

Práctica 2. Tipos de datos, variables y constantes

Objetivos

- Familiarizarse con la estructura básica de un programa
- Empezar a aprender a utilizar entrada y salida estándar
 - `fprintf` para mostrar resultados (por ahora, por pantalla)
 - `fscanf` para leer datos (por ahora, de teclado)
- Familiarizarse con la declaración de variables locales
- Familiarizarse con el uso de constantes (directiva `#define`)
- Utilizar tipos de datos básicos (`int`, `char`, `float`,...)

Estructura básica de un programa en c

```
#include ... /* inclusión de bibliotecas */
#define ... /* definición de constantes */

funciones /* definición de funciones */
```

La función `main`

- Cuando se ejecuta un programa, se llama a la función `main`
- Sin `main` el compilador **no genera un ejecutable** (aunque puedes compilar tus propias **bibliotecas de funciones** para incluirlas en otros programas, en ese caso no hace falta la función `main`)
- Cuando la función `main` finaliza, el programa termina

Estructura básica de un programa en c

```
#include ... /* inclusión de bibliotecas */
#define ... /* definición de constantes */

/* aquí podremos añadir funciones auxiliares */

int main() { /* pero main siempre tiene que estar */
    tipodato variable=valor_inicial; /* variables locales */
    ... /* variables locales */
    instruccion1; /* instrucciones */
    instruccion2;

    ...
    getch(); /* normalmente, para dar
              tiempo a ver la salida */
    return 0;
}
```

Atención

- Las instrucciones terminan siempre con punto y coma ;
- Exceptuando `#include` y `#define`

Estructura básica de un programa en c

Copia este programa y Pruébalo:

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
#define PI 3.141592 /* definimos una constante */
int main() {
    float radio; /* variable local de tipo float */
    fprintf(stdout, "Dime un radio: "); /* mensaje de cortesía */
    fscanf (stdin, "%f", &radio); /* leemos un valor */
    fprintf(stdout, "El area del circulo de radio %f es %f\n",
            radio, PI*radio*radio);
    getch();
    return 0;
}
```

Ejercicios:

1. Modifica el programa para que también calcule el perímetro.
 2. Quita el símbolo & del `fscanf` y ejecuta de nuevo ¿qué ocurre?
-

Otro programa sencillo

```
#include <stdio.h>
#include <conio.h>
int main() {
    float precio_unidad;
    int numero_unidades;
    fprintf(stdout, "Dime el precio por unidad: ");
    fscanf (stdin, "%f", &precio_unidad);
    fprintf(stdout, "Dime el numero de unidades: ");
    fscanf (stdin, "%d", &numero_unidades);
    getch();
    return 0;
}
```

Ejercicios:

1. Completa el programa anterior para que muestre por pantalla el precio total.
2. Además, haz que muestre por pantalla el precio con IVA (puedes suponer que el precio por unidad no incluye el IVA y que éste es del 16%).

Rectángulo

Ejercicio:

Basándote en los programas anteriores, escribe un programa para el cálculo del área y el perímetro de un rectángulo.

- El programa debe preguntar al usuario el valor de la base y de la altura del rectángulo.
- A continuación debe mostrar por pantalla:
 - Los datos introducidos por el usuario.
 - El área del rectángulo.
 - El perímetro del rectángulo.

Distancia

Escribe un programa que pida al usuario por teclado las coordenadas (x,y) de dos puntos (x1,y1), (x2,y2) y que calcule la distancia que hay entre ellos. Como pista te ponemos un pequeño programa que tiene piezas útiles para tu programa:

```
#include <stdio.h>
#include <conio.h>
#include <math.h> /* para utilizar sqrt */

int main() {
    float x1, y1, raiz;
    fprintf(stdout, "Introduce x1 e y2 (separados por espacio): ");
    fscanf (stdin, "%f%f", &x1, &y1);
    raiz = sqrt(x1); /* sqrt significa square root */
    fprintf(stdout, "La raiz cuadrada de x1 es %f\n", raiz);
    getch();
    return 0;
}
```

Jugando con los caracteres

Copia este programa y pruébalo:

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
int main() {
    char c='A';
    fprintf(stdout,"c contiene el caracter \"%c\"\\n", c);
    c = c+1;
    fprintf(stdout,"Ahora c contiene \"%c\"\\n", c);
    c = c+1;
    fprintf(stdout,"Ahora c contiene \"%c\"\\n", c);
    getch();
    return 0;
}
```

Ejercicio:

Escribe un programa que pida un número entero y muestre el caracter de la tabla ASCII correspondiente por pantalla.

Lectura de caracteres

Copia este programa y pruébalo:

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
int main() {
    char c;
    fprintf(stdout,"Escribe un caracter:");
    fscanf (stdin, "%c",&c);
    fprintf(stdout,"Has escrito: %c\\n",c);
    getch();
    return 0;
}
```

Lectura de caracteres

Ahora prueba esto:

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
int main() {
    char c1, c2;
    fprintf(stdout, "Escribe un caracter:");
    fscanf (stdin, "%c", &c1);
    fprintf(stdout, "Escribe otro caracter:");
    fscanf (stdin, "%c", &c2);
    fprintf(stdout, "Has escrito: %c y luego %c\n", c1, c2);
    getch();
    return 0;
}
```

- ¿Qué ocurre?
 - Prueba a escribir 2 letras al contestar la primera pregunta
 - El problema es que al pulsar enter, el ordenador se guarda el caracter del enter ('\n')
-

Uso de constantes

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
#define mensaje "hola, soy una cadena de caracteres"
#define nombre "Perico"
#define formato "Hola %s!\n"
int main() {
    fprintf(stdout, mensaje);
    fprintf(stdout, "%s\n", mensaje);
    fprintf(stdout, "%s\n", formato);
    fprintf(stdout, nombre);
    fprintf(stdout, "nombre");
    fprintf(stdout, formato, nombre);
    getch();
    return 0;
}
```

- Comenta y descomenta (con /* */) algunos fprintf para ver lo que ocurre.
- ¿Por qué no funcionaría bien fprintf(stdout, formato); ?

Lectura de cadenas

```
#include <stdio.h> /* para usar fprintf */
#include <conio.h> /* para usar getch */
#define MAXLONG 80
int main() {
    char nombre[MAXLONG];
    fprintf(stdout, "Escribe tu nombre: ");
    fscanf (stdin, "%s", nombre); /* IMPORTANTE: %s NO LLEVA & */
    fprintf(stdout, "Hola %s!\n", nombre);
    getch();
    return 0;
}
```

Ejercicio:

Modifica el programa anterior para que además del nombre pregunte por el color favorito del usuario, a continuación el programa contestará como en este ejemplo:

```
Escribe tu nombre: Juan
Dime tu color favorito: azul
Hola Juan, mi color favorito tambien es el azul
```

La función sizeof

nos dice el número de bytes que ocupa un tipo de datos

```
#include <stdio.h>
#include <conio.h>
int main() {
    fprintf(stdout, "unsigned char ocupa %d bytes\n"
                  "char ocupa %d bytes\n"
                  "unsigned int ocupa %d bytes\n"
                  "int ocupa %d bytes\n",
            sizeof(unsigned char), sizeof(char),
            sizeof(unsigned int), sizeof(int));
    getch();
    return 0;
}
```

Ejercicio:

Ejecuta el programa anterior y anota lo que sale por pantalla.

