

Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia

# Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Pimentel
<i>Asignatura:</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	8
<i>Integrante(s):</i>	Uno Karin Natalia
<i>No. de Equipo de cómputo empleado:</i>	27
<i>No. de Lista o Brigada:</i>	8723 #50
<i>Semestre:</i>	1
<i>Fecha de entrega:</i>	7 de octubre del 2019
<i>Observaciones:</i>	

CALIFICACIÓN: \_\_\_\_\_

# PRÁCTICA 8.

## ESTRUCTURAS DE SELECCIÓN

### OBJETIVO

Desarrollar programas en lenguajes C con condiciones if, if - else, switch y ternaria para la solución de un problema.

### INTRODUCCIÓN

El lenguaje de programación en C, es un lenguaje conocido como de alto nivel. Una de las características del lenguaje de programación en C, es que es un lenguaje estructurado, lo que permite generar código claro y sencillo, ya que esta basado en la modularidad.

El lenguaje de programación en C, esta estructurado en tres partes fundamentales, las cuales son, una librería estándar, un programa compilador y un preprocesador.

Una variable en C es un espacio que reservamos en memoria para poder guardar información, las variables se utilizan a menudo para guardar números caracteres, entre otros tipos de datos, no solo en C si no en cualquier lenguaje de programación. En C cada variable (espacio reservado en memoria) puede utilizarse para guardar información, y dicha información se representa por medio de un tipo de dato específico, que determina como la interpretará el lenguaje en sí.

Recopilado en: <http://codigoprogramacion.com/cursos/tutoriales-c/variables-y->

En lenguaje C para ejecutar una acción se usa una instrucción como:

**1) Switch** (alternativa múltiple) permite seleccionar, por medio de una expresión, el siguiente bloque de instrucciones a ejecutar de entre varios posibles. En lenguaje C, para escribir una instrucción alternativa múltiple (switch) se utiliza la sintaxis:

```
switch ( <expresión> )
{
    case <expresión_1> : [ <bloque_de_instrucciones_1> ]
                        [ break; ]
    case <expresión_2> : [ <bloque_de_instrucciones_2> ]
                        [ break; ]
    ...
    case <expresión_n> : [ <bloque_de_instrucciones_n> ]
                        [ break; ]
    [ default : <bloque_de_instrucciones_n+1> ]
```

```
}
```

El resultado de evaluar la <expresión> debe ser un valor perteneciente a un tipo de dato finito y ordenado, es decir, entero, lógico, carácter, enumerado o subrango.

Dependiendo del valor obtenido al evaluar la <expresión>, se ejecutará un bloque de instrucciones u otro.

Opcionalmente, se puede escribir un <bloque\_de\_instrucciones\_n+1> después de default :. Este bloque de instrucciones se ejecutará en el caso de que el valor obtenido al evaluar la <expresión>, no se encuentre en ninguna de las expresiones después de cada case.

**2)** Otra, es la alternativa doble (if else) se utiliza la sintaxis:

```
if ( <expresión_lógica> )
{
    <bloque_de_instrucciones_1>
}
else
{
    <bloque_de_instrucciones_2>
}
```

A la <expresión\_lógica> de una instrucción alternativa doble (if else) también se le denomina condición.

Para que se ejecute el <bloque\_de\_instrucciones\_1>, la condición tiene que ser verdadera. Por el contrario, si la condición es falsa, se ejecutará el <bloque\_de\_instrucciones\_2>.

En resumen, en C una instrucción if else (alternativa doble) permite seleccionar, por medio de una condición, el siguiente bloque de instrucciones a ejecutar, de entre dos posibles.

**3)** El if se utiliza para hacer una toma de decisiones lógicas, es decir, para ejecutar un código e ignorar otro código, dependerá de la expresión que realicemos.

Sentencia if

```
if (condition) {
    Instrucción // La instrucción se ejecuta si la condición es true o verdadera
}
```

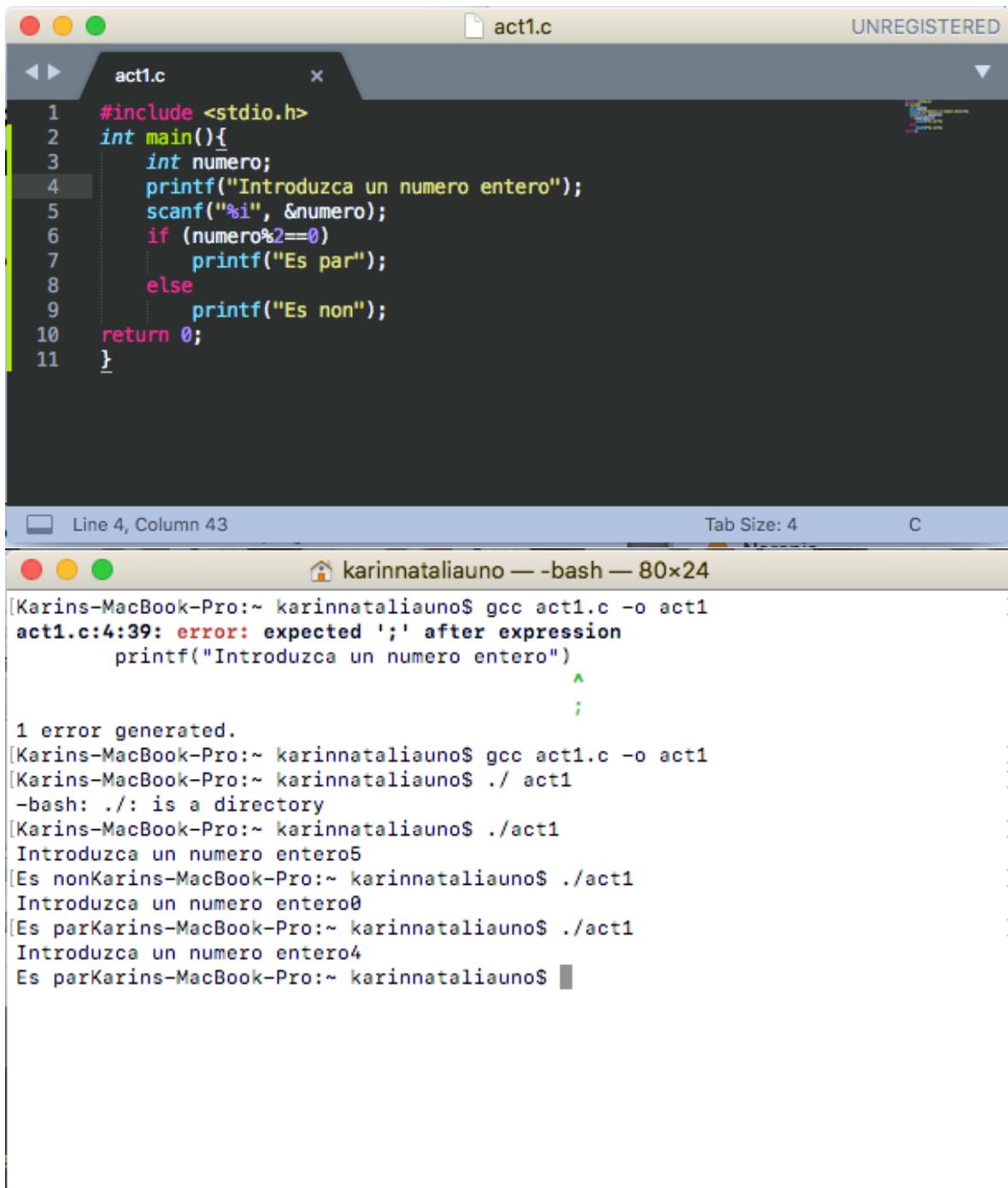
Los sentencia if comprueba si la condición que se encuentra entre los paréntesis ()

es verdadera o no. Si la condición es verdadera, la instrucción dentro de las {} del if se ejecuta, pero si la condición es falsa, la instrucción dentro del if se ignora.

Recopilado en: <http://codigocbasico.blogspot.com/2016/05/if-en-c.html>

## DESARROLLO/RESULTADOS

### Actividad 1. Par o non.



```
act1.c
UNREGISTERED

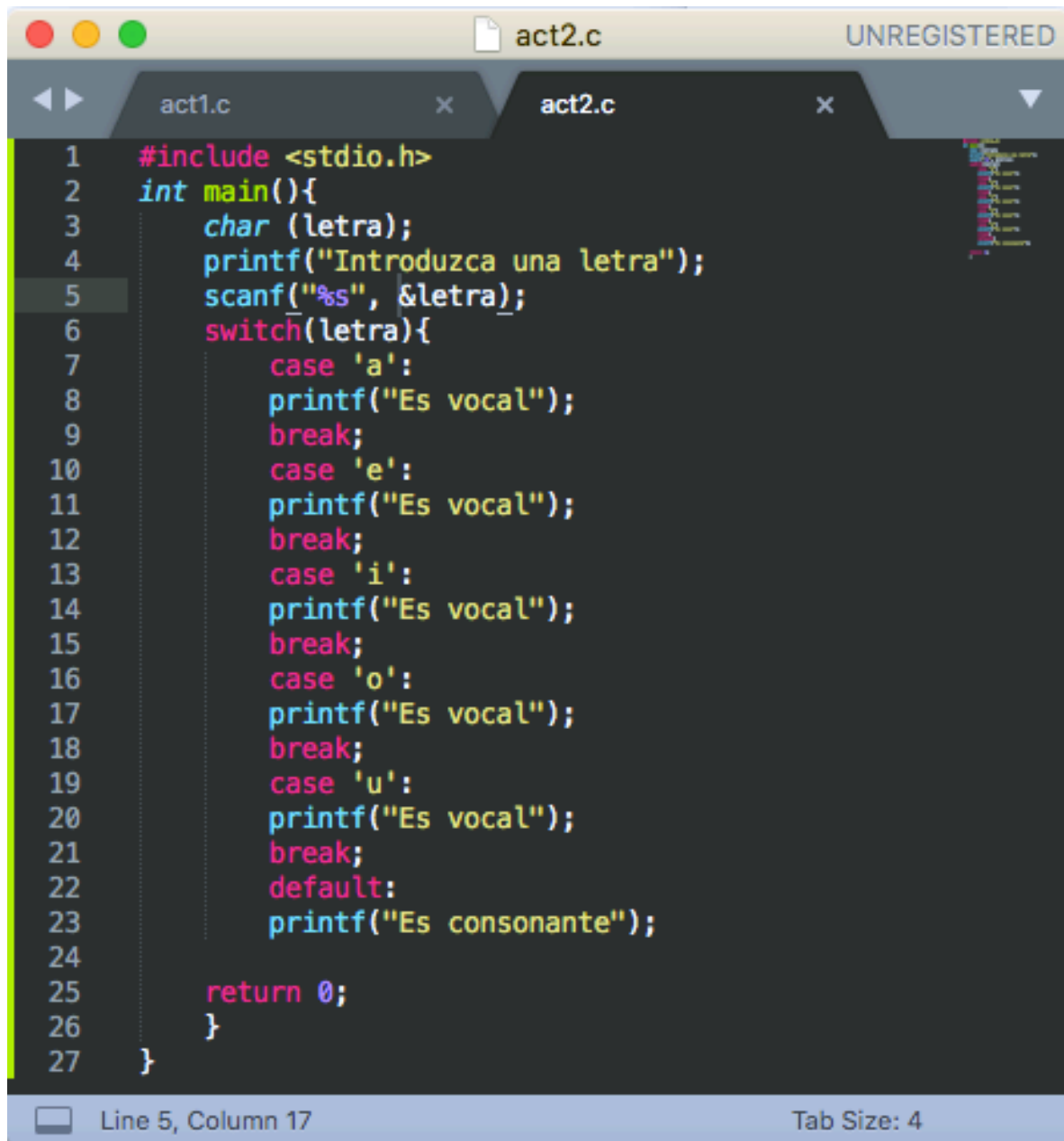
1  #include <stdio.h>
2  int main(){
3      int numero;
4      printf("Introduzca un numero entero");
5      scanf("%i", &numero);
6      if (numero%2==0)
7          printf("Es par");
8      else
9          printf("Es non");
10     return 0;
11 }
```

Line 4, Column 43      Tab Size: 4      C

```
karinnataliauno — -bash — 80x24

[Karins-MacBook-Pro:~ karinnataliauno$ gcc act1.c -o act1
act1.c:4:39: error: expected ';' after expression
    printf("Introduzca un numero entero")
                                   ^
                                   ;
1 error generated.
[Karins-MacBook-Pro:~ karinnataliauno$ gcc act1.c -o act1
[Karins-MacBook-Pro:~ karinnataliauno$ ./ act1
-bash: ./: is a directory
[Karins-MacBook-Pro:~ karinnataliauno$ ./act1
Introduzca un numero entero5
[Es nonKarins-MacBook-Pro:~ karinnataliauno$ ./act1
Introduzca un numero entero0
[Es parKarins-MacBook-Pro:~ karinnataliauno$ ./act1
Introduzca un numero entero4
Es parKarins-MacBook-Pro:~ karinnataliauno$
```

## **ACTIVIDAD 2. Vocal o consonante**



The image shows a code editor window with a title bar containing three colored window control buttons (red, yellow, green) on the left, a file icon and the name 'act2.c' in the center, and the word 'UNREGISTERED' on the right. Below the title bar, there are two tabs: 'act1.c' and 'act2.c', both with close buttons. The 'act2.c' tab is active. The editor area has a dark background with syntax-highlighted C code. Line numbers 1 through 27 are visible on the left side of the code. The code is as follows:

```
1  #include <stdio.h>
2  int main(){
3      char letra;
4      printf("Introduzca una letra");
5      scanf("%s", &letra);
6      switch(letra){
7          case 'a':
8              printf("Es vocal");
9              break;
10         case 'e':
11             printf("Es vocal");
12             break;
13         case 'i':
14             printf("Es vocal");
15             break;
16         case 'o':
17             printf("Es vocal");
18             break;
19         case 'u':
20             printf("Es vocal");
21             break;
22         default:
23             printf("Es consonante");
24
25     return 0;
26 }
27 }
```

At the bottom of the window, a status bar shows a cursor icon, the text 'Line 5, Column 17', and 'Tab Size: 4'.

```
karinnataliauno — -bash — 80x16
Introduzca un numero entero4
[Es parKarins-MacBook-Pro:~ karinnataliauno$ gcc act2.c -o act2 ]
act2.c:5:13: warning: format specifies type 'int *' but the argument has type
      'char (*)' [-Wformat]
      scanf("%d",&letra);
            ^~~~~~
            %s
1 warning generated.
[Karins-MacBook-Pro:~ karinnataliauno$ gcc act2.c -o act2 ]
[Karins-MacBook-Pro:~ karinnataliauno$ ./act2 ]
Introduzca una letrae
[Es vocalKarins-MacBook-Pro:~ karinnataliauno$ ./act2 ]
Introduzca una letray
[Es consonanteKarins-MacBook-Pro:~ karinnataliauno$ ./act2 ]
Introduzca una letrao
Karins-MacBook-Pro:~ karinnataliauno$
```

### ACTIVIDAD 3. Valor absoluto.

```
act3.c
UNREGISTERED
act1.c x act2.c x act3.c x
1 #include <stdio.h>
2 int main(){
3     int a,b;
4     int diff;
5     printf("Introducir un numero:");
6     scanf("%d",&a);
7     printf("Introducir otro numero:");
8     scanf("%d",&b);
9     if(a>b)
10        diff=a-b;
11    else diff=b-a;
12    printf("\nLa diferencia en el valor absoluto entre %d y %d es =%d", a, b, diff);
13    return 0;
14 }
Line 12, Column 50
Tab Size: 4
C
karinnataliauno — -bash — 90x18
Introduzca una letrao
[Karins-MacBook-Pro:~ karinnataliauno$ gcc act3.c -o act3 ]
[Karins-MacBook-Pro:~ karinnataliauno$ ./act3 ]
Introducir un numero:3
Introducir otro numero:5

[La diferencia en el valor absolutoentre 3 y 5 es =2Karins-MacBook-Pro:~ karinnataliauno$ ]
Karins-MacBook-Pro:~ karinnataliauno$ ./act3
Introducir un numero:20000
Introducir otro numero:182431242

[La diferencia en el valor absolutoentre 20000 y 182431242 es =182411242Karins-MacBook-Pro:~ karinnataliauno$ ./act3 ]
Introducir un numero:1435647897765432134567865432
Introducir otro numero:23456898765432134567898765

[La diferencia en el valor absolutoentre -1 y -1 es =0Karins-MacBook-Pro:~ kkkkkkkkkKaKKKKK ]
Karins-MacBook-Pro:~ karinnataliauno$
```

## **CONCLUSIÓN**

Podemos concluir que los programas en lenguaje C pueden ejecutarse de distintas maneras, en este caso se utilizó if, if-else, switch, así los programas serán más concretos en el proceso que se quiere llegar, haciéndolos más entendibles y con menos comandos. Además, estos programas creados en un editor de texto se pueden comprobar por medio de la misma terminal, así se podrá demostrar de manera explícita si el programa funciona o no.