

LLENGUATGES DE PROGRAMACIÓ

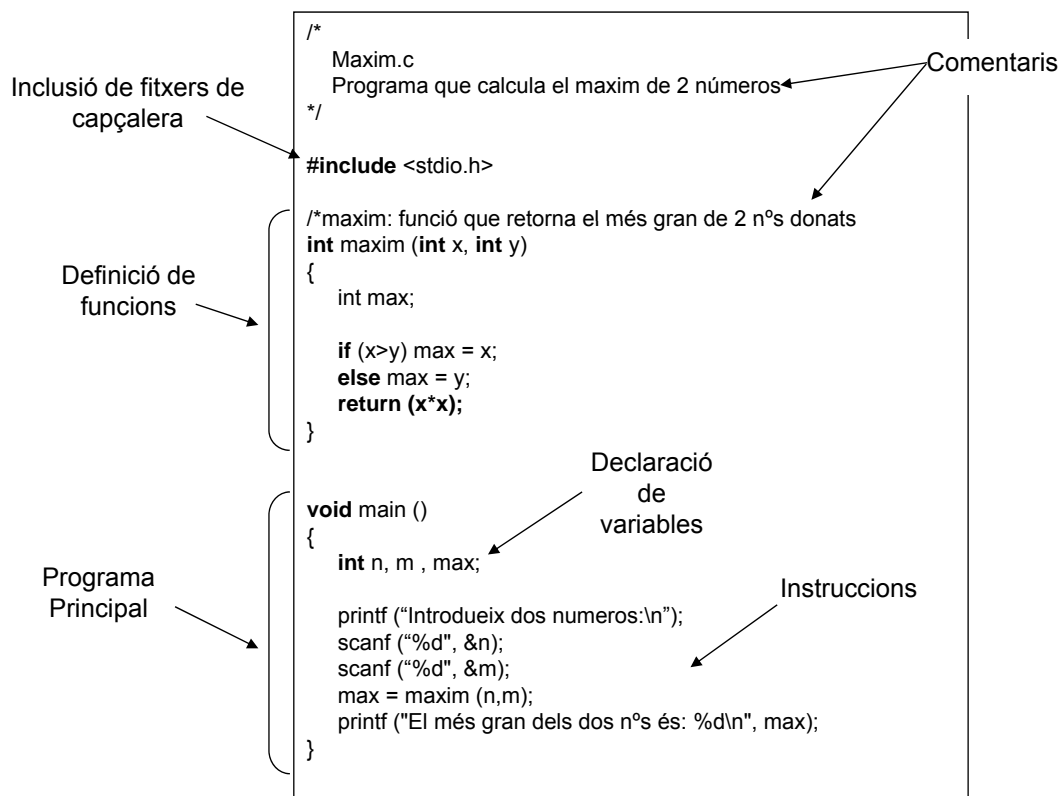
BLOC 1 – SEMINARI 1

INTRODUCCIÓ ALS PROGRAMES EN C

TIPUS DE DADES SIMPLES

ESTRUCTURES DE CONTROL CONDICIONALS

Elements bàsics d'un programa en C



Elements sintàctics d'un programa en C

Símbols especials

Símbol	Significat
#	Indicador de directives de pre-compilació (#define, #include, ...)
/* */	Delimitadors dels comentaris. Són ignorats pel compilador
;	Final de sentència. Totes les sentències han d'acabar en ;
{ }	Delimitadors d'un bloc d'instruccions que formen una unitat

Paraules reservades

auto	do	goto	signed	unsigned
break	double	if	sizeof	void
case	else	int	static	volatile
char	enum	long	struct	while
const	extern	register	switch	
continue	float	return	typedef	
default	for	short	union	

Estil d'escriptura d'un programa en C

Estil d'escriptura correcte

```
/*
Maxim.c
Programa que calcula el maxim de 2 números
*/

#include <stdio.h>

/*maxim: funció que retorna el més gran de 2 n°s donats
int maxim (int x, int y)
{
    int max;

    if (x>y)
        max = x;
    else
        max = y;
    return (x*x);
}

void main ()
{
    int n, m , max;

    printf ("Introdueix dos numeros:\n");
    scanf ("%d", &n);
    scanf ("%d", &m);
    max = maxim (n,m);
    printf ("El més gran dels dos n°s és: %d\n", max);
}
```

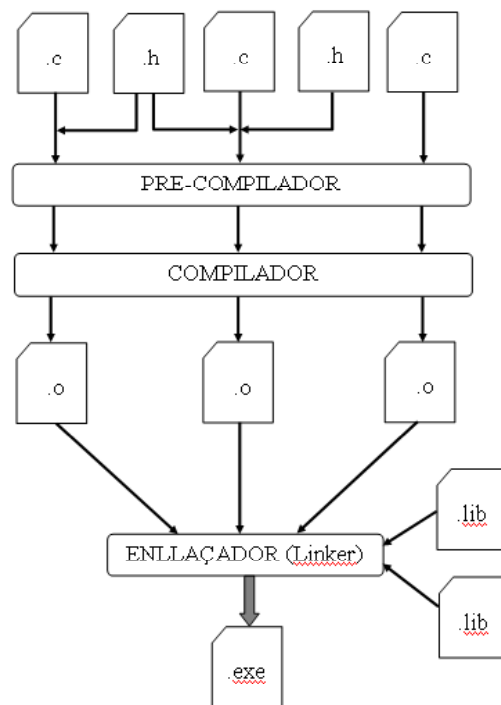
Estil d'escriptura incorrecte

```
#include <stdio.h>
    int Funcio (int x, int y){int a;
if (x>y) a = x; else a = y; return (x*x);}
void main ()
{int n, m , b;
printf ("Introdueix dos numeros:\n"); scanf ("%d", &n);
scanf ("%d", &m); b = Funcio (n,m);
printf ("El més gran dels dos n°s és: %d\n", b);
}
```

Regles d'escriptura

- Separar els blocs del programa
- Utilitzar comentaris
- Utilitzar les tabulacions
- Utilitzar noms de variables i funcions significatius

Compilació d'un programa en C



Declaració de variables

Sintaxi

```
<tipus_dades>    <nom_variable>;  
<tipus_dades>    <nom_variable>, <nom_variable>, ... ;
```

PSEUDOCODI

var n_alumnes: enter
num, den: real
lletra: caràcter
fivar

LLENGUATGE C

int n_alumnes;
float num,den;
char lletra;

Tipus de dades bàsics

Tipus pseudocodi	Tipus llenguatge C	Rang de valors
caràcter	char	-128 a 127
enter	int	-2.147.486.648 a 2.147.486.647
real	float	1.7E-38 a 3.4E+38
lògic	No existeix → int	Cert si ≠ 0, fals si = 0

Tipus 'char'

Codi	Car.	Codi	Car.	Codi	Car.	Codi	Car.	Codi	Car.	Codi	Car.	Codi	Car.
32		64	@	96	`	128	€	160		192	À	224	à
33	!	65	A	97	a	129	□	161	j	193	Á	225	á
34	"	66	B	98	b	130	,	162	g	194	Â	226	â
35	#	67	C	99	c	131	f	163	£	195	Ã	227	ã
36	\$	68	D	100	d	132	..	164	ç	196	Ä	228	ä
37	%	69	E	101	e	133	...	165	¥	197	Å	229	å
38	&	70	F	102	f	134	†	166	¡	198	Æ	230	æ
39	'	71	G	103	g	135	‡	167	§	199	Ç	231	ç
40	(72	H	104	h	136	ˆ	168	˘	200	È	232	è
41)	73	I	105	i	137	%o	169	©	201	É	233	é
42	*	74	J	106	j	138	§	170	ª	202	Ê	234	ê
43	+	75	K	107	k	139	<	171	«	203	Ë	235	ë
44	,	76	L	108	l	140	œ	172	¬	204	Ï	236	ï
45	-	77	M	109	m	141	□	173		205	Í	237	í
46	.	78	N	110	n	142	Ž	174	®	206	Î	238	î
47	/	79	O	111	o	143	□	175	ˉ	207	Ï	239	ï
48	0	80	P	112	p	144	□	176	°	208	Ð	240	ð
49	1	81	Q	113	q	145	'	177	±	209	Ñ	241	ñ
50	2	82	R	114	r	146	'	178	²	210	Ò	242	ò
51	3	83	S	115	s	147	"	179	³	211	Ó	243	ó
52	4	84	T	116	t	148	"	180	'	212	Ô	244	ô
53	5	85	U	117	u	149	•	181	µ	213	Õ	245	õ
54	6	86	V	118	v	150	-	182	¶	214	Ö	246	ö
55	7	87	W	119	w	151	—	183	·	215	×	247	×
56	8	88	X	120	x	152	˘	184	,	216	Ø	248	ø
57	9	89	Y	121	y	153	Ö	185	ˆ	217	Ù	249	ù
58	:	90	Z	122	z	154	š	186	°	218	Ú	250	ú
59	;	91	[123	{	155	>	187	»	219	Û	251	û
60	<	92	\	124		156	œ	188	¼	220	Ü	252	ü
61	=	93]	125	}	157	□	189	½	221	Ý	253	ý
62	>	94	^	126	~	158	ž	190	¾	222	Þ	254	þ
63	?	95	_	127	□	159	Ÿ	191	¿	223	ß	255	ÿ

Cada número correspon a una lletra, un dígit o un símbol reconegut per l'ordinador.

Al teclat:

ALT + N^o → caràcter de la taula

Exemple:

ALT + 169 → ®

IMPORTANT!!!

char lletra;

lletra='A';

lletra=65;

Són equivalents!!!

Inicialització de variables

Tenim dues possibilitats:

- En el moment de la declaració de la variable:

```
<tipus_dades> <nom_variable> = <expressió>;
```

Exemple:

```
void main()
{
    int num1=0;
    float num2=5.0;
    .....
}
```

- Després de la declaració de la variable:

```
<nom_variable> = <expressió>;
```

Exemple:

```
void main()
{
    int num1;
    .....
    num1=0;
}
```

Constants

Sintaxi

```
#define <nom_constant> <valor>
```

Directiva de pre-compilació.

En la precompilació es substitueix el nom de la constant pel seu valor allà on apareix.

Exemple

```
#include <stdio.h>

#define PI 3.141592 /* Defineix la constant PI */

main()
{
    float radi, perimetre, area;

    printf("Introdueix el radi del cercle: ");
    scanf("%f", &radi);
    perimetre = 2*PI*radi;
    area = PI*radi*radi;
    printf("Perimetre del cercle: %f\n", perimetre);
    printf("Area del cercle: %f\n", area);
}
```

Identificadors

Regles:

- Caràcters vàlids: lletres (a..z i A..Z), dígit (0-9) i el caràcter del subratllat (_).
- Qualsevol altre caràcter està prohibit.
- Primer caràcter: una lletra o el caràcter del subratllat. No pot ser un dígit.
- No es poden utilitzar les paraules reservades de C.
- Sensibilitat a majúscules i minúscules:

area_cercle <> Area_Cercle <> AREA_CERCLE

Exemples:

Identificadors vàlids	Identificadors invàlids
area_cercle	1x
_area_cercle	Area Cercle
AreaCercle_1	Area#Cercle
x1	x:
AquestIdentificadorEsValid	AquestNoEsVàlid

Entrada/Sortida bàsica

Entrada bàsica: scanf

```
scanf ("%identificador_del_tipus", &nom_de_la_variable);
```

Sortida bàsica: printf

```
printf ("missatge_de_text");  
printf ("%identificador_del_tipus", nom_de_la_variable);
```

Identificadors de tipus

Tipus	Enter	Real	Caràcter
Tipus C	int	float	char
Identificador tipus	%d	%f	%c

PSEUDOCODI

```
var num: enter fivar  
...  
llegir(num)  
...  
escriure ("El resultat es: ")  
escriure (num)  
...  
escriure ("El resultat es: ",num)
```

LLENGUATGE C

```
int num;  
...  
scanf("%d",&num);  
...  
printf ("El resultat es: ");  
printf ("%d",num);  
...  
printf("El resultat es: %d",num);
```

Operadors

Operadors aritmètics

Operació	Símbol	Tipus operands	Exemple
Suma	+	reals, enters	$5 + 3 \rightarrow 8$
Resta	-	reals, enters	$5 - 3 \rightarrow 8$
Multiplicació	*	reals, enters	$5 * 3 \rightarrow 15$
Divisió real	/	reals	$10.0 / 3.0 \rightarrow 3.333$
Divisió entera	/	enters	$10 / 3 \rightarrow 3$
Mòdul	%	enters	$10 \% 3 \rightarrow 1$

Operadors relacionals

<, <=, >, >=, ==, !=

Aplicables a operands de qualsevol tipus.

Si la comparació és certa, el resultat és un valor diferent de 0.

Si la comparació és falsa, el resultat és 0.

Exemples: Suposem que tenim definida $x=3$;

$x < 5 \rightarrow 1$ (CERT)	$x >= 3 \rightarrow 1$ (CERT)
$x <= 2 \rightarrow 0$ (FALS)	$x == 3 \rightarrow 1$ (CERT)
$x > 3 \rightarrow 0$ (FALS)	$x != 3 \rightarrow 0$ (FALS)

Operadors

Operadors lògics

Operació	Símbol	Exemple	Resultat
AND	&&	(5 < 3) && (10 < 20)	Fals (== 0)
OR		(5 < 3) (10 < 20)	Cert (!= 0)
Negació	!	!(5 < 3)	Cert (!= 0)

&&	Cert	Fals
Cert	Cert	Fals
Fals	Fals	Fals

	Cert	Fals
Cert	Cert	Cert
Fals	Cert	Fals

!	
Cert	Fals
Fals	Cert

Operadors

Operadors d'assignació compostos

Operador	Exemple	Significat
+=	x += y;	x = x + y;
-=	x -= y;	x = x - y;
*=	x *= y;	x = x * y;
/=	x /= y;	x = x / y;
%=	x %= y;	x = x % y;

Operadors d'increment i decrement

Operador	Significat
x++	Utilitza x. Després incrementa el seu valor (x = x + 1)
++x	Incrementa el valor de x (x = x + 1). Després utilitza el nou valor
x--	Utilitza x. Després decrementa el seu valor (x = x - 1)
--x	Decrementa el valor de x (x = x - 1). Després utilitza el nou valor

Prioritat dels operadors

Prioritat dels operadors

Prioritat	Operadors
1	Parèntesi ()
2	++, --, !, - (unari)
3	*, /, %
4	+, -
5	<, <=, >=, >
6	==, !=
7	&&
8	
9	=, *=, /=, %=, +=, -=

Exemples

int a=6, b=12, c=2, d=1;

$a + b / a * c - d \rightarrow 9$

$(a + b) / a * c - d \rightarrow 5$

$a + b / (a * (c - d)) \rightarrow 8$

$(a + b) / (a * c) - d \rightarrow 0$

$(a + b / a) * c - d \rightarrow 15$

Conversió de tipus

Rang dels tipus

char < int < long < float < double

Conversió implícita

- Expressions: conversió automàtica de tots els operands al tipus de rang més gran.
- Assignacions: conversió automàtica al tipus de la variable

Conversió explícita

- Força la conversió de l'expressió a un tipus determinat

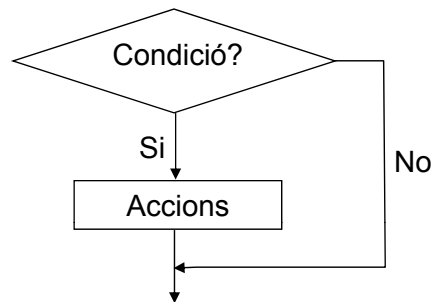
(<tipus>) <expressió>

Exemple: (int) x

```
int enter1 = 1000, enter2 = 20;
char caracter;
float real = 5.5;
```

```
real = enter1 + real;          /* Conversió implícita int-float sense pèrdua */
real = enter1 + enter2;       /* Conversió implícita int-float sense pèrdua */
real = (float) (enter1 + enter2); /* Conversió explícita int- float sense pèrdua */
caracter = enter2;           /* Conversió implícita amb possible pèrdua */
caracter = (char) enter1;    /* Conversió explícita amb possible pèrdua */
enter1 = real;               /* Conversió implícita amb possible pèrdua */
enter2 = (float) enter1 * 2.0; /* Conversió explícita sense pèrdua + implícita amb possible pèrdua */
```


Alternativa simple: if



PSEUDOCODI

```
si <condició> aleshores
  <accions>
fisi
```

LLENGUATGE C

```
if (condició)
  Acció_Si;
Instruccions_Comunes;
```

```
if (condició)
{
  Acció_Sí_1;
  ...
  Acció_Sí_N;
}
Instruccions_Comunes;
```

Alternativa simple: if

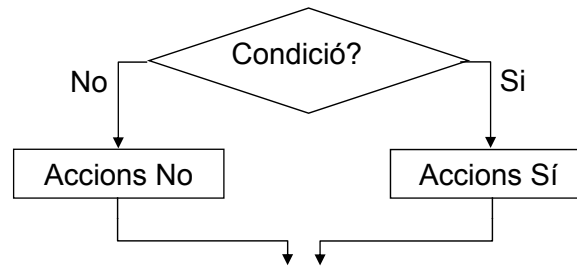
EXEMPLE

```
/* Programa que calcula àrea d'un cercle */
#include <stdio.h>
#define PI 3.141592 /* Definició de la constant PI */
main()
{
  float radi, area;

  printf ("Introdueix el radi del cercle:");
  scanf ("%f",&radi);

  if (radi > 0) /* Comprovació que el radi sigui positiu */
  {
    area = PI * radi * radi;
    printf ("L'area del cercle és: %f", area);
  }
}
```

Alternativa doble: if - else



PSEUDOCODI

```
si <condició> aleshores
    <accions sí>
sino
    <accions no>
fisi
```



LLENGUATGE C

```
if (condició)
{
    Acció_Sí_1;
    ...
    Acció_Sí_N;
}
else
{
    Acció_No_1;
    ...
    Acció_No_N;
}
Instruccions_Comunes;
```

Alternativa doble: if - else

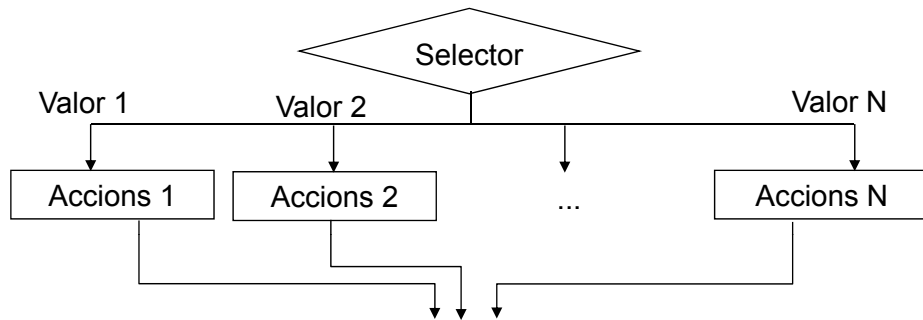
EXEMPLE

```
/* Programa que calcula àrea d'un cercle */
#include <stdio.h>
#define PI 3.141592 /* Definició de la constant PI */
main()
{
    float radi, area;


    printf ("Introdueix el radi del cercle:");
    scanf ("%f",&radi);

    if (radi > 0) /* Comprovació que el radi sigui positiu */
    {
        area = PI * radi * radi;
        printf ("L'area del cercle és: %f\n", area);
    }
    else /* Comprovació que el radi sigui positiu */
        printf ("El valor del radi ha de ser positiu.");
}
```

Alternativa múltiple: switch - case



PSEUDOCODI

```
cas <selector>
  <valor 1> : <accions 1>
  <valor 2> : <accions 2>
  
  <valor N> : <accions N>
  en altre cas : <accions resta casos>
fi cas
```

LLENGUATGE C

```
switch (expressió)
{
  case valor_1: Accions_1;
    break;
  case valor_2: Accions_2;
    break;
  ...
  case valor_N: Accions_N;
    break;
  default:
    Accions_resta_casos;
    break;
}
```



Alternativa múltiple: switch - case

```
/* Programa que simula el funcionament d'una calculadora simple */
#include <stdio.h>

main()
{
  char operacio;      /* Operació que s'ha de realitzar. */
  float x,y;         /* Valors d'entrada. */

  printf("Introdueix dos nombres:\n");
  scanf("%f %f",&x, &y);
  printf("Introdueix una operació vàlida: (+,-,*,/)\n");
  scanf("%c",&operacio);
  switch (operacio)
  {
    case '+':        printf ("%f", x+y);
                     break;
    case '-':        printf ("%f", x-y);
                     break;
    case '*':        printf ("%f", x*y);
                     break;
    case '/':        printf ("%f", x/y);
                     break;
    default:         printf("Operació incorrecta.\n");
                     break;
  }
}
```