



Fundamentos de Informática

4. Primeros Programas, Primeros Diseños

Fundamentos de Informática
Grado en Ingeniería Química

Contenidos

- 1 Empezando con C**
- 2 Variables**
- 3 Entrada de datos**
- 4 Operadores**
- 5 Condiciones**

3

Primeros diseños, primeros programas

1. Empezando con C

4

Primeros diseños, primeros programas

Empezando con C (estructura general)

- Un programa en C está formado por una o más funciones que se llaman entre si, más una cabecera al principio del archivo.
- **main()** es una función obligatoria en todo programa C, y es la primera función que se ejecuta al comenzar el programa.
- Una función es un grupo de instrucciones que realizan una o más acciones.
- Las funciones requieren los paréntesis porque pueden llevar cosas (parámetros) dentro de ellos.
- Las llaves { } encierran el cuerpo de las funciones
- Las funciones ayudan a modularizar el programa, dividiéndolo en partes pequeñas.

```
cabecera...  
funcion1( ... )  
{  
  ...  
}  
funcion2( ... )  
{  
  ...  
}  
...  
main()  
{  
  ...  
}
```

5

Primeros diseños, primeros programas

Empezando con C (estructura general)

De momento construiremos programas escribiendo instrucciones solo en la función **main()**, pero más adelante crearemos otras funciones.

cabecera...

```
main()
{
  ...
}
```

La palabra **main**, es una palabra reservada.
C tiene otras palabras reservadas, que usa para fines específicos (ej: *float*, *while*, *if*...)

6

Primeros diseños, primeros programas

Empezando con C (include)

```
/* mi primer programa */

#include <stdio.h>

main()
{
  printf("Hola Mundo\n");
  getchar();
}
```

- **#include** es una *directiva del preprocesador* que incluye archivos de cabecera que contienen funciones predefinidas

- **stdio.h** es la "standar input output", librería estandar de entrada y salida, y contiene las funciones **printf**, **getchar**, y otras muchas.

- < > indica que es un archivo de la biblioteca estándar.

- **#include "hola.txt"** leería hola.txt de la carpeta en la que se está trabajando (actual).

- También **#include "c:/data/hola.txt"**

Primeros diseños, primeros programas

Empezando con C (comentarios)

```
/* mi primer programa */
```

```
#include <stdio.h>
```

```
main()
```

```
{  
    printf("Hola Mundo\n");  
    getchar();  
}
```

- **/* mi primer programa */** es un comentario.
- Es información útil para el programador.
- El compilador ignora los comentarios.
- Los comentarios se pueden poner *casi* en cualquier sitio.

```
/* mi primer programa */  
#include <stdio.h>  
/* empiezo */  
main()  
{  
    /* voy a escribir en pantalla */  
    printf("Hola Mundo\n");  
    getchar();  
}
```

Primeros diseños, primeros programas

Empezando con C (comentarios)

- Se pueden poner comentarios con **//**
- Significa que toda la línea es un comentario

Es importante poner comentarios en los programas

```
// mi primer programa
```

```
/* este comentario  
es correcto */
```

```
// y este comentario  
no lo es
```

```
/* este tambien */  
/* es un comentario */  
/* usual */
```

```
/*  
y este tambien  
es un comentario  
usual  
*/
```

```
// A algunos programadores  
// les gusta poner asi  
// los comentarios
```

9

Primeros diseños, primeros programas

Empezando con C (printf)

```

/* mi primer programa */

#include <stdio.h>

main()
{
    printf("Hola Mundo\n");
    getchar();
}

```

- **printf()** es una función predefinida de C, que está en `stdio.h`
- **printf()** produce una salida en el dispositivo de salida estándar (normalmente la pantalla).

Hola Mundo

- **\n** es el carácter de control "nueva línea", que indica al sistema que termine la línea (lo siguiente que se escriba aparecerá en otra línea)

10

Primeros diseños, primeros programas

Empezando con C (getchar)

```

/* mi primer programa */

#include <stdio.h>

main()
{
    printf("Hola Mundo\n");
    getchar();
}

```

- **getchar()** es otra función de e/s que lee un carácter pulsado en el teclado. En este caso se usa para esperar a que se pulse una tecla antes de acabar el programa.

- Todas las sentencias C acaban en ;
- Se pueden poner varias sentencias en la misma línea

```

main()
{
    printf("Hola Mundo\n"); getchar();
    printf("¿Qué tal "); printf("os va?\n"); getchar();
}

```

Primeros diseños, primeros programas

Empezando con C (define y printf)

```
#include <stdio.h>
#define COLOR 13
#define PI 3.1416
main()
{
    printf("Constante %d. Pi %f \n", COLOR,PI );
    getchar();
}
```

- La directiva **#define** sustituye cada vez que aparece en el programa el nombre por el valor
- Es una operación del preprocesador
- Se usa para definir *constantes*
- Las constantes en C se suelen escribir en mayúsculas

printf()

- Su primer *argumento* es el texto a sacar, entre comillas
- Dentro del primer argumento, se indica con % el formato de los datos que se quieran sacar (Ej: %d para un valor entero, o %f para un real)
- El resto de argumentos son los valores a sacar
- Los argumentos se separan por comas

Primeros diseños, primeros programas

Empezando con C (formateo de printf)

• %d	entero	:13
• %4d	al menos 4 espacios	: 13
• %f	float (real)	:3.1416
• %5.2f	en total 5 espacios, 2 decimales	: 3.14
• %5.3f	total 5 espacios, 3 decimales	:3.142
• %.3f	3 decimales	:3.142
• %e	notación científica	:-6.25e-23
• %c	un caracter	:a
• %s	una cadena de caracteres	:un bocadillo

```
printf("Voy %c comer %s rico\n", 'a' ,"un bocadillo");
```

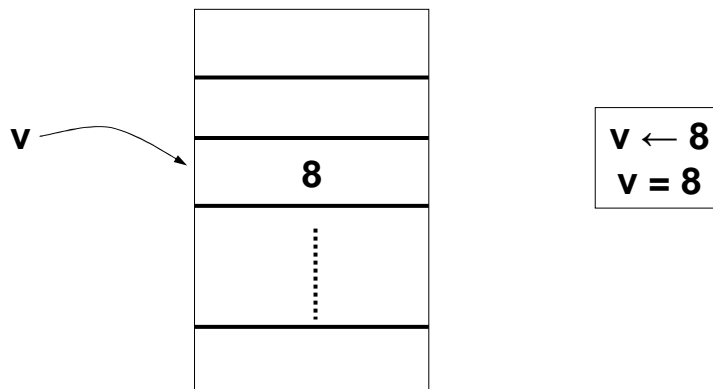
Primeros diseños, primeros programas

2. Variables

Primeros diseños, primeros programas

Variables (definición)

Variable: espacio de memoria reservado para almacenar un valor



Primeros diseños, primeros programas

Variables (características)

```

main()
{
    int v;

    v = 1;
    printf("El valor de v es %d \n",v);
    v = 2;
    printf("El valor de v es %d \n",v);
    v = v + 4;
    printf("El valor de v es %d \n",v);
    getchar();
}

```

C distingue mayúsculas / minúsculas

- **int v**; define **v** como una variable de tipo entero

- Es obligatorio declarar las variables
- Las variables se declaran al principio de la función

- Las variables en C se suelen escribir en minúsculas, salvo la primera letra.

- Empiezan por una letra, y pueden llevar letras, números y "_", hasta 32 caracteres.

contador, contadorSegundos
Contador2, contador_segundos

Primeros diseños, primeros programas

Variables (principales tipos de datos y orden de ejecución)

- **float** define variables de tipo real
- **char** de tipo carácter
- observar **c=getchar()**

Dentro de una función, las instrucciones se ejecutan por orden, de arriba a abajo

```

main()
{
    int v, pasos, contador;
    int casas, coches, motos;
    float media, ratio;
    float total;
    char c, salto;

    v=1;
    salto='z';
    pasos=v+3;
    contador=0;
    media=3.2;
    ratio = media * 2.0;
    c=getchar();
}

```


Primeros diseños, primeros programas

3. Entrada de datos

Primeros diseños, primeros programas

Entrada de datos (scanf)

```
#include <stdio.h>
main()
{
    int edad,a,b;
    float x,y;

    printf("introduce tu edad:");
    scanf("%d",&edad);
    printf("Dos reales separados por espacio:");
    scanf("%f %f",&x,&y);
    printf("Dos enteros separados por coma:");
    scanf("%d,%d",&a,&b);
    printf("%d,%f,%f,%d,%d\n",edad,x,y,a,b);
    getchar();
}
```

- **scanf()** es una función de **stdio.h** para leer datos del dispositivo estándar (normalmente el teclado).
- La variable a leer debe llevar delante el símbolo **&**
- El formato de **scanf()** es como el de **printf()**
- Normalmente se lee un dato por cada **scanf()**

Primeros diseños, primeros programas

4. Operadores

Primeros diseños, primeros programas

Operadores (de enteros)

```
main()
{
    int a,b,c;

    a=1;b=2;c=4;
    c=a+b;
    c=c+3;
    a=(b * c) - a;
    b=a/2;
    c=a%2;
}
```

+, **-**, *****, **/**, **%** son operadores para números enteros

/ calcula el cociente

% el resto (módulo)

Ojo a los paréntesis

21

Primeros diseños, primeros programas

Operadores (de reales)

```
main()
{
    float a,b,c;

    a=1.2;b=2.0;c=4.53;
    c=a+b;
    c=c+3.15;
    a=(b * c) - a;
    b=a/2;
}
```

+, -, * /son operadores para números reales

Ojo a los paréntesis

/ se pueden hacer conversiones entre int y float */*

int a,b; float x,y;

a=53; x=32.56;
b=x; y=a;

22

Primeros diseños, primeros programas

No olvidemos ...

- Un programa C tiene **palabras reservadas, funciones**, constantes, variables, operadores y algunos otros elementos (separadores, cadenas, formateo, etc.)

- Un programa se ejecuta linealmente, empezando en la primera instrucción de *main()*.

- Cada función tiene sus propios parámetros.

```
#include <stdio.h>
#define CO 7
```

```
main()
{
    int casas,b,c;
    float media;
    char car = 'y';
```

```
casas = 10;
printf("Total: %d y %d \n",casas, CO);
media = (casas + CO) / 2.0;
printf(" %c su media es %f",car,media);
getchar();
}
```

Primeros diseños, primeros programas

No olvidemos ...

- También :
 - Directivas del preprocesador
 - Una función *main* obligatoria
 - Declaraciones
 - Sentencias

```
#include <stdio.h>
#define CO 7

main()
{
    int casas,b,c;
    float media;
    char car;

    car = 'y';
    casas = 10;
    printf("Total: %d y %d \n",casas, CO);
    media = (casas + CO ) / 2.0;
    printf(" %c su media es %f",car,media);
    getchar();
}
```

Primeros diseños, primeros programas

Condiciones

Primeros diseños, primeros programas

Condiciones (if básico)

```

main() {
    int edad;

    printf("Tu edad"); scanf("%d",&edad);

    if (edad<18)
        printf("Eres menor. Ve acompañado\n");

    printf("Adelante\n");
}

```

Si se cumple la condición, se ejecuta la instrucción.

if es una palabra reservada.

Si hay que ejecutar varias instrucciones, se necesita un bloque entre { }

```

if (edad<18)
{
    printf("Eres menor. Ve acompañado\n");
    a=b+2;
}
printf("Adelante\n");

```

Primeros diseños, primeros programas

Condiciones (if else)

```

main() {
    int edad;

    printf("Tu edad"); scanf("%d",&edad);

    if (edad<18)
        printf("Eres menor. No pasas\n");
    else
        printf("Eres mayor. Pasa\n");
}

```

Si se cumple la condición, se ejecuta la zona *if*.

Si no se cumple, se ejecuta la zona *else*.

else es una palabra reservada.

También con bloques

```

if (edad<18)
    {b=2;a=b+2;}
else
{
    b=1;a=b+2;
    printf("%d",b);
}

```

Primeros diseños, primeros programas

Condiciones (estructura)

Repasemos

```
if (condición)
    sentencia;
```

```
if (condición)
    sentencia;
else
    sentencia;
```

“sentencia;” puede ser una sola instrucción, o un bloque de instrucciones entre llaves

```
{
    sentencia;
    sentencia;
    ...
}
```

Primeros diseños, primeros programas

Condiciones (operadores relacionales)

```
if (edad < 18) printf("Menor de 18\n");
if (edad <= 18) printf("Menor o igual de 18\n");
if (edad > 18) printf("Mayor de 18\n");
if (edad >= 18) printf("Mayor o igual de 18\n");
if (edad == 18) printf("Igual a 18\n");
if (edad != 18) printf("distinto de 18\n");
```

La condición puede contener expresiones

```
if ( (b+2)*3 != c/50 ) h=0;
```

Primeros diseños, primeros programas

Condiciones (if anidados)

Los **if** se pueden anidar:

```
if (condición)
    sentencia;
else if (condición2)
    sentencia;
else if (condición3)
    sentencia;
else
    sentencia;
```

Los **if** anidados se usan para decisiones multialternativas

Primeros diseños, primeros programas

Condiciones (if anidados)

También se puede:

```
if (condición)
    if (condición2)
        sentencia;
    else
        sentencia;
else
    sentencia;
```

```
if (a>0)
    if (b>0) a=a+1;
    else
        if (c>0)
            if (a<5) b=b-1;
            else
                if (b<5) c=c+1;
                else a=a-1;
        else
            if (c<5) b=b-1;
            else c=c-1;
else
    a=0;
```

Ejemplo

Primeros diseños, primeros programas

Condiciones

<pre> if (a>0) -> if (b>0) a=a+1; -> else -> if (c>0) -> if (a<5) b=b-1; -> else -> if (b<5) c=c+1; -> else a=a-1; -> else -> if (c<5) b=b-1; -> else c=c-1; else a=0; </pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">Ejemplo</div>
---	---

Esto es lo mismo. El ordenador lo entiende; nosotros no.

```

if (a>0) if (b>0) a=a+1; else if (c>0) if (a<5) b=b-1; else if (b<5)
c=c+1; else a=a-1; else if (c<5) b=b-1; else c=c-1; else
a=0;

```

Primeros diseños, primeros programas
