

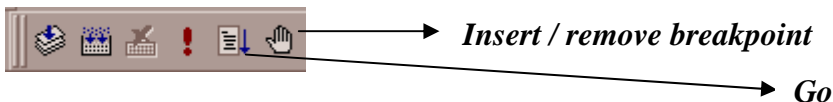
# ALGORISMES I PROGRAMACIÓ (GESTIÓ AERONÀUTICA)

## SESSIÓ 21/02/2005

### 1. Debugger

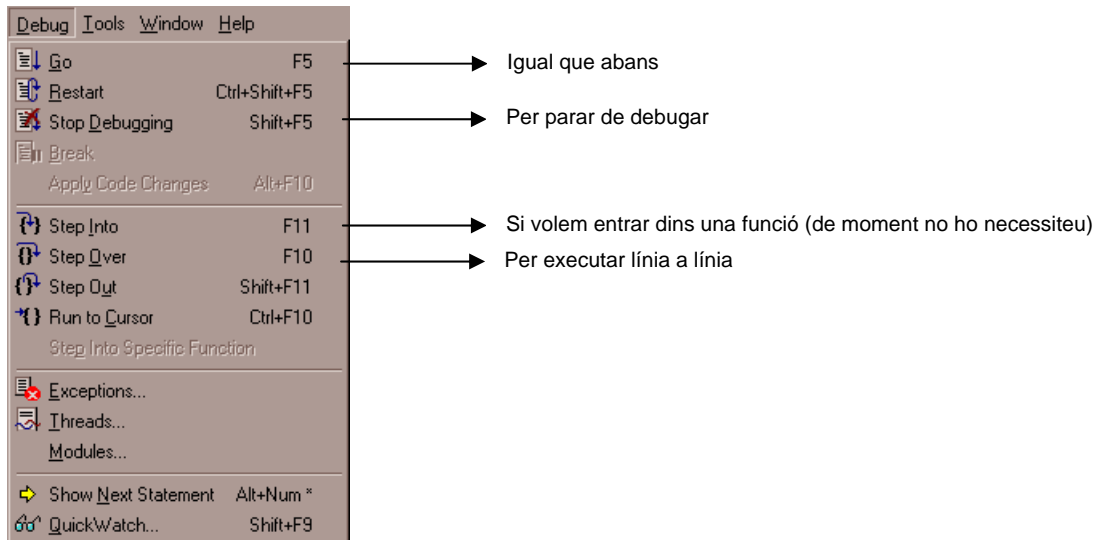
Quan els vostres programes creixin i us convingui controlar el valor que van prenent les variables i/o saber quina línia de codi s'executa en cada instant, necessitareu conèixer les eines del debugger. Anem a veure una petita introducció.

Si premem la tecla F10 (*step over*), començarem a executar el programa línia a línia. Si volem anar directament a una línia en concret, el que hauré de fer-li és marcar-li un *breakpoint*. Això es fa posant el cursor sobre aquesta línia i prement la mà que es mostra a la figura següent:

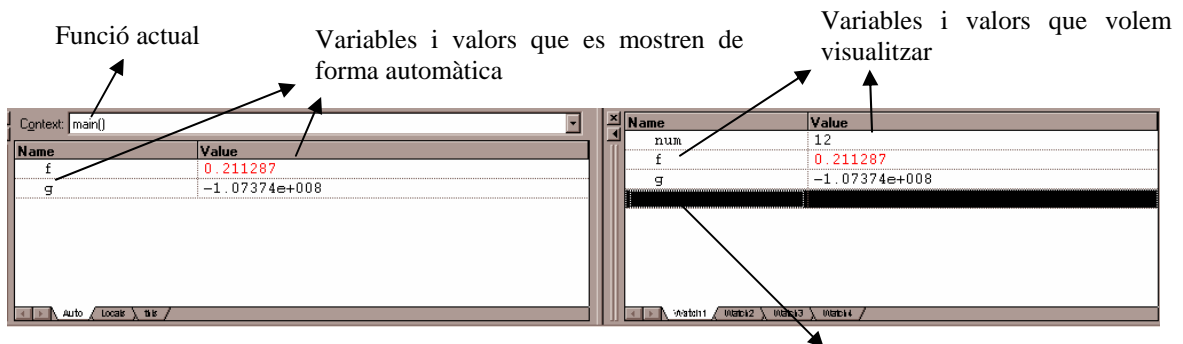


Una vegada tenim el breakpoint posat, per anar-hi, premem el botó de Go. A partir d'aquí continuem executant línia a línia amb F10.

Una vegada hem començat a debugar, al menú ens apareix una nova opció: Debug. Si la despleguem veurem:



A la part inferior us apareixeran uns quadres que us permetran de controlar quin és el valor d'una variable en tot moment:



Podem anar introduint quines variables volem veure clickant aquí i escrivint el nom de la variable

## 2. Definició de constants

Podem trobar-nos en el cas que, al llarg del programa, necessitem fer referència a algun valor especial que roman fixe, és a dir, una constant. La manera de declarar constants en C és la següent. Abans del *main* escrivim

```
#define nom_constant valor_de_la_constant
```

Els següents exemples ho il·lustren:

```
#define Pi 3.141592
#define E 2.7183
```

En el primer exemple, quan el compilador de C trobi *Pi*, automàticament ho substituirà pel valor 3.141592.

**Exemple>** Observeu el següent codi:

```
#define UnaConstant 45.3

void main(void)
{
    // Declaració de variables
    char c;
    int var1,var2;
    float var3;

    // Inicialització variables
    c='a';
    var1=22;
    var2=var1+10;
    var3=UnaConstant*var2;

    printf("\nNo us mostro el resultat porque el mirareu tot debugant\n");
}
```

**Exercici #1>** Debugueu el programa pas a pas (prement la tecla F10) observant el valor que van agafant les variables.

Una cosa que heu de tenir molt en compte quan feu operacions numèriques és els formats dels nombres i variables que hi intervenen:

- Si feu una operació que dona com a resultat un nombre real, la variable on l'emmagatzemeu ha de ser real, sinó els decimals es perdran. Feu la prova de canviar, a l'exemple anterior, el tipus de la variable *var3* de *float* a *int* i debugueu. Veieu?
- Si feu una operació que dona com a resultat un nombre enter i el guardeu dins una variable real, el què passarà és que al final del nombre s'hi afegirà .0 . Per exemple, si guardem 22 dins una variable real, el què guardem és, de fet, 22.00 . Feu ara la prova d'assignar a la variable *var3* (de tipus *float*) el contingut de la variable *var1*.

**Exercici #2>** Feu un programa en què definiu la constant *Pi* i declareu un radi enter *r* que inicialitzareu a 4. En dues variables *p* i *a*, del tipus apropiat (per no perdre decimals!!), hi guardareu el perímetre i l'àrea d'un cercle de radi *r*. Mostreu el resultat per pantalla.

### 3. Operacions aritmètiques

Operen sobre dades numèriques (floats i enters). Tenim els següents:

Símbol	Significat	Exemple
+	Suma	$r = 3 + 5$ ( valdrà 8)
-	Resta	$r = 5 - 3$ ( valdrà 2)
*	Multiplicació	$r = 5 * 3$ ( valdrà 15)
/	Divisió	$r = 10 / 2$ ( valdrà 5)
%	Mòdul	$r = 3 \% 2$ (valdrà 1)

Taula 1. Operadors aritmètics

Totes les operacions poden realitzar-se tant sobre floats com sobre enters, excepte l'operació Mòdul (residu de la divisió entera) que només pot realitzar-se entre dos enters.

El tipus del resultat d'una operació depèn dels tipus dels números (arguments) que intervinguin. Si algun dels arguments és un float, l'operació (i per tant el resultat) serà un float (tindrà decimals).

**Exercici #3>** Digueu el tipus de dades (int o float) de les següents operacions:

- a)  $5 / 2$                       b)  $5.0 / 2$                       c)  $5 / 2.0 + 1$                       d)  $5 / 2 + 1.0$   
e)  $2.0 * (2 - 1)$                       f)  $1.0 + 12 (1 * 2) - 1$                       g)  $4 + 11 / (2.0 * (2 - 1))$

**Exercici #4>** Feu un programa que comprovi que el resultat que havíeu previst en les operacions f) i g) de l'exercici anterior és correcte.

Per realitzar correctament les operacions aritmètiques s'han de tenir en compte l'ordre de prioritats entre operacions. A continuació teniu l'ordre de prioritats entre les operacions més comunes:

Prioritat	Operació
Primera	Parèntesis ()
Segona	Multiplicació, divisió, mòdul (*, /, %)
Tercera	Suma, resta (+, -)

Taula 2. Prioritats dels operadors aritmètics

Recordeu que els parèntesis tenen prioritats màxima i poden alterar l'ordre de prioritats de la taula anterior:

$$5 + 4 / 2 = 5 + 2 = 7$$

$$(5 + 4) / 2 = 9 / 2 = 4.5$$

**Exercici #5>** Realitzeu les següents operacions a mà i després amb l'ordinador per  $x=2$ :

a)  $\frac{2}{x-4}$                       b)  $\frac{x-3}{x+2}$                       c)  $\frac{1+4}{3*(1+3)}$

## 4. Operadors relacionals i sentències lògiques

La forma més comú de generar dades booleanes és mitjançant els operadors de relació i comparació ( >, >=, <, <=, ==, != ). Aquests operadors comparen el valor de dos dades o variables i retornen un valor diferent de zero (generalment un valor 1) per una condició VERDADERA (TRUE) i un valor ZERO per una condició FALSA (FALSE).

### Exemple>

- |           |         |        |         |
|-----------|---------|--------|---------|
| 1) 5 < 5  | → FALSE | 5 < 7  | → TRUE  |
| 2) 5 <= 5 | → TRUE  | 7 <= 5 | → FALSE |
| 3) 5 > 5  | → FALSE | 7 > 5  | → TRUE  |
| 4) 5 >= 4 | → TRUE  | 5 >= 7 | → FALSE |
| 5) 5 == 5 | → TRUE  | 5 == 7 | → FALSE |
| 6) 5 != 4 | → TRUE  | 5 != 5 | → FALSE |

Mitjançant els operadors lògics 'AND' (&&), 'OR' (||) i la negació (!) podem combinar variables lògiques per obtenir expressions o sentències lògiques més elaborades. Per calcular el resultat d'una operació lògica, només cal substituir FALSE per 0, TRUE per 1, el '&&' per '\*', el '||' per '+' i efectuar el càlcul aritmètic.

### Exemple >

- $(8 \leq 9) \&\& (10 < 1) \ || \ (2 \neq 0) = (\text{TRUE} \&\& \text{FALSE}) \ || \ \text{TRUE} == (1 * 0) + 1 = 1$  (TRUE)
- $(-2 >= 0) \&\& (6 >= 2) \&\& (5 == 9) = \text{TRUE} \&\& (\text{TRUE} \&\& \text{FALSE}) == 1 * (1 * 0) = 0$  (FALSE)

**Exercici #6>** Escriviu les següents sentències lògiques com en l'exemple anterior i calculeu el seu resultat.

- $(5 < 7) \&\& ((8 > 5) \ || \ (7 > 3))$
- $(6 == 'c') \&\& (0 < -5) \&\& ((9 >= 100) \ || \ (6 < 6))$
- $(9 >= 9) \&\& (13 < 11) \ || \ (-36 >= 90)$

**Exercici #7>** Feu un programa que comprovi que el resultat que heu calculat de les operacions anteriors és correcte.

**Exercici #8>** Donada la sentència lògica:

$((\text{numero} >= 1.0) \&\& (\text{numero} <= 10.0)) \ || \ ((\text{numero} >= 90.0) \&\& (\text{numero} <= 100.0))$

- Quin resultat obtindrem per: numero=11, numero=5, numero=105, numero=-1?
- Quin rang de valors podem donar a la variable 'numero' per a que el resultat de l'expressió sigui: 1 || 0, 1 || 1?

## 5. Operadors de Pre/Post Increment i Pre/Post Decrement

Aquests operadors (++) i (--) són operadors aritmètics que incrementen (++) o disminueixen (--) **en una unitat** el valor de la variable a la qual afecten. Aquests operadors poden anar

immediatament davant o darrera de la variable. Si precedeixen a la variable, aquesta és incrementada o decrementada abans de que el valor de la variable sigui utilitzat en la expressió en la que apareix. Si és la variable la que precedeix a l'operador, la variable és incrementada o decrementada després de ser utilitzada en la expressió. A continuació es presenta un exemple d'aquests operadors:

### Exemple>

```
i = 2;
j = 2;

m = i++; // després d'executar-se aquesta sentència m=2 i i=3,
         // per tant, equival a:
         //     m=i;
         //     i=i+1;

n = ++j; // després d'executar-se aquesta sentència n=3 i j=3,
         // per tant, equival a:
         //     j=j+1;
         //     n=j;
```

Com aquests operadors són molt utilitzats, és important entendre molt bé perquè els resultats  $m$  i  $n$  de l'exemple anterior són diferents.

**Exercici #9>** Digueu quin és el valor de la variable  $j$  si  $i=1$ :

- a)  $j = (i++) + 2$ ;      b)  $j = (i--) + 2$ ;      c)  $j = (++i) + 2$ ;      d)  $j = (--i) + 2$ ;

**Exercici #10>** Incorporeu les operacions de l'exercici anterior a un petit programa i utilitzeu el Debugger per comprovar que la vostra predicció és correcta.

## 6. Altres exercicis

**Exercici #11>** Suposant que teniu les definicions següents:

```
int n = 10, i = 0 ;
float x = 5.0 ;
```

Quin és el resultat de les expressions i el valor de totes les variables després de les següents operacions?

- a)  $n = n * i++$ ;  
b)  $n += ++i$ ;  
c)  $x = (\text{int}) x / n$ ;  
d)  $(x * (n+i++)) > 50$   
e)  $((x*(n - ++i)) > (n / ++i)) \&\& (((\text{int}) x / n) > 0)$   
f)  $!(i--<0) \parallel ((n / (\text{int}) x) > 0)$

Escriviu els programes que us permetin comprovar el resultat i el valor de les variables després de cada expressió.

**Exercici #12>** Digueu si les següents expressions són certes o be són falses :

- a)  $(2 + 3) * 6 == 2 + 3 * 6$  ;  
b)  $(2 + 3) + 6 == 2 + 3 + 6$  ;

- c)  $(44 - 1) * (10 - 22) != 44 + (-1 * 10) - 22 ;$   
d)  $(4 * 54) + (2 * 3) != 4 * 54 + 2 * 3 ;$

**Exercici #13>** Feu un petit programa en el que l'usuari entri dos nombres enters i que imprimeixi per pantalla el valor 1 si són iguals o 0 si són diferents.

**Exercici #14>** La temperatura en graus centígrads (Celsius) es pot convertir al seu equivalent en graus Fahrenheit, utilitzant la fórmula següent:

$$f = \left(\frac{9}{5}\right)c + 32$$

Escriure un programa que demani la temperatura en graus centígrads i escrigui el seu equivalent en graus Fahrenheit.

**Exercici #15>** Escriure un programa que calculi l'àrea d'un triangle utilitzant la fórmula següent:

$$Area = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$$

on p és el semi-perímetre del triangle,  $p = (a+b+c)/2$ , i a, b i c són els tres costats del triangle que s'han de demanar a l'usuari.