Informe de Prácticas

Convocatoria extraordinaria de junio

|  |  |
| --- | --- |
| Apellidos y Nombre: | Gonzalez Bullido, Gonzalo |
| Curso: | 2016/17 |
| Grupo: | GT12 |
| Número de matrícula: | BN0027 |

Para entregar la práctica debes rellenar los siguientes apartados obligatoriamente utilizando el espacio que necesites en cada uno de ellos.

## 1. Breve explicación del algoritmo utilizado

Tras completar todas las funciones obligatorias y crear algunas auxiliares, en main he hecho uso de los ciclos **do while**,en mi caso 3 encadenados para que al final de adivinar una palabra el usuario pueda finalizar el programa, pedir otra palabra de la misma temática o cambiar de temática.

## 2. Descripción de las principales variables utilizadas en la función main

**i:** variable que toma el valor del total de letras sustituidas en la palabra a adivinar.

**numLetras:** nº de letras de la palabra a adivinar.

**linea:** nº de la línea de la matriz fichero que se ha elegido, donde se encuentra la palabra adivinar.

**opcionMP:** nº introducido por el usuario para elegir la temática.

**pregunta:**  nº introducido por el usuario para elegir si quiere resolver o introducir letra.

**igual:** en el caso de que el usuario haya decidido resolvertoma el valor 1 si está bien o 0 si está mal.

**numError:** nº de errores que se han cometido.

**letrasCorrectas:** letras sustituidas tras introducir una letra, estas se suman luego a la variable **i.**

**opcionTrasAcabar:** Variable condicion que rompe o reinicia los ciclos **do while** del main.

**fichero[MaxL][MaxC]**: La matriz fichero es donde se guardarán todas las palabras del .txt.

**palabra[MaxC]:** Array con lapalabra que se debe adivinar, se ha aislado de la matriz fichero.

**cadGuion[MaxC]:** cadena con tantos guiones como letras tiene la palabra a adivinar.

**letraUSR:** letra que va a introducir el usuario para adivinar la palabra.

**letrasIndroducidas[MaxC]:** array para ir guardando las letras que el usuario ha introducido y mostrárselas, usaremos ',' como separación entre ellas.

**resolverPalabra[MaxC]:** Matriz donde se guardará la entrada del usuario si quiere resolver.

## 3. Descripción de las funciones en las que se ha dividido el programa principal y breve explicación de su funcionamiento

El programa principal se divide en 3 ciclos **do while:**

**1.**El primero es simplemente para cerrar/finalizar el programa tras adivinar o fallar la resolución una palabra

**2.**El segundo ciclo sirve para volver imprimir el menú de temáticas y abre el fichero acorde a la temática elegida. Es decir, ese ciclo se reiniciará si el usuario al adivinar o fallar la resolución una palabra quiere cambiar de temática.

**3.**Este ciclo contiene el programa en si; elige una palabra al azar(mediante un nº aleatorio), inicializa todas las matrices sabiendo ya la palabra adivinar, imprime información sobre el progreso que lleva el usuario para adivinar la palabra (número de errores, letras ya introducidas, temática en la que se encuentra, el tamaño de la palabra mostrado en guiones y donde se sustituyen las letras acertadas, el dibujo del ahorcado acorde al número de errores), pregunta al usuario si quiere introducir letras o resolver, comunica al usuario si su letra era correcta o incorrecta, o si ha resuelto bien la palabra o no.

Además, en cada pregunta sobre introducir letras o resolver, se realiza una llamada a una función auxiliar que comprueba si se ha introducido una letra en el caso de que haya querido introducir letras y no cualquier carácter distinto a una letra; también si el usuario decide resolver comprueba que la palabra introducida este conformada por letras únicamente; en estos dos casos si no se cumplen esas condiciones se pedirá al usuario introducir un dato valido acorde a lo elegido.

Por otro lado, si el usuario mete una letra ya introducida anteriormente se avisará para que introduzca una nueva letra no introducida con anterioridad.

En el caso de la opción de introducir letras o resolver si el usuario introduce letras mayúsculas estas se transformarán a minúsculas mediante una función auxiliar que comprueba la existencia de letras mayúsculas y la intercambia por su letra minúscula correspondiente gracias al código ASCII.

Por último, tras adivinar o fallar la resolución de una palabra este ciclo se reiniciará si el usuario quiere una palabra de la misma temática.

## 4. Juego de pruebas utilizado para comprobar que la ejecución del programa es correcta

He utilizado 4 ficheros .txt de una temática distinta como se indica en el enunciado, cada .txt contiene 50 palabras. Lógicamente no he probado con las 200 palabras, pero he probado con todos los tamaños y he buscado fallas al programa introduciendo caracteres no acordes a lo que el programa me pide pedían y las he ido solventando a la marcha.

A continuación, mostrare el contenido de cada fichero:

**Primer fichero(Animales):**

leopardo

pantera

jaguar

chimpance

orangutan

gorila

pavo

gallina

pulpo

periquito

loro

garrapata

escorpion

abeja

oveja

cabra

raton

hamster

centollo

erizo

pez

jirafa

elefante

huron

mamut

lobo

koala

mariposa

babosa

gusano

zorro

tejon

hormiga

hipopotamo

tiburon

caballo

dragon

humano

mariquita

unicornio

escarabajo

medusa

anguila

tigre

gato

perro

leon

cocodrilo

tortuga

caracol

centollo

erizo

escorpion

abeja

oveja

cabra

raton

hamster

centollo

erizo

pavo

gallina

pulpo

periquito

loro

garrapata

escorpion

abeja

oveja

cabra

raton

hamster

centollo

erizo

**Segundo fichero(Profesiones):**

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

guardabosque

enfermero

taxista

chofer

piloto

militar

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

escritor

psicologo

actriz

cantante

mecanico

filosofo

guardabosque

enfermero

taxista

chofer

piloto

militar

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

periodista

ingeniero

quimico

fisico

matematico

terapeuta

escritor

psicologo

actriz

cantante

mecanico

filosofo

guardabosque

enfermero

taxista

chofer

piloto

militar

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

bombero

policia

panadero

asegurador

cocinero

profesor

periodista

ingeniero

quimico

fisico

matematico

terapeuta

escritor

psicologo

actriz

cantante

mecanico

filosofo

guardabosque

enfermero

taxista

chofer

piloto

militar

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

arquitecto

informatico

medico

abogado

biologa

farmaceutico

bombero

policia

panadero

asegurador

cocinero

profesor

periodista

ingeniero

quimico

fisico

matematico

terapeuta

escritor

psicologo

actriz

cantante

mecanico

filosofo

guardabosque

enfermero

taxista

chofer

piloto

militar

fotografo

modelo

pintor

encofrador

estilista

peluquero

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

astronauta

dermatologa

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

maquillador

dependiente

camarera

basurero

zapatero

politico

banquero

carcelero

albañil

empresario

geologo

astronoma

astronauta

dermatologa

**Tercer fichero(Nombres):**

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

maria

alberto

sara

lucia

lidia

gonzalo

stefan

jaime

brenda

aaron

caroline

elena

bonnie

hannah

victoria

ikram

arpita

manuel

sergio

ignacio

dora

leandro

lorenzo

carmen

alba

emily

alison

spencer

toby

mike

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

alba

emily

alison

spencer

toby

mike

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

sergio

ignacio

dora

leandro

lorenzo

carmen

alba

emily

alison

spencer

toby

mike

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

bonnie

hannah

victoria

ikram

arpita

manuel

sergio

ignacio

dora

leandro

lorenzo

carmen

alba

emily

alison

spencer

toby

mike

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

stefan

jaime

brenda

aaron

caroline

elena

bonnie

hannah

victoria

ikram

arpita

manuel

sergio

ignacio

dora

leandro

lorenzo

carmen

alba

emily

alison

spencer

toby

mike

amaya

marcos

ivan

alex

darren

miguel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

clara

isabel

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

teresa

marina

irene

aitor

fernando

leyda

jorge

ruth

ruben

adrian

diego

abel

clara

isabel

**Cuarto fichero(Países):**

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

andorra

monaco

mexico

venezuela

colombia

bolivia

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

francia

portugal

italia

grecia

inglaterra

noruega

polonia

belgica

irak

iran

pakistan

uzbekistan

andorra

monaco

mexico

venezuela

colombia

bolivia

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

marruecos

brasil

madagascar

chile

argentina

canada

francia

portugal

italia

grecia

inglaterra

noruega

polonia

belgica

irak

iran

pakistan

uzbekistan

andorra

monaco

mexico

venezuela

colombia

bolivia

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

suecia

suiza

islandia

egipto

irlanda

libia

marruecos

brasil

madagascar

chile

argentina

canada

francia

portugal

italia

grecia

inglaterra

noruega

polonia

belgica

irak

iran

pakistan

uzbekistan

andorra

monaco

mexico

venezuela

colombia

bolivia

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

alemania

china

mongolia

rusia

austria

australia

suecia

suiza

islandia

egipto

irlanda

libia

marruecos

brasil

madagascar

chile

argentina

canada

francia

portugal

italia

grecia

inglaterra

noruega

polonia

belgica

irak

iran

pakistan

uzbekistan

andorra

monaco

mexico

venezuela

colombia

bolivia

croacia

cuba

dinamarca

hungria

rumania

bulgaria

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

etiopia

haiti

panama

armenia

serbia

ucrania

congo

zimbabue

guatemala

nicaragua

finlandia

peru

ecuador

uruguay

panama

armenia

**Observación:** Lógicamente como C no permite tildes , todas las palabras están sin acentuar.

## 5. Referencias utilizadas

Para hacer el main y orientarme he utilizado la solución de la practica 12 ya que era un programa parecido al ahorcado

<https://moodle.upm.es/titulaciones/oficiales/pluginfile.php/1202302/mod_resource/content/11/practica12%20cadenas%2Bsol%20.pdf>

**OBSERVACIONES**

1. En el main no he podido utilizar “**return** 0**;**” pues tras varia iteraciones el buffer se llena y al intentar finalizar el programa tras adivinar o fallar la resolución de una palabra, el programa colapsa y deja de funcionar, he intentado solucionarlo poniendo antes del “**return** 0**;**” un:

“fflush**(** stdin **);**”

Pero no da resultado; tras más de una iteración el programa al intentar finalizar se colapsa así que sustituí el “**return** 0**;**” por el método de finalización puesto en el menú de elección de temática que es la función “exit**(**0**);**”; que no da errores tras mucha iteraciones y cierra el programa exitosamente. Esta función está incluida en la librería stdlib.h y está permitido su uso en esta práctica (Como dice en moddle “Respecto a las especificaciones de diseño, indicar que además de la librería  stdio.h  se podrán utilizar las librerías time.h  y  stdlib.h”).

1. Respecto a la función obligatoria “borrarConsola**();**”utilizo la función“system**(**"cls"**);**” ya que limpia la consola sin importar en el tamaño en el que se abra, esta función está incluida en la librería

stdio.h y una de las profesoras me dio permiso para utilizarla ya que es más óptima.

## 6. Código fuente en C

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MaxL 50 //Líneas max de fichero.

#define MaxC 25 //Máximos caracteres por línea.

//#####################################################################################//###################### **PROTOTIPOS DE FUNCIONESOBLIGATORIAS** ########################## //#####################################################################################

**void** cabecera**();**//imprime nombre y número de matrícula.

**void** borrarConsola**();**//Despeja la consola.

**int** longCad**(char** palabra**[**MaxC**]);**//Devuelve el nº de letras de la palabra a adivinar.

**int** cadenasIguales**(char** palabra**[**MaxC**],char** resolverPalabra**[**MaxC**]);**

//Si el usuario decide resolver la palabra se compara la entrada con la palabra a adivinar.

**void** iniciaCad**(char** cadGuion**[**MaxC**],int** tamPalabra**);**

//Rellena una cadena con tantos '-' como nº de letras de la palabra adivinar.

**int** actualizaCad**(char** palabra**[**MaxC**],char** cadGuion**[**MaxC**],char** letra**);**

//Si la letra está en la palabra a adivinar, introduce en la cadena de guiones la letra en su

//espacio correspondiente. He modificado la función a tipo int para que devuelva el nº de guiones

//que se han sustituido ya que por ejemplo la letra 'r' sustituye dos guiones en la palabra 'perro',

//entonces si se ha realizado un cambio la función devolverá un nº distinto de 0, por lo cual la

//letra del usuario era correcta, y si devuelve cero significa que no se ha sustituido ningún guion

//entonces la letra del usuario era errónea y por lo tanto avisar que introduzca otra.

//(ESTO AHORRA HACER UNA FUNCION AUXILIAR).

**int** actualizaLista**(char** LetrasIndroducidas**[**MaxC**],char** letra**);**

//Crea una lista con las letras que ha introducido el usuario y en caso de que introduzca una de la

//lista avisara de que esa letra ya ha sido introducida, y pedirá que se introduzca otra.

//(ESTO AHORRA HACER UNA FUNCION AUXILIAR).

**void** imprimeFigura**(int** NumError**);**//Imprime la figura correspondiente al número de errores.

//#####################################################################################//####################### **PROTOTIPOS DE FUNCIONES AUXILIARES** ########################## //#####################################################################################

**void** palabraAislada**(char** fichero**[**MaxL**][**MaxC**],char** palabra**[**MaxC**],** **int** linea**);**

//Copia la palabra propuesta de la matriz fichero en la cadena palabra.

**int** imprimirMenu**();**//Donde se elige la temática de la palabra

**void** aperturaFichero**(char** fichero**[**MaxL**][**MaxC**],int** opcion**);**

//Saca las palabras del .txt y las copia en la matriz fichero.

**int** numAleatorio**();**//Devuelve un numero aleatorio para la elección de una palabra.

**int** informacion**(int** numError**,char** letrasIndroducidas**[**MaxC**],char** cadGuion**[**MaxC**],int** tematica**);**

//Imprime la información que se menciona en el enunciado y también la temática antes de cualquier lectura, y //además devuelve '1' si el usuario quiere resolver o '0' si quiere seguir introduciendo letras.

**void** comprobador**(char** **\***letra**,char** resolverPalabra**[**MaxC**],int** elegir**);**

//Comprueba que, tanto en la entrada de letras como en el array para resolver la palabra, está conformada //solo por letras, en caso de que haya un carácter distinto de una letra avisara y se repetirá la entrada de //el dato. Para elegir que se va a comprobar está el valor int elegir, que será '1' si va comprobar la letra //metida por el usuario o'0' si va comprobar el array de caracteres con el que el usuario quiere resolver la //palabra. Tras comprobar que son letras, automáticamente se lanzara la función 'Conversor'.

**void** conversor**(char** **\***letra**,char** resolverPalabra**[**MaxC**],int** elegir**);**

//Toda entrada de usuario de tipo char, pasa por esta función y en el caso de que haya introducido cualquier

//carácter en mayúscula, se convertirá en minúscula, mediante una comprobación, si por ejemplo se introduce

//'A', se hará esta operación 'A'+32, que dará como resultado 'a'; esto es mejor ya que en vez de hacer un

//switch de todo el abecedario, se hace una simple operación para convertir mayúsculas en minúsculas.

//Para elegir que se va a comprobar está el valor int elegir, que será '1' si va comprobar la letra metida

//por el usuario o'0' si va comprobar el array de caracteres con el que el usuario quiere resolver la palabra.

//#####################################################################################//#################################### **MAIN** ########################################### //#####################################################################################

**int** main**()**

**{**

**int** i**,**numLetras**,**linea**,**opcionMP**,**pregunta**,**igual**,**numError**,**letrasCorrectas**,**opcionTrasAcabar**;**

**char** fichero**[**MaxL**][**MaxC**];**//La matriz fichero es donde se guardaran todas las palabras del .txt

**char** palabra**[**MaxC**];**//Palabra que se debe adivinar

**char** cadGuion**[**MaxC**];**//cadena con tantos guiones como letras tiene la palabra a adivinar

**char** letraUSR**;**//letras que va a introducir el usuario para adivinar la palabra

**char** letrasIndroducidas**[**MaxC**];**

//Son 10 letras solo, pero usaremos ',' como separación entre ellas por eso necesitamos más de 10 espacios //en la cadena y también inicializamos a NULL para indicar que todavía no se ha introducido ninguna letra.

**char** resolverPalabra**[**MaxC**];**//Matriz donde se guardará la entrada del usuario si quiere resolver

**do**//Por si al adivinar o fallar la palabra el usuario quiere finalizar

**{**

**do**

**{**

borrarConsola**();**

cabecera**();**

fflush**(** stdin **);**

opcionMP**=**imprimirMenu**();**

aperturaFichero**(**fichero**,**opcionMP**);**

**do**

**{**

linea**=**numAleatorio**();**

palabraAislada**(**fichero**,**palabra**,**linea**);**

numLetras**=**longCad**(**palabra**);**

iniciaCad**(**cadGuion**,**numLetras**);**

letrasIndroducidas**[**0**]=**'\0'**;**//Por si hay más de una iteración que se reinicie el

//valor NULL que es el centinela de impresión a la posición cero

i**=**0**;**

numError**=**0**;**

**while(**numLetras**!=**i **&&** numError**!=**10**)**

**{**

borrarConsola**();**

printf**(**"\n"**);**

pregunta**=**informacion**(**numError**,**letrasIndroducidas**,**cadGuion**,**opcionMP**);**

**if(**pregunta**==**1**)**//El usuario quiere seguir introduciendo letras

**{**

printf**(**"\nIntroduzca letra: "**);**

scanf**(**" %c"**,&**letraUSR**);**

comprobador**(&**letraUSR**,**resolverPalabra**,**1**);**

fflush**(** stdin **);**

conversor**(&**letraUSR**,**resolverPalabra**,**1**);**

**while(!**actualizaLista**(**letrasIndroducidas**,**letraUSR**))**

**{**

printf**(**"\nEsa letra ya ha sido introducida introduzca otra: "**);**

scanf**(**" %c"**,&**letraUSR**);**

fflush**(** stdin **);**

**}**

letrasCorrectas**=**actualizaCad**(**palabra**,**cadGuion**,**letraUSR**);**

**if(**letrasCorrectas**!=**0**)**

**{**

//La letra era correcta

printf**(**"\n\tLETRA CORRECTA :D \n\n"**);**

i**=**i**+**letrasCorrectas**;**

**}**

**else**

**{**

//La letra no era correcta

printf**(**"\n\tLETRA INCORRECTA :(\n\n"**);**

numError**++;**

**}**

system**(**"pause"**);**

**}**

**else**//El usuario ha decidido resolver

**{**

fflush**(** stdin **);**//He puesto fflush ya que se saltaba el gets

printf**(**"\nLa palabra a adivinar es: "**);**

gets**(**resolverPalabra**);**

comprobador**(&**letraUSR**,**resolverPalabra**,**0**);**

conversor**(&**letraUSR**,**resolverPalabra**,**0**);**

igual**=**cadenasIguales**(**palabra**,**resolverPalabra**);**

**if(**igual**)**

**{**

i**=**numLetras**;**//Para salir del bucle

**}**

**else**

**{**

numError**=**10**;**//Para salir del bucle

**}**

**}**

**}**

borrarConsola**();**

**if(**numLetras**==**i**)**

**{**

printf**(**"\n\n\n\t\t#####################################"**);**

**for(**i**=**0**;**i**<=**numLetras**;**i**++)** printf**(**"%c"**,**'#'**);**

printf**(**"\n\t\tFELICIDADES HAS ADIVINADO LA PALABRA:%s\n"**,**fichero**[**linea**]);**

printf**(**"\t\t#####################################"**);**

**for(**i**=**0**;**i**<=**numLetras**;**i**++)** printf**(**"%c"**,**'#'**);**

printf**(**"\n\n"**);**

**}**

**else** **if(** numError**==**10**)**

**{**

printf**(**"\n\n\n\nPalabra incorrecta,la palabra es:%s\n\n"**,**fichero**[**linea**]);**

**}**

printf**(**"%cQuiere otra palabra de la misma tematica(1),cambiar de tematica(2) o

finalizar(0)?: "**,**168**);**//168==¿, pero si pongo '¿' me imprime otro signo

scanf**(**"%d"**,&**opcionTrasAcabar**);**

**while(**opcionTrasAcabar**>**2 **||** opcionTrasAcabar**<**0**)**

**{**

printf**(**"\nEsa opcion no existe, porfavor introduzca 1 o 2 o 0 si quiere otra

palabra de la misma tematica, cambiar de tematica\no finalizar,

respectivamente: "**);**

fflush**(** stdin **);**

scanf**(**"%d"**,&**opcionTrasAcabar**);**

**}**

**}while(**opcionTrasAcabar**==**1**);**

**}while(**opcionTrasAcabar**==**2**);**

**}while(**opcionTrasAcabar**!=**0**);**

exit**(**0**);**

**}**

//#####################################################################################//############################## **FUNCIONES OBLIGATORIAS** ############################### //#####################################################################################

**void** cabecera**()**//########################################## **CABECERA** ###################

**{**

printf**(**"NOMBRE:GONZALO GONZALEZ BULLIDO\nMATRICULA: BN0027\n\n"**);**

**}**

**void** borrarConsola**()**//############################################# **BORRAR CONSOLA** ######

**{**

system**(**"cls"**);**

//Esta función la he sacado de la librería stdio.h, puesto que limpia la consola

//sin tener que poner saltos de línea sea cual sea el tamaño en el que se abra la

//consola ya que en cada ordenador el tamaño de la consola varia por lo tanto no

//se podría poner un número fijo de saltos de línea.

**}**

**int** longCad**(char** palabra**[**MaxC**])**//################################# **LONG CAD** ##############

**{**

**int** cont**=**0**;**

**while(**palabra**[**cont**]!=**'\0'**)**//En vez de poner '\0' también se podría poner NULL

**{**

cont**++;**

**}**

**return** cont**;**

**}**

**int** cadenasIguales**(char** palabra**[**MaxC**],char** resolverPalabra**[**MaxC**])**//**CADENAS IGUALES** ###

**{**

**int** acertada**=**0**,**i**=**0**;**

**while(**palabra**[**i**]!=**'\0' **&&** resolverPalabra**[**i**]!=**'\0' **&&** palabra**[**i**]==**resolverPalabra**[**i**])**

**{**

i**++;**

**}**

**if(**palabra**[**i**]==**resolverPalabra**[**i**])** acertada**=**1**;**

**return** acertada**;**

**}**

**void** iniciaCad**(char** cadGuion**[**MaxC**],int** tamPalabra**)**//################## **INICIA CAD** ##########

**{**

**char** gui**=**'-'**;**

**int** i**;**

**for(**i**=**0**;**i**<**tamPalabra**;**i**++)** cadGuion**[**i**]=**gui**;**

cadGuion**[**i**]=**'\0'**;**//Pongo caracter nulo para poder imprimir luego solo los guiones

**}**

**int** actualizaCad**(char** palabra**[**MaxC**],char** cadGuion**[**MaxC**],char** letra**)**//## **ACTUALIZA CAD** #######

**{**

**int** pos**,**resul**=**0**;**

**for(**pos**=**0**;**palabra**[**pos**]!=**'\0'**;**pos**++)**//Hacemos ciclo for porque la letra puede estar dos veces en una

//misma palabra.EJM: 'r' en perro

**{**

**if(**palabra**[**pos**]==**letra**)**

**{**

cadGuion**[**pos**]=**letra**;**

resul**++;**//La letra del usuario era correcta

**}**

**}**

**return** resul**;**

**}**

**int** actualizaLista**(char** letrasIndroducidas**[**MaxC**],cha**r letra**)**//######## **ACTUALIZA LISTA** ######

**{**

**int** i**=**0**;**

**while(**letrasIndroducidas**[**i**]!=**letra **&&** letrasIndroducidas**[**i**]!=**'\0'**)**

**{**

i**++;**

**}**

**if(**letrasIndroducidas**[**i**]==**'\0'**)**//La letra no se habria introducido se procede a incluirla en la lista

**{**

**if(**i**==**0**)**//Para la primera letra

**{**

letrasIndroducidas**[**i**]=**letra**;**

letrasIndroducidas**[**i**+**1**]=**'\0'**;**

**return** 1**;**

**}**

**else**//Demas letras

**{**

letrasIndroducidas**[**i**]=**','**;**

letrasIndroducidas**[**i**+**1**]=**letra**;**

letrasIndroducidas**[**i**+**2**]=**'\0'**;**//Ponemos '\0' para saber en una proxima llamada que hemoos

//llegado al final y tambien para imprimir hasta donde queremos

**return** 1**;**

**}**

**}**

**return** 0**;**//Ya se habia introducido esta letra

**}**

**void** imprimeFigura**(int** numError**)**//################################# **IMPRIMIR FIGURA** ######

**{**

**switch(** numError **)**

**{**

**case** 0**:** printf**(**"\t.......\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t. .\n"**);**

printf**(**"\t.......\n"**);**

**break;**

**case** 1**:** printf**(**"\n"**);**

printf**(**"\n"**);**

printf**(**"\n"**);**

printf**(**"\n"**);**

printf**(**"\n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 2**:**printf**(**"\n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 3**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 4**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 5**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 6**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| /| \n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 7**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| /|\\\n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 8**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| /|\\\n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| / \n"**);**

printf**(**"\t------\n"**);**

**break;**

**case** 9**:**printf**(**"\t\_\_\_\_\_\n"**);**

printf**(**"\t|/ O \n"**);**

printf**(**"\t| /|\\\n"**);**

printf**(**"\t| | \n"**);**

printf**(**"\t| / \\\n"**);**

printf**(**"\t------\n"**);**

**break;**

**default:** **break;**

**}**

**}**

//#####################################################################################//############################## **FUNCIONES AUXILIARES** ################################# //#####################################################################################

**void** palabraAislada**(char** fichero**[**MaxL**][**MaxC**],char** palabra**[**MaxC**],int** fila**)**//**PALABRA AISLADA** ###

**{**

**int** i**=**0**;**

**while(**fichero**[**fila**][**i**]!=**'\0'**)**

**{**

palabra**[**i**]=**fichero**[**fila**][**i**];**

i**++;**

**}**

palabra**[**i**]=**'\0'**;**//Pongo caracter nulo para tener un valor centinela al luego comparar

**}**

**int** imprimirMenu**()**//################################### **IMPRIMIR MENU** ##################

**{**

**int** opcion**;**

printf**(**"\tMENU PRINCIPAL\n\n"**);**

printf**(**"\t(1) Animales\n\n"**);**

printf**(**"\t(2) Profesiones\n\n"**);**

printf**(**"\t(3) Nombres\n\n"**);**

printf**(**"\t(4) Paises\n\n"**);**

printf**(**"\t(0) Salir\n\n"**);**//Por si el usuario quería finalizar pero se equivoca y elige volver al menú

//principal

printf**(**"\nElija una tematica o salga poniendo el numero correspondiente: "**);**

scanf**(**"%d"**,&**opcion**);**

**while(**opcion**>**4 **||** opcion**<**0**)**

**{**

printf**(**"\nHa introducido una opcion que no esta en el menu, por favor introduzca una

viable: "**);**

fflush**(** stdin **);**

scanf**(**"%d"**,&**opcion**);**

**}**

**return** opcion**;**

**}**

**void** aperturaFichero**(char** fichero**[**MaxL**][**MaxC**],int** opcion**)**//##### **APERTURA DE FICHERO** #######

**{**

**int** i**=**0**;**

**FILE** **\***F**;**

**switch(** opcion **)**

**{**

**case** 1**:** F **=** fopen **(**"Animales.txt"**,**"r"**);**

**break;**

**case** 2**:** F **=** fopen **(**"Profesiones.txt"**,**"r"**);**

**break;**

**case** 3**:** F **=** fopen **(**"Nombres.txt"**,**"r"**);**

**break;**

**case** 4**:** F **=** fopen **(**"Paises.txt"**,**"r"**);**

**break;**

**case** 0**:** exit**(**0**);**//Comando para terminar la ejecución entera del programa

**break;**

**}**

**if(**F**==NULL)**

**{**

printf**(**"\n\t\tERROR:el fichero no se ha abierto correctamente\n"**);**

exit**(**0**);**

**}**

**else**

**{**

//el fichero se ha abierto correctamente

fscanf**(**F**,**"%s"**,**fichero**[**i**]);**

**while(!**feof**(**F**))**

**{**

i**++;**

fscanf**(**F**,**"%s"**,**fichero**[**i**]);**

**}**

fclose**(**F**);**

**}**

**}**

**int** numAleatorio**()**//################################## **NUMERO ALEATORIO** ################

**{**

srand**(**time**(NULL));**

**return** rand**()%**50**;**

**}**

**int** informacion**(int** numError**,char** letrasIndroducidas**[**MaxC**],char** cadGuion**[**MaxC**],int** tematica**)** //############################# **INFORMACION** ############################

**{**

**int** j**=**0**,**opcion**;**

printf**(**"TEMATICA:"**);**//Imprime la tematica de la palabra a adivinar

**if(**tematica**==**1**)** printf**(**"ANIMALES\n"**);**

**else** **if(**tematica**==**2**)** printf**(**"PROFESIONES\n"**);**

**else** **if(**tematica**==**3**)** printf**(**"NOMBRES\n"**);**

**else** printf**(**"PAISES\n"**);**

printf**(**"Errores=%d de 10\n"**,**numError**);**//Imprime numero de errores

printf**(**"Ustes ya ha introducido estas letras: "**);**//Imprime las letras ya introducidas

puts**(**letrasIndroducidas**);**

printf**(**"\n"**);**

j**=**0**;**

imprimeFigura**(**numError**);**//Imprime la figura del ahorcado correspondiente al número de error

printf**(**"\n"**);**

printf**(**"Su progreso:"**);**

puts**(**cadGuion**);**//Imprime la cadena de guiones

printf**(**"\n"**);**

printf**(**"\n"**);**//Aqui se pregunta si el usuario quiere resolver o introducir letras

printf**(**"%cQuiere introducir letras (1) o resolver(0)?: "**,**168**);**//168==¿, pero si pongo '¿' me

//imprime otro signo

scanf**(**"%d"**,&**opcion**);**

**while(**opción**<**0 **||** opción**>**1**)**

**{**

printf**(**"\nEsa opcion no existe, porfavor introduzca 1 o 0 si quiere seguir introduciendo

letras o resolver, respectivamente: "**);**

fflush**(** stdin **);**

scanf**(**"%d"**,&**opcion**);**

**}**

**return** opcion**;**

**}**

**void** comprobador**(char** **\***letra**,char** resolverPalabra**[**MaxC**],int** elegir**)**//##### **COMPROBADOR** #####

**{**

**int** j**=**0**;**

**if(**elegir**)**

**{**

**while((\***letra**<**'A' **||** **\***letra**>**'Z'**)&&** **(\***letra**<**'a'**||** **\***letra**>**'z'**))**//Comprueba si se ha es una letra

**{**

printf**(**"\nNo ha introducido una letra, porfavor introduzca una letra: "**);**

scanf**(**" %c"**,**letra**);**

fflush**(** stdin **);**

**}**

**}**

**else**

**{**

**while(**resolverPalabra**[**j**]!=**'\0'**)**//Recorre el array en busca de caracteres distintos a letras

**{**

**if** **((**resolverPalabra**[**j**]<**'A' **||** resolverPalabra**[**j**]>**'Z'**)&&** **(**resolverPalabra**[**j**]<**'a'**||**

resolverPalabra**[**j**]>**'z'**))**

**{**

j**=**0**;**

printf**(**"\nHa introducido algún carácter distinto de una letra,escriba solo con letras:"**);**

gets**(**resolverPalabra**);**

fflush**(** stdin **);**

**}**

**else** j**++;**

**}**

**}**

**}**

**void** conversor**(**char **\***letra**,**char resolverPalabra**[**MaxC**],int** elegir**)**//######## **CONVERSOR** #######

**{**

**int** j**=**0**;**

**if(**elegir**)**

**{**

**if(\***letra**>=**'A' **&&** **\***letra**<=**'Z'**)**//Comprueba si la letra es una mayuscula y la tranformar a minuscula

**{**

**\***letra**=\***letra**+**32**;**//Ejemplo si \*letra='A' entonces al hacer \*letra+32, \*letra='a';

**}**

**}**

**else**

**{**

**while(**resolverPalabra**[**j**]!=**'\0'**)**//Recorre el array en busca de mayúsculas y las transforma en

//minisculas mediante el mismo metodo que las letras

**{**

**if** **(**resolverPalabra**[**j**]>=**'A' **&&** resolverPalabra**[**j**]<=**'Z'**)**

**{**

resolverPalabra**[**j**]=**resolverPalabra**[**j**]+**32**;**

j**++;**

**}**

**else** j**++;**

**}**

**}**

**}**