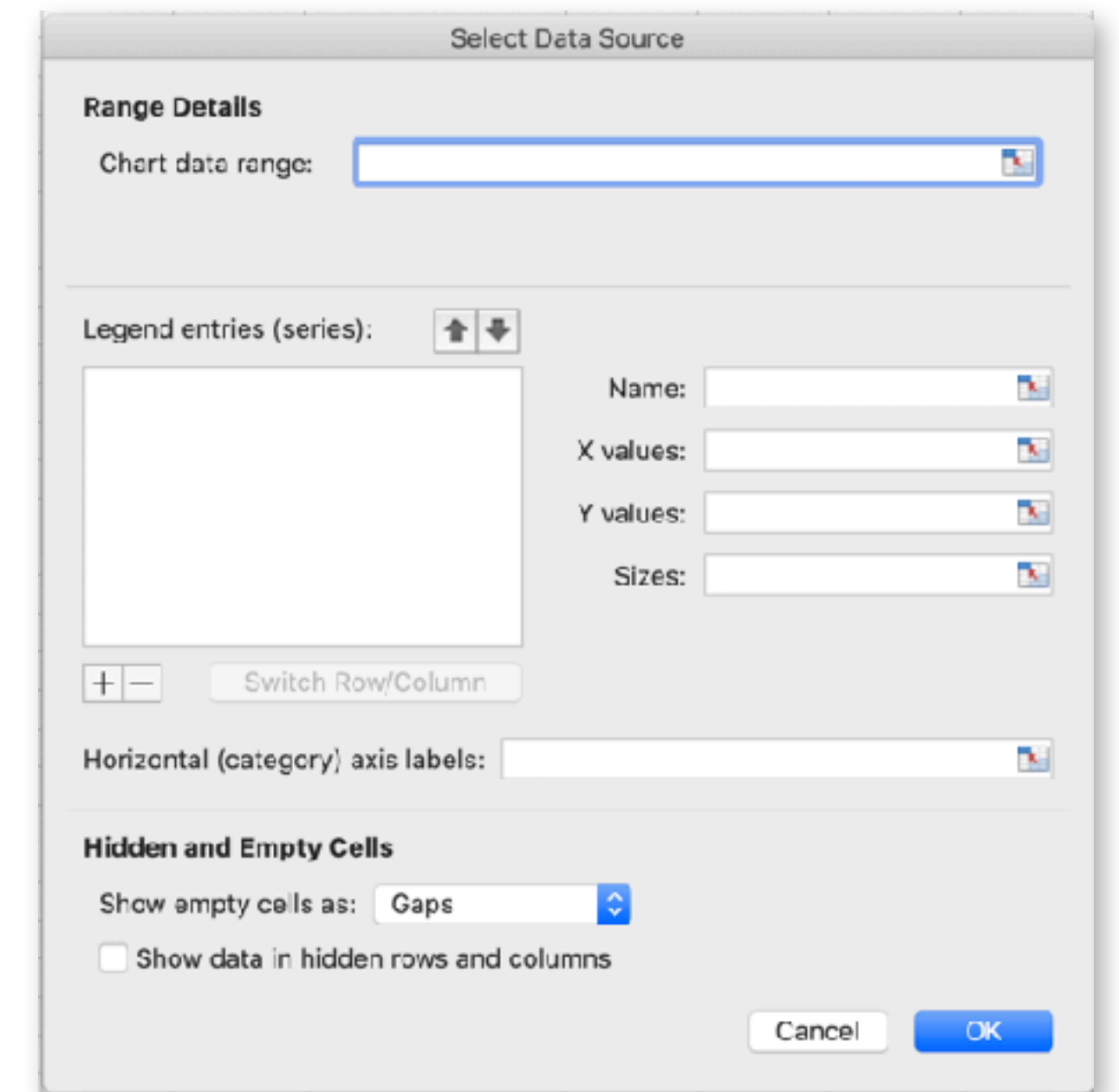
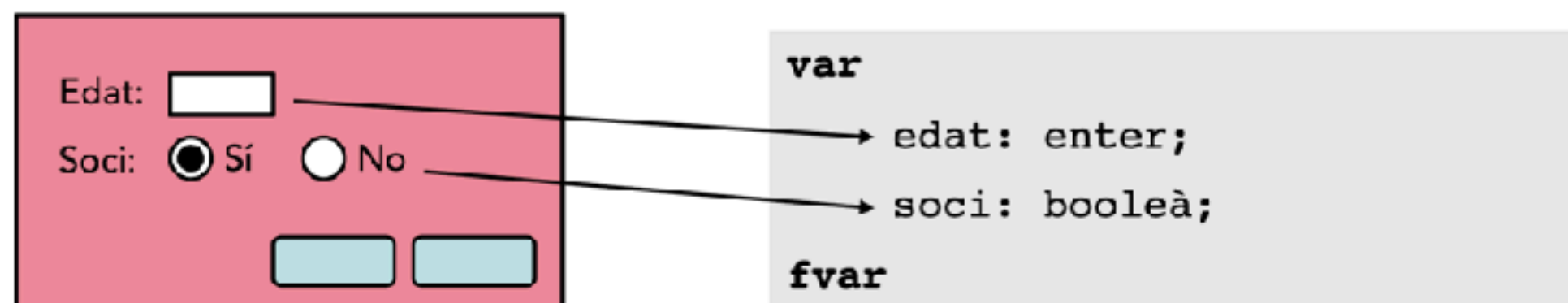


Interacció amb l'usuari

Entrada i sortida de teclat i pantalla

- Sovint trobem programes que necessiten la interacció de l'usuari:
 - Introduir dades
 - Emplenar formularis
 - Escollir opcions d'un menú
 - Prémer botons per processar la informació, sortir del programa, etc.
- La majoria de programes actuals que han d'interactuar amb l'usuari presenten una interfície gràfica d'usuari (**GUI**, Graphic User Interface))

- Una interfície gràfica d'usuari conté elements per a què l'usuari introdueixi dades o bé interaccioni amb el programa:
 - Les accions sobre els botons governen el flux d'execució del programa, activen processos, etc.
 - Els elements de formulari permeten emplenar els valors de les variables que farà servir el programa.

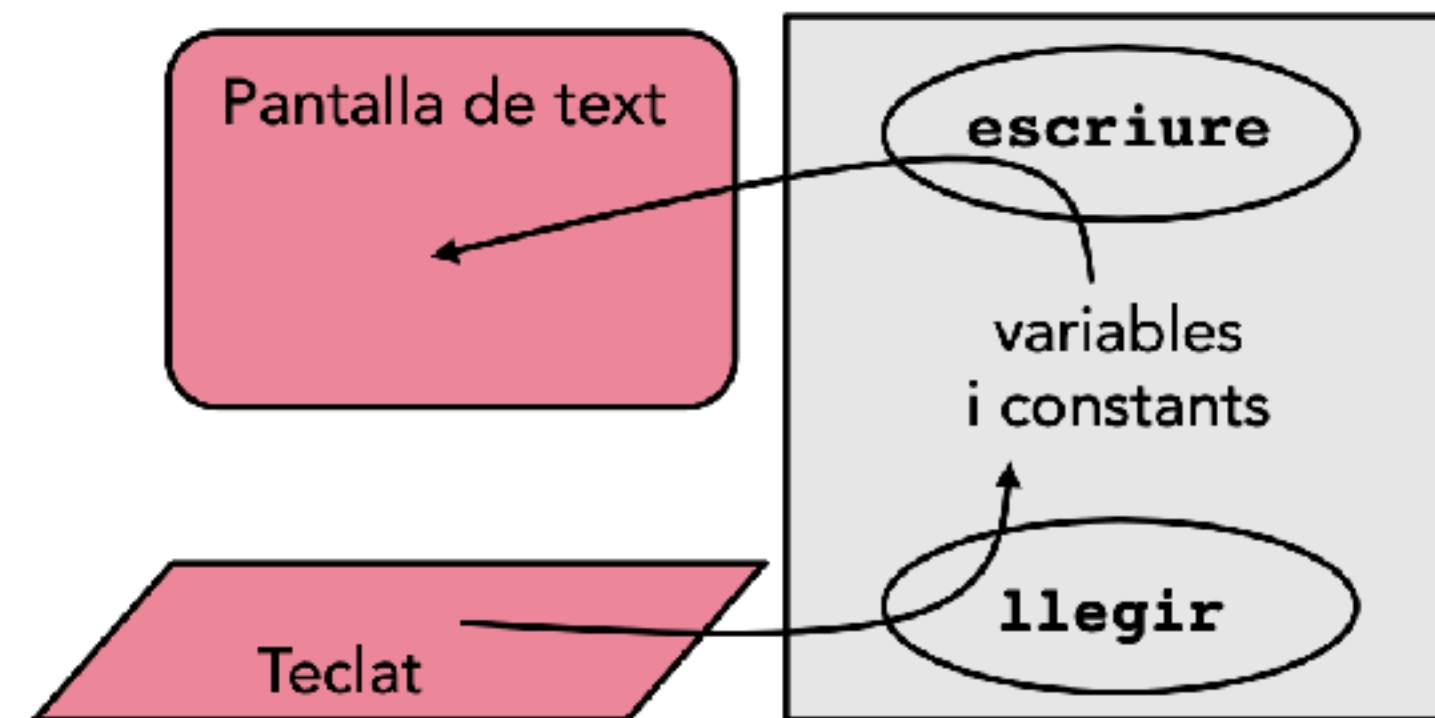


Menú d'inserir gràfics a MS. Excel

Interacció amb l'usuari

Entrada i sortida de teclat i pantalla

- Nosaltres ens centrarem en crear programes sense **GUI**
- Pels nostres propòsits (programació científica) programar una GUI consumeix molt de temps i no aporta cap benefici, al contrari, a nosaltres ens interessen tasques automatitzables.
- Com interactuem? A través del terminal:
 - La introducció de dades per part de l'usuari es fa a través del teclat.
 - La sortida d'informació es fa per mitjà de línies de text.



Interacció amb l'usuari

Entrada i sortida de teclat i pantalla

- En pseudocodi:
- Per interactuar amb el terminal en pseudocodi utilitzem els procediments **llegir** i **escriure**

escriure("text", variable, "text", variable, ...)

- Mostra un text per pantalla, permetent concatenar diferents textos i variables. Quan acaba d'escriure fa un salt de línia

```
...  
  a := 100;  
  escriure("Hola, què tal?");  
  escriure("La variable a val", a);  
...
```

```
$ Sortida  
$ Hola, què tal?  
$ La variable a val 100
```

llegir(nom_variable)

- Llegeix un enter, real, o caràcter per teclat. Aquest valor s'emmagatzema a la variable que s'indica per paràmetre. Quan s'acaba d'entrar la dada, cal que l'usuari premi <Enter>

```
escriure ("Quina opció tries?");  
escriure ("[1] Continuar");  
escriure ("[2] Abandonar");  
llegir (op);  
escriure ("Has escollit ", op);
```

Interacció amb l'usuari

Exemple

- Escriure un programa que demani a l'usuari dos nombres i en calculi la suma

```
algorisme suma és
var
    a, b, c: enter;
fvar
inici
    escriure ("Introdueix el primer operand:");
    llegir (a);                $ Obtenim primera dada
    escriure ("Introdueix el segon operand:");
    llegir (b);                $ Obtenim segona dada
    c := a + b;                $ Realitzem càlcul
    escriure ("La suma val: ",c); $ Mostrem resultat
falgorisme
```


OPERACIONS

Nombres

Operacions amb nombres

- Els operadors aritmètics permeten fer operacions amb valors numèrics (constants i/o variables)
- Usarem operadors i parèntesis per crear expressions aritmètiques
- En l'avaluació d'expressions (quan hi ha més d'un operador), la prioritat dels operadors en determina el resultat (com a matemàtiques)

Quin és el resultat de calcular $2 * 4 + 5$?

Sempre podem utilitzar parèntesis per forçar l'ordre

$$2 * (4 + 5)$$

Nombres

Operacions amb nombres

Op.	Significat	Exemple
+	La suma	<code>a := b + 2 + c;</code>
-	Resta, o canvi de signe	<code>a := b - c; d := -a;</code>
*	Multiplicació	<code>a := n * 2;</code>
div	Divisió entera, de dos enters	<code>b := 5;</code> <code>c := b div 2; \$ c valdrà 2</code>
mod	Residu, de dividir dos enters	<code>b := 5;</code> <code>c := b mod 2; \$ c valdrà 1</code>
/	Divisió amb decimals. Cal desar resultat en un real, sinó perdem els decimals.	<code>r := 5</code> <code>f := r / 2,0; \$ f valdrà 2,5</code>

Operacions com ara la potència o l'arrel quadrada es fan per mitjà de procediments que encapsulen els algorismes que fan els càlculs.

Nombres

Operació de divisió

- Les especificitats de la divisió entera o amb decimals depenen de cada llenguatge de programació
- Cal comprendre com funciona la divisió perquè, si no es fa bé, pot portar a què els programes no funcionin adequadament (i no ens n'adonem fins que sigui massa tard!)
- El **mòdul** només es pot fer entre nombres **enters**
- **El divisor no pot ser zero.** És una situació que sovint caldrà comprovar quan fem una divisió



Un error molt comú en C:

```
#include <stdio.h>

int main(){

    int a = 5;
    int b = 2;
    float result;

    result = a / b;

    printf("El resultat és: %f\n", result);

    return 0;
}
```

Quin creieu que és el resultat?

El resultat és: 2.000000

Nombres

Operacions de comparació

- Els operadors relacionals permeten realitzar comparacions:

$=, \neq, <, >, \leq, \geq$

- El resultat d'una operació serà **cert** o **fals** en funció de si el resultat de la comparació es compleix o no.


```
var
    a: enter;
fvar
inici
    a := 2;
    $ a = 2 és cert, a ≠ 2 és fals
    $ a > 2 és fals
    $ a ≥ 2 és cert
```


Booleans i expressions lògiques


Expressions lògiques

- Amb els operadors de comparació, els operadors lògics i els parèntesis definim expressions lògiques.
- Operadors lògics:
 - Operador lògic **i** (**and**): És cert quan **tots** els membres de l'expressió són certs.
 - Operador lògic **o** (**or**): És cert quan **algun** dels membres de l'expressió és cert.
 - Operador lògic **no** (**not**): És cert quan l'expressió que modifica és fals, i fals quan l'expressió és certa.

Recordeu les taules de la veritat

NOT	
	
INPUT	OUTPUT
A	
0	1
1	0

AND		
		
INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR		
		
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

Booleans i expressions lògiques

Expressions lògiques

```
var
    x, y, z: enter;
    a, b, c, d: booleà;
fvar
...
x := 5;
y := -5;
a := x ≠ 5;
b := no (x ≠ 5);
c := b i y > 10;
d := b o y > 10;
```

Aquest tipus d'expressions les farem servir més endavant, per a les estructures condicionals i als bucles. És allà on tenen més utilitat.

Booleans i expressions lògiques

Expressions lògiques

- Cal anar en compte amb expressions que són vàlides matemàticament parlant, com:

$$2 < x < 7$$

La variable 'x' és més gran que 2 i menor que 7

- Que en pseudocodi (i en C) haurem d'especificar amb dues operacions binàries de dos operands

$$(x > 2) \text{ i } (x < 7)$$

Booleans i expressions lògiques

Àlgebra de Boole

– Distributiva

$$a \text{ i } (b \text{ o } c) = (a \text{ i } b) \text{ o } (a \text{ i } c)$$

$$a \text{ o } (b \text{ i } c) = (a \text{ o } b) \text{ i } (a \text{ o } c)$$

– Associativa

$$a \text{ i } b \text{ i } c = a \text{ i } (b \text{ i } c)$$

$$a \text{ o } b \text{ o } c = a \text{ o } (b \text{ o } c)$$

– Involució

$$\text{no } (\text{no } a) = a$$

– Element neutre

$$a \text{ i } \text{cert} = a$$

$$a \text{ o } \text{fals} = a$$

– Element complementari

$$a \text{ i } (\text{no } a) = \text{fals}$$

$$a \text{ o } (\text{no } a) = \text{cert}$$

– Idempotència

$$a \text{ i } a = a$$

$$a \text{ o } a = a$$

– Llei d'absorció

$$a \text{ i } (a \text{ o } b) = a$$

$$a \text{ o } (a \text{ i } b) = a$$

– Llei de Morgan

$$\text{no } (a \text{ i } b) = (\text{no } a) \text{ o } (\text{no } b)$$

$$\text{no } (a \text{ o } b) = (\text{no } a) \text{ i } (\text{no } b)$$

Booleans i expressions lògiques

Exercici

- Tenim dos punts en el pla: (x_1, y_1) i (x_2, y_2)
- Escriu l'expressió lògica que retornarà **cert** si els dos punts són iguals i **fals** en cas contrari


```
var
    x1, y1, x2, y2: enter;
    diferents: booleà;

fvar
    $ són iguals si x i y dels dos punts són iguals
diferents :=
```

Parèntesi

Prioritat dels operadors

- A l'hora d'avaluar expressions amb diversos operadors, es té en compte la seva **prioritat**.



Operador	Associativitat
()	d'esquerra a dreta
no	de dreta a esquerra
* / div mod	d'esquerra a dreta
+ -	d'esquerra a dreta
< ≤ > ≥	d'esquerra a dreta
= ≠	d'esquerra a dreta
i	d'esquerra a dreta
o	d'esquerra a dreta
:=	de dreta a esquerra

En cas de tenir dubtes, els parèntesis ens permetran indicar com s'ha d'avaluar l'expressió.

Operacions amb caràcters

Com que els caràcters són, en el fons, nombres, podem operar-hi

Què imprimeix aquest codi?

```
algorisme exercici
var
  lletra: caràcter;
fvar
inici
  lletra := 'd';
  lletra := lletra + 1;
  escriure("Lletra:", lletra);
falgorisme
```

Sortida:

Lletra: e

A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

Taula ASCII (només lletres)

Operacions amb caràcters

Com que els caràcters són, en el fons, nombres, podem operar-hi

Què fa aquest codi?

```
algorisme exercici
var
  lletra: caràcter;
fvar
inici
  escriure(“Introdueix una lletra minúscula”);
  llegir(lletra);
  lletra := lletra - ('a' - 'A');
  escriure(“Lletra:”, lletra);
falgorisme
```

La sortida serà la lletra que hagi introduït l'usuari però en majúscula (sempre i quant la lletra introduïda fos minúscula)
Què passa si l'usuari introdueix una lletra que no és minúscula? Com ho comprovem?

A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

Taula ASCII (només lletres)

Operacions amb caràcters

Com que els caràcters són, en el fons, nombres, podem operar-hi

Què pretén fer aquest codi i quin problema hi ha?

```
algorisme exercici
var
  a,b: caràcter;
fvar
inici
  escriure("Introdueix dos nombres");
  llegir(a);
  llegir(b);
  escriure("La suma és:", a+b);
falgorisme
```

Si l'usuari introdueix els nombres 4 i 5, quina sortida tindrem?

La suma és: i

0	48	A	65	a	97
1	49	B	66	b	98
2	50	C	67	c	99
3	51	D	68	d	100
4	52	E	69	e	101
5	53	F	70	f	102
6	54	G	71	g	103
7	55	H	72	h	104
8	56	I	73	i	105
9	57	J	74	j	106
:	58	K	75	k	107
;	59	L	76	l	108
<	60	M	77	m	109
=	61	N	78	n	110
>	62	O	79	o	111
?	63	P	80	p	112
		Q	81	q	113
		R	82	r	114
		S	83	s	115
		T	84	t	116
		U	85	u	117
		V	86	v	118
		W	87	w	119
		X	88	x	120
		Y	89	y	121
		Z	90	z	122

Taula ASCII (fragment)

A CONTINUACIÓ...

ESTRUCTURES DE CONTROL