

**OBJETIVO:** trabajar con cadenas de caracteres y ficheros de texto.

### Ejercicio 1.

Codifica una función de cabecera: `int iguales (char c1[], char c2[]);`  
que devuelve 1 si las cadenas de caracteres son iguales y 0 en caso contrario.

Probar la función con las siguientes cadenas:

papa y papa

papa y pepito

papa y papaito

papaito y papa

Copia a continuación el código de la función:

```
int iguales(char c1[],char c2[])
{
    int i=0,og=1;
    while(((c1[i]!='\0')||(c2[i]!='\0'))&&(og==1))
    {
        if(c1[i]==c2[i])
        {
            og=1;
        }
        else
        {
            og = 0;
        }
        i++;
    }
    return og;
}
```

### Ejercicio 2:

Codifica una función de cabecera:

```
int buscarPal (char vPal[N][NCAR], char pal[NCAR]);
```

que a partir de un array de cadenas de caracteres (vPal) y de una cadena de caracteres (pal) comprueba si pal está en vPal y devuelva el índice si lo encuentra y -1 si no lo encuentra.

NCAR y N son constantes definidas en el programa.

Se deberá utilizar la función del ejercicio 1.

Para probar la función se debe utilizar el siguiente segmento de programa:

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

```
#define NCAR 20      //define el tamaño máximo de las cadenas de caracteres
#define N 8         //cantidad de palabras del vector de palabras

int buscarPal (char vPal[N][NCAR], char pal[NCAR]);

int main()
{
    char cadena[NCAR]= "else"; //Cambiar de cadena para probar la función
    char vectorPal [N][NCAR] =
    {"for", "while", "if", "else", "switch", "int", "char", "void"};
    ...
}
```

Copia a continuación el código de la función:

```
int buscarPal(char vPal[N][NCAR], char pal[NCAR])
{
    int x, sol=-1;
    for(x=0;x<N;x++)
    {
        if(iguales(vPal[x],pal)==1)
        {
            sol=x;
        }
    }
    return sol;
}
```

### Ejercicio 3:

Disponemos de un fichero de texto que contiene un programa en C. Se pide hacer un programa que cuente cuántas veces aparece en el programa una determinada palabra reservada, por ejemplo, la palabra for.

Suponemos que la palabra reservada que se busca siempre aparece en el fichero con un blanco o fin de línea antes y después de la palabra.

Se deberán utilizar, al menos, la función del ejercicio 1.

Copia a continuación el código del programa:

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

```
#include <stdlib.h>
#define NCAR 80
int iguales(char c1[],char c2[]);

int main()
{
    char palRes[]="for";
    char pal[NCAR];
    int cont= 0;
    FILE *F;
    F = fopen("programa.txt","r");
    while(fscanf(F,"%s", pal)!=EOF)
    {
        if(iguales(pal,palRes)!=0)
        {
            cont++;
        }
    }
    fclose(F);
    if(cont==0)
    {
        printf("La palabra '%s' se repite una vez", palRes);
    }
    else
    {
        if(cont==1)
        {
            printf("La palabra '%s' se repite una vez", palRes);
        }
        else
        {
            printf("La palabra '%s' se repite %d veces", palRes,cont);
        }
    }
    return 0;
}

int iguales(char c1[],char c2[])
{
    int i=0,og=1;
    while(((c1[i]!='\0') || (c2[i]!='\0'))&&(og==1))
```

```
{
    if(c1[i]==c2[i])
    {
        og=1;
    }
    else
    {
        og = 0;
    }
    i++;
}
return og;
}
```

**Ejercicio 4:**

Disponemos de un fichero de texto que contiene un programa en C. Se pide hacer un programa que cuente cuántas veces aparece cada una de las N palabras reservadas que tendremos almacenadas en un vector.

Suponemos que la palabra reservada que se busca siempre aparece en el fichero con un blanco o fin de línea antes y después de la palabra.

Se deberán utilizar, al menos, las funciones de los ejercicios 1 y 2:

El nombre del fichero se leerá desde teclado, utilizando el siguiente código:

```
#define NCAR 30 //define el tamaño máximo de la palabra
#define N 8 //cantidad de palabras reservadas a buscar
. . .
char reservadas [N][NCAR] =
{ "for", "while", "if", "else", "switch", "int", "char", "void" };
char nombre[NCAR];
FILE *fProgram;
. . .
printf ("\nIntroduce el nombre de fichero que se debe abrir: ");
scanf ("%s", nombre);
fProgram= fopen (nombre, "r");

while (fProgram==NULL)
{
    printf ("\nFichero no encontrado, intentalo de nuevo\n");
    printf ("\nIntroduce el nombre de fichero que se debe abrir: ");
    scanf ("%s", nombre);
    fProgram= fopen (nombre, "r");
}
. . .
```

Ejemplo de ejecución del programa:

```
Introduce el nombre de fichero que se debe abrir: programa
Fichero no encontrado, intentalo de nuevo
Introduce el nombre de fichero que se debe abrir: programa.txt

Cantidad de veces que aparecen las palabras reservadas:
Palabra reservada: for         aparece 8 veces
Palabra reservada: while      aparece 0 veces
Palabra reservada: if         aparece 6 veces
Palabra reservada: else       aparece 6 veces
Palabra reservada: switch     aparece 0 veces
Palabra reservada: int        aparece 18 veces
Palabra reservada: char       aparece 5 veces
Palabra reservada: void       aparece 6 veces
```

Copia a continuación el código del programa:

```
#include <stdio.h>
#include <stdlib.h>
#define NCAR 30
#define N 8
int iguales(char c1[],char c2[]);
int buscarPal(char vPal[N][NCAR], char pal[NCAR]);

int main()
{
    char res[N][NCAR]={"for","else","while","if","char","void","int","switch"};
    char nombre[NCAR];
    int veces[N]={0,0,0,0,0,0,0,0},x,a;
    FILE *fProgram;
    printf("Introduce el nombre del fichero que quiere abrir:");
    scanf("%s", nombre);
    fProgram = fopen(nombre,"r");
    while(fProgram==NULL)
    {
        printf("Fichero no encontrado");
        printf("Introduce el nombre del fichero que quiere abrir:");
        scanf("%s", nombre);
        fProgram=fopen(nombre,"r");
    }
    printf("Cantidad de veces que salen alguna de las palabras");
```

```
while(fscanf(fProgram,"%s",nombre)!=EOF)

{
x = buscarPal(res,nombre);
if(x!=-1)
{
    veces[x]= veces[x]+1;
}
}
for(a=0;a<N;a++)
{
printf("palabra reservada: %s \taparece %2d veces.", res[a], veces[a]);
}
fclose(fProgram);
return 0;
}
int iguales(char c1[],char c2[])
{
int i=0,og=1;
while(((c1[i]!='\0') || (c2[i]!='\0'))&&(og==1))
{
if(c1[i]==c2[i])
{
og=1;
}
else
{
og = 0;
}
i++;
}
return og;
}

int buscarPal(char vPal[N][NCAR], char pal[NCAR])
{
int x, sol=-1;
for(x=0;x<N;x++)
{
```

```
    if(iguales(vPal[x],pal)==1)
    {
        sol=x;
    }

}
return sol;
}
```