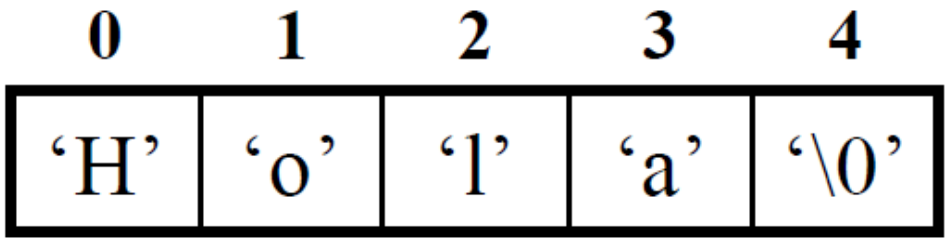
Práctica 11 de programación en C

Cadenas de caracteres

- Una cadena o string es un vector de caracteres (char).
- En C las cadenas acaban con el carácter 'fin de cadena', representado mediante '\0'
- Por ejemplo, si hacemos la declaración

char cadena[]="Hola", la variable cadena quedaría almacenada como



- La función **strlen** devuelve el número de caracteres de una cadena. Por ejemplo, sin escribimos strlen(cadena) nos devolvería el valor 4.

El Juego del Ahorcado

El ahorcado es un conocido juego por el cual un jugador piensa una palabra que debe adivinar otro jugador.

```
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

- Inicialmente se muestra la palabra ocultando las letras mediante el carácter _ y permite al segundo jugador que pregunte por las diferentes letras del abecedario, una por turno.
- Si la letra se encuentra entre las que componen la palabra a adivinar, esta se muestra en la posición o posiciones correctas.
- Si la letra no se encuentra entre las de la palabra oculta, se apunta un fallo.
- El juego continúa hasta que se adivina la palabra oculta o se cometan 6 fallos.

ahorcado.c (bajar de Poliformat)

ESCOLA TÈCNICA SUPERIOR ENGINYERIA INDUSTRIAL VALÈNCIA

```c
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h> // Para malloc
#define MAXPAL 31
#define NPALABRAS 26457
#define MAXFALLOS 6

/* Función numpal */
int numpal(){

  srand(time(NULL)+getpid());
  return (rand()%NPALABRAS);
}

/* Escribe la función selecciona_palabra */

/* Escribe la función inicializa_palabra */

/* Escribe la función busca_letra */

int main(){
  /* Añade la declaración de las variables que se necesiten */
  char pal_oculta[MAXPAL], palabra[MAXPAL];
  int tam_paloculta;


  /* 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 */

  /* inicializamos la palabra a una cadena de guiones '-' -> 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 */

  }

  /* Indicar el resultado del juego -> Se ha ganado o perdido */
  /* Si se pierde mostrar la palabra oculta */
  /* Si se gana mostrar el número de fallos realizados */

  return 0;
}
```

Genera un nº aleatorio para seleccionar la palabra oculta

Implementa estas funciones

```c
/* 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*/
}
```

Implementa lo que aparece en los comentarios

- **Constantes**:

MAXPAL= Número máximo de caracteres de la palabra oculta

NPALABRAS= Número de palabras contenidas en el fichero de palabras ocultas

MAXFALLOS= Número máximo de fallos permitidos

- **Variables** que aparecen en la función main:

pal_oculta: cadena de caracteres que contiene la palabra oculta

tam_paloculta: almacena el nº de caracteres de la palabra oculta

palabra: cadena de caracteres que contiene la cadena que se va formando

int selecciona_palabra(char pal_oculta[])

Esta función tiene como parámetro de salida la cadena de caracteres **pal_oculta**, la cual almacenará la palabra a adivinar. Seguir los siguientes pasos:

- Invoca a la función **numpal**, para que nos devuelve un número entero al azar entre 0 y NPALABRAS-1:

```
indpal=numpal(); //invocamos a la funcion
```

- Abrir el fichero **palabras.txt**, devolviendo -1 si hay error (return -1)
- Mediante un bucle **for** comenzando en 0 hasta ese valor (**indpal**), almacena en **pal_oculta** la cadena leída del fichero, usando el carácter de control %s.
- Si no ha habido error, el programa devolverá el valor 0 (return 0).

Ejemplo: Si el valor **indpal** fuese 3, entonces **pal_oculta** contendría la cadena " abacero "

Fichero palabras.txt:

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

void inicializa_palabra(char palabra[], int n)

Mediante un bucle for rellena las componentes 0, 1, 2, …,n-1 del array **palabra** con el carácter '_' y la componente n-ésima con el carácter de control '\0'.

int busca_letra(char paloculta[], char palabra[], char letra, int n)

La función busca el carácter almacenado en la variable **letra** dentro del vector **pal_oculta**, utilizando para ello un bucle for:

- Si la letra se encuentra en el vector **palabra**, la función devuelve -1 (return -1).
- Si la letra se encuentra en el vector **pal_oculta**, para cada coincidencia hay que sustituir el carácter '_' por el contenido de la variable **letra.** La función devolverá el número de sustituciones realizadas.

  Ejemplo: Si, por ejemplo, **pal_oculta** contiene la cadena **calabaza** y al invocar a la función la cadena **palabra** contiene la cadena **c _ _ _ b _ _ _** y el carácter almacenado en la variable **letra** es 'a' ,entonces el nuevo contenido de **palabra** será **c a _ a b a _ a**, devolviendo 4, pues se han hecho 4 sustituciones.
- Si la letra no se encuentra en **pal_oculta**, la función devuelve 0.

En el programa principal hay que implementar el código que se indica

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

```
/* 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 !