

# Pràctica 11 de programació en C

- Una cadena o string és un vector de caràcters (char).
- En C les cadenes acaben amb el caràcter 'fi de cadena', representat mitjançant '\0'.
- Per exemple, si fem la declaració `char cadena[]="Hola"`, la variable cadena quedaria emmagatzemada com

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
'H'	'o'	'l'	'a'	'\0'

- La funció `strlen` retorna el nombre de caràcters d'una cadena. Per exemple, si escrivim `strlen(cadena)` ens tornaria el valor 4

El penjat és un joc conegut en què un jugador pensa una paraula que un altre jugador ha d'endevinar:

```
Llevas 0 fallos
Puedes permitirte 6 fallos

Palabra: _____
Introduce la letra: a
Muy bien, has encontrado la 'a' y aparece 1 veces

Llevas 0 fallos
Puedes permitirte 6 fallos

Palabra: ___a___
Introduce la letra: e
Muy bien, has encontrado la 'e' y aparece 1 veces

Llevas 0 fallos
Puedes permitirte 6 fallos

Palabra: ___a_e_
Introduce la letra: c
Lo siento, la 'c' no esta entre las letras que faltan

Llevas 1 fallos
Puedes permitirte 5 fallos

Palabra: ___a_e_
Introduce la letra:
```

# El Juego del Ahorcado

- Inicialment es mostra la paraula ocultant les lletres mitjançant el caràcter \_ i es permet al segon jugador que pregunte per les diferents lletres de l'abecedari, una per torn.
- Si la lletra es troba entre les que formen la paraula a endevinar, aquesta es mostra en la posició o posicions correctes.
- Si la lletra no es troba entre les de la paraula oculta, s'anota una errada.
- El joc continua fins que s'endevina la paraula oculta o es cometen 6 errades.



## ahorcado.c (Poliformat)

Genera un número aleatori per a seleccionar la paraula oculta

Implementa aquestes funcions

Implementa el que apareix en els comentari

### Constants:

- **MAXPAL** = Nombre màxim de caràcters de la paraula oculta
- **NPALABRAS** = Nombre de paraules contingudes en el fitxer de paraules ocultes
- **MAXFALLOS** = Nombre màxim de fallades permeses

### Variables que apareixen en la funció main:

- **pal\_oculta**: cadena de caràcters que conté la paraula oculta
- **tam\_paloculta**: emmagatzema el nombre de caràcters de la paraula oculta
- **palabra**: cadena de caràcters que conté la cadena que es va formant

```
1 #include <stdio.h>
2 #include <time.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <stdlib.h> // Para malloc
6 #define MAXPAL 31
7 #define NPALABRAS 26457
8 #define MAXFALLOS 6
9
10 /* Función numpal */
11 int numpal(){
12
13     srand(time(NULL)+getpid());
14     return (rand()%NPALABRAS);
15 }
16
17 /* Escribe la función selecciona_palabra */
18
19 /* Escribe la función inicializa_palabra */
20
21 /* Escribe la función busca_letra */
22
23 int main(){
24     /* Añade la declaración de las variables que se necesiten */
25     char pal_oculta[MAXPAL], palabra[MAXPAL];
26     int tam_paloculta;
27
28     /* Leemos de fichero la palabra oculta -> función selecciona_palabra */
29
30     /* si hay error en la lectura -> finalizar el programa */
31
32     /* calculamos el tamaño de la palabra oculta */
33
34     /* inicializamos la palabra a una cadena de guiones '-' -> función inicializa_palabra */
35
36     /* mientras el número de aciertos sea menor que el tamaño
37     de la palabra oculta y el número de fallos menor que el
38     máximo de fallos -> el juego continúa */
39     while ( /* Completa la condición */){
40
41         /* se muestra el número de fallos que se lleva acumulados */
42         /* y los que todavía se pueden realizar */
43
44         /* Se muestra el vector palabra */
45
46         /* se lee la letra con la que se juega por teclado */
47         printf("Introduce la letra: ");
48         fflush(stdin);
49         scanf("%c", &letra);
50
51         /* se busca la letra en la palabra oculta -> invocar a la función busca_letra */
52
53         /* Según el resultado de la búsqueda indicar si se ha acertado, se ha fallado
54         o se ha introducido una letra ya adivinada -> Actualizar contadores de fallos y aciertos */
55     }
56
57     /* Indicar el resultado del juego -> Se ha ganado o perdido */
58     /* Si se pierde mostrar la palabra oculta */
59     /* Si se gana mostrar el número de fallos realizados */
60
61     return 0;
62 }
63
64
65
```

```
/* Escribe la función selecciona_palabra */
int selecciona_palabra(char pal_oculta[]){
    /*poner código*/
}

/* Escribe la función inicializa_palabra */
void inicializa_palabra(char palabra[], int n){
    /*poner código*/
}

/* Escribe la función busca_letra */
int busca_letra(char paloculta[], char palabra[], char letra, int n){
    /*poner código*/
}
```

**selecciona\_palabra**: aquesta funció rep el vector *pal\_oculta* i emmagatzema en ell la paraula a endevinar. La paraula a endevinar es llegirà del fitxer *palabras.txt*. Per a açò, s'utilitzarà la funció *numpal* que ens retorna un nombre enter a l'atzar entre 0 i 26456 (aquesta funció es proporciona ja implementada). Si el valor retornat per la funció *numpal* és *i*, s'escollirà com a paraula oculta exactament la paraula que es troba en la línia *i+1* del fitxer. La funció retornarà -1 si es produeix algun error en obrir el fitxer o 0 si s'ha pogut escollir la paraula oculta.

Fichero palabras.txt:

```
a  
ababa  
ababol  
abacero  
abacial  
abad  
abadejo  
abadengo  
abadesa  
.....
```

Declaració de la funció:

```
int selecciona_palabra(char pal[]){  
    FILE *df;  
    int i, indpal;  
    indpal=numpal();  
    .....  
}
```

La invocació que es fa des del programa principal és:

```
err=selecciona_palabra(pal_oculta);
```

Exemple: Si el valor retornat per la funció és 3, esta funció retornarà en **pal\_oculta** la paraula **abacero**.

**inicializa\_palabra:** aquesta funció rep el vector *palabra* i la grandària de la paraula a buscar (*tam\_paloculta*). Posa en el vector el caràcter '-' (des del principi) tantes vegades com indica *tam\_paloculta*, i, darrere, posa el caràcter '\0'.

**busca\_letra:** aquesta funció rep el vector *pal\_oculta*, el vector *palabra*, la lletra a cercar *letra*, i el nombre de caràcters de la paraula oculta *n*. La funció cerca la lletra *letra* en el vector *pal\_oculta* i realitza les següents accions:

- Si la troba i no havia sigut ja endevinada, modifica el vector *palabra* substituint el caràcter '-' per *letra*. Retorna el nombre de substitucions realitzades.
- Si la troba però ja havia sigut endevinada, retorna -1. Per a saber si ja havia sigut endevinada es pot consultar si en eixa mateixa posició en la variable *palabra* hi ha un caràcter diferent de '-'.
- Si no la troba, retorna 0.

Exemple: Si, per exemple, *pal\_oculta* conté la cadena calabaza i en invocar la funció la cadena *palabra* conté la cadena c \_ \_ \_ b \_ \_ \_ i el caràcter emmagatzemat en la variable *letra* és 'a', aleshores el nou contingut de *palabra* serà c a \_ a b a \_ a, retornant 4, ja que s'han fet 4 substitucions.

Si la lletra no es troba en *pal\_oculta*, la funció retorna 0.

En el programa principal cal implementar el codi que s'indica:

```
29  /* leemos de fichero la palabra oculta -> función selecciona_palabra */ err=selecciona_palabra(pal_oculta);
30
31  /* si hay error en la lectura -> finalizar el programa */ if(err == -1){
32  /* calculamos el tamaño de la palabra oculta */ tam_paloculta = strlen(pal_oculta); printf("Error abriendo fichero palabras.txt\n");
33  /* inicializamos la palabra a una cadena de guiones '-' -> función inicializa_palabra */ return -1;}
34  /* mientras el número de aciertos sea menor que el tamaño inicializa_palabra (palabra,tam_paloculta);
35  de la palabra oculta y el número de fallos menor que el
36  máximo de fallos -> el juego continúa */
37
38
39
```

```
/* leemos de fichero la palabra oculta -> función selecciona_palabra */
```

```
/* si hay error en la lectura -> finalizar el programa */
```

```
/* calculamos el tamaño de la palabra oculta */
```

```
/* creamos el vector palabra dinámicamente -> función malloc */
```

```
/* si no se puede crear -> finalizar programa*/
```

```
/* inicializamos la palabra a una cadena de guiones bajos '_' -> función inicializa_palabra */
```

```
/* mientras el número de aciertos sea menor que el tamaño  
de la palabra oculta y el número de fallos menor que el  
máximo de fallos -> el juego continúa */
```

```
while ( /* Completa la condición */){  
  /* se muestra el número de fallos que se lleva acumulados */  
  /* y los que todavía se pueden realizar */  
  /* Se muestra el vector palabra */  
  /* se lee la letra con la que se juega por teclado */  
  printf("Introduce la letra: ");  
  fflush(stdin);  
  scanf("%c", &letra);  
  /* se busca la letra en la palabra oculta -> invocar a la función busca_letra */  
  /* Según el resultado de la búsqueda indicar si se ha acertado, se ha fallado  
  o se ha introducido una letra ya adivinada -> Actualizar contadores de fallos y aciertos */  
  .....  
}
```

```
Llevas 0 fallos  
Puedes permitirte 6 fallos  
  
Palabra: __a_e_  
Introduce la letra: c  
Lo siento, la 'c' no esta entre las letras que faltan
```

Muchas gracias  
y  
¡ a programar !

