Práctica 1. Introducción al C

22-23 de Febrero de 2016

Objetivos de la práctica:

- Adquirir la capacidad de diseñar algoritmos para la resolución de problemas básicos.
- Conocer cómo se traduce un algoritmo escrito en pseudocódigo en un lenguaje de programación.

1. Introducción

En las prácticas sucesivas programaremos en un lenguaje de alto nivel como es el C. Necesitamos un compilador que nos traduzca los programas de lenguaje de alto nivel, como es el C, a lenguaje máquina (0s y 1s que es lo que entiende la máquina).

Las etapas a seguir son:

- En papel, diseñar el algoritmo que resuelve el problema que queremos resolver.
- Traducir el algoritmo a lenguaje C. Para ello usaremos un editor de textos como por ejemplo el *gedit*.
- Compilar el programa para obtener un ejecutable. Vamos a utilizar el compilador *gcc*. El comando a utilizar sería: `$gcc -o nombreprograma nombreprograma.c`
- Ejecutar el programa y probar que su funcionamiento es el deseado. Emplearemos el comando `$./nombreprograma`

2. Algoritmos

Un programa es la redacción en un lenguaje de programación de un algoritmo. El algoritmo resuelve un problema concreto, indicando qué pasos hay que seguir, qué datos se necesitan, qué datos obtiene, etc. Podemos definirlo como "Serie de pasos organizados que describen de forma detallada y precisa el proceso que se debe seguir para dar solución a un problema específico".

2.1. Algoritmo Imprime_Num

El algoritmo siguiente, imprime un mensaje por pantalla, lee un número e imprime por pantalla el número leído por teclado:

```
ALGORITMO Imprime_num
 ENTRADAS:
  Num: Entero ; Número que se lee
 SALIDAS:
 VARIABLES:
  Num: Entero
 INICIO
  ESCRIBA ‘‘Escribe un número: ’’
  LEA Num
  ESCRIBA ‘‘El número introducido es:’’
  ESCRIBA Num
  ESCRIBA ‘‘Fin del algoritmo’’
 FIN
```

2.2. Algoritmo Convierte

El algoritmo siguiente convierte una cantidad de kilómetros en su correspondiente cantidad de yardas, millas y pies:

```
ALGORITMO Convierte
 ENTRADAS:
  Km: Real ; Número de km
 SALIDAS:
  Millas: Real ;
  Pies: Real;
  Yardas: Real;
 VARIABLES:
  Km: Real ;
  Millas: Real ;
  Pies: Real;
  Yardas: Real;
 INICIO
  ESCRIBA ‘‘Escribe un número: ’’
  LEA Km
  Millas ← Km / 1.60 ;
  Pies ← Km * 100000 / 30.48 ;
  Yardas ← Km * 100000 / 91.44;
  ESCRIBA ‘‘Las millas son: ‘‘
  ESCRIBA Millas
  ESCRIBA ‘‘Las yardas son: ‘‘
  ESCRIBA Yardas
  ESCRIBA ‘‘Los pies son: ‘‘
  ESCRIBA Pies
  ESCRIBA ‘‘Fin del algoritmo’’
 FIN
```

3. Programación en lenguaje C

Los programas escritos en el lenguaje C se guardan en archivos con extensión `.c`. En esta práctica nuestros programas van a tener la siguiente estructura básica:

```
//Librería que contiene las funciones scanf y printf
#include <stdio.h>

//Función principal del programa
int main ()
{

  //Aquí van las instrucciones de nuestro programa.
  //Todas acaban en ;

  return 0;
}
```

3.1. Variables

Cuando se ejecuta un programa, tanto las instrucciones que componen ese programa como los datos (la información) que maneja, están en la memoria del computador.

- Una variable es un nombre que identifica una posición de memoria donde se va almacenar información.
- El tipo de datos de la variable especifica qué datos se pueden almacenar.
- Estos datos son los que manipula un programa cuando se está ejecutando.
- Una varible necesita un nombre (identificador) válido: sin espacios, sin acentos, etc.
- Antes de usar una variable hay que declararla. Se declara así: *tipo_ de_ dato identificador_ variable;*

Los tipos de datos que puede almacenar una variable son:

- char: para almacenar caracteres (letras). Ocupan 8 bits
- short, int y long: para almacenar números enteros.
- float y double: para almacenar números reales.

3.2. Entradas y Salidas en C

En C, para imprimir o leer datos por pantalla utilizamos la librería `stdio.h` que contiene las siguientes funciones:

- `printf`: sirve para imprimir datos.
- `scanf`: sirve para leer datos.

Las estudiaremos con más profundidad en las clases prácticas posteriores. Sin embargo, para realizar los ejercicios propuestos en esta práctica, las utilizaremos de la siguiente forma:

- Leer número entero: `scanf (‘‘%d’’, &variableEntera);`
- Leer número real: `scanf (‘‘%f’’, &variableReal);`
- Imprimir número entero: `printf (‘‘%d’’, variableEntera);`
- Imprimir número real: `printf (‘‘%f’’, variableReal);`
- Imprimir cadena: `printf (‘‘Cadena que queremos imprimir’’);`

3.3. Programa Imprime_Num

Teniendo en cuenta que hay que sustituir la función LEA y ESCRIBE por la correspondiente de las descritas en el apartado anterior, el programa en C quedaría:

```
//Librería que contiene las funciones scanf y printf
#include <stdio.h>

//Función principal del programa
int main ()
{
  // Este programa imprime por pantalla un número leído por teclado

  // Declaro las variables de mi función
  int Num;

  //Sustituyo la función ESCRIBA ‘‘cadena’’ por printf
  printf(‘‘Escribe un número: ’’);

  //Sustituyo la función LEA Num por scanf (‘‘%d’’, &variableEntera);
  scanf(‘‘%d’’, &Num); //Guarda el número leído en la variable Num


  //Sustituyo la función ESCRIBA ‘‘cadena’’ por printf
  printf(‘‘El número introducido es:’’);

  //Sustituyo la función ESCRIBA Num por printf(‘‘%d’’, variableEntera);
  printf(‘‘%d’’,Num);

  //Sustituyo la función ESCRIBA ‘‘cadena’’ por printf
  printf(‘‘Fin del algoritmo’’);

  //Fin del programa
  return 0;
}
```

3.4. Programa Convierte

El algoritmo convierte trabaja con números reales. Su traducción a lenguaje C sería:

```
//Librería que contiene las funciones scanf y printf
#include <stdio.h>

//Función principal del programa
int main ()
{
  // Este programa convierte los km leídos en millas, yardas y pies
  // mostrando el resultado por pantalla

  // Declaro las variables de mi función
  float Km, Millas, Yardas, Pies;

  //Sustituyo la función ESCRIBA ‘‘cadena’’ por printf
  printf(‘‘Escribe un número de kilómetros: ’’);

  //Sustituyo la función LEA Km por scanf (‘‘%f’’, &variableReal);
  scanf(‘‘%f’’, &Km); //Guarda el número leído en la variable Km

  //Convierto los km a millas:
  Millas = Km /1.60;

  //Convierto los km a pies:
  Pies = Km * 100000 / 30.48 ;

  //Convierto los km a yardas:
  Yardas = Km * 100000 / 91.44;

  //Imprimo los resultados
  printf(‘‘Las millas son: ‘‘);
  printf(‘‘%f’’, Millas);

  printf(‘‘\n Los pies son: ‘‘);
  printf(‘‘%f’’, Pies);

  printf(‘‘\nLas yardas son: ‘‘);
  printf(‘‘%f’’, Yardas);

  printf(‘‘\nFin del algoritmo’’);

  //Fin del programa
  return 0;
}
```

4. Ejercicios propuestos

4.1. Ejercicio 1

- Escribe en un archivo con extensión `.c` el programa `Imprime_Num`. Compílalo con el compilador `gcc`. Después, ejecútalo para comprobar su funcionamiento.
- Para que nos imprima la cadena `Fin del algoritmo` en la línea siguiente, le podemos añadir al principio de la cadena el carácter `\n` que hace referencia a un retorno de línea. Nos quedaría la instrucción `printf(‘‘\nFin del algoritmo’’);` Realiza ese cambio en tu código fuente (archivo `.c`) y vuelve a compilarlo y ejecutarlo para ver la diferencia
- Además, todos los `printf` que muestran los resultados se pueden agrupar en una única llamada a la función de la siguiente forma:

```
printf(‘‘El número introducido es: %d \nFin del algoritmo’’, Num);
```

- Realiza ese cambio en tu código fuente (archivo `.c`) y vuelve a compilarlo y ejecutarlo para ver la diferencia

4.2. Ejercicio 2

- Escribe en un archivo con extensión `.c` el programa `Convierte`. Compílalo con el compilador `gcc`. Después, ejecútalo para comprobar su funcionamiento.
- Sustituye los `printf` para mostrar resultados por la instrucción siguiente:

```
  printf(‘‘Las millas son: %f \nLas yardas son: %f \n Los pies son: %f
\nFin del algoritmo\n’’, Millas, Yardas, Pies);
```

- Comprueba que funciona compilando y ejecutando de nuevo el programa.

4.3. Ejercicio 3

Crea un programa que lea una temperatura en grados Celsius por teclado y lo pase a Farenheit imprimiendo el resultado por pantalla.

Para ello ten en cuenta que las operaciones son + (suma), - (resta), * (producto), / (división), % (resto de la división entera).

La instrucción:

a=b+1;

Guarda en la variable a el valor de la variable b más 1.