Cada archivo de cabecera añade a nuestro programa todo un conjunto de funciones que expande el lenguaje.   Es necesario hacer uso de la directiva #include. El signo # indica que la instrucción va

para el compilador o preprocesador no para el programa ejecutable.  El nombre de archivo  se escribe entre los  signos “” para archivos cabecera guardados en la  carpeta predeterminada o entre (“ “) si son personales y se suelen guardar junto con el archivo fuente.

**Líneas 1-4:**

-#include <conio.h> :*<conio.h>* es el encabezado de una biblioteca no estándar de C, que incluye funciones específicas como (getch(), (esta es muy útil si desea leer una entrada de caracteres en un teclado),para el control de entrada y salida por consola de una determinada plataforma, tal como DOS. Puede usarse en C++ para la funcionalidad requerida.  Esta cabecera declara varias funciones útiles para mejorar el rendimiento de la «entrada y salida por consola» desde un programa.

- #include <stdio.h>

Esta línea es necesaria en nuestro programa para poder utilizar funciones de entrada y salida estándar. La entrada y salida estándar es la consola, en la que se pueden imprimir mensajes así como leer lo que teclea el usuario. El significado intuitivo de esta línea de código es decir que “incluya” (include) el fichero stdio.h (std=estándar, io=input, output, entrada y salida; stdio: entrada y salida estándar).

-#include<stdlib.h>esta línea es para poder utilizar los comandos srand() y rand(), que sirven para que el programa seleccione un número aleatorio del 1 al número que el programador indique.

-#include<string.h>Muchas de las funciones de manejo común de cadenas de caracteres no es necesario implementarlas, ya que ya están programadas en una librería, cuyo fichero de cabecera que hay que incluir se llama

**Líneas 6-9**

-Comando struct:  
Para declarar un registro (un registro es un conjunto de datos, pudiendo ser cada uno de estos de un tipo diferente), se recurre al comando ***struct***, especificando entre llaves cada campo.

En la línea se usa struct para hacer un registro de las figura de nuestro juego programa que almacene los datos. Además, se mostrará esta información una vez que la hayamos introducido.

El código del programa donde se observa la manera de declarar un registro aparece a continuación, así como la salida en el terminal.

Toda variable que vaya a ser utilizada en un programa debe ser previamente declarada.Declarar una variable es definir su tipo, asignarle un nombre y opcionalmente asignarle un valor inicial. El tipo indica los valores que puede tomar (enteros, decimales, caracteres, etc.) y las operaciones que están permitidas, con el objeto de ahorrar espacio en memoria. Los tipos de datos enteros son: char, short, int, long

La diferencia entre los distintos tipos de datos enteros y reales, es el tamaño que ocupan en memoria, o lo que es lo mismo, el número de bytes que utilizan para codificar el valor correspondiente. En función de dicho tamaño, el rango de valores numéricos admisibles varia. En general se puede afirmar que el tamaño char<=short<=int<=long y que el tamaño float<=double.

-Comando char nombre: Para almacenar letras o caracteres se utiliza el tipo de datos char (o unsigned char). Dado que este tipo de datos e en la Tabla ASCII.

Si el tipo del vector definido es de tipo char se trata de una es de números enteros, lo que realmente se guarda en la variable declarada es el número entero correspondiente al caracter, tal y como se defincadena de caracteres, en este caso 30. En una cadena cada caracter ocupa una posición de memoria (1 Byte) y debe terminar siempre con el caracter nulo (\0), que el programador debe incluir de forma obligada. Tal como sucede en los vectores, el nombre de la cadena de caracteres es un puntero al primer elemento. Cada caracter se almacena como su código ASCII y el caracter nulo como 0. Una de las utilidades del caracter nulo radica en indicar a las funciones que manejan caracteres donde termina la cadena. Estas funciones se encuentran en el fichero de cabecera string.h. Si por error no se incluye el \0 se tendrá error de ejecución.

Aquí estas declarando una variable, nombre, Este comando, en el programa, sirve simplemente para introducir el nombre que dar a tu usuario/avatar.

-Comando int num: Sirve para introducir la nueva variable num, y lo haremos con la palabra clave ‘int’ abreviatura de “integer”, entero en inglés.

**Líneas 11-16:**

-Comando int main: Esta línea define el punto de entrada al programa, esto es, la función main() o principal. Cualquier programa en C necesita la existencia de una función denominada main() y solo una función puede ser denominada así. Y

La llave de apertura define el comienzo de un nuevo bloque de sentencias. En este caso esta llave define el comienzo del bloque de sentencias correspondiente a la función main(). lo haremos con la palabra clave ‘int’ abreviatura de “integer”, entero en inglés.

-Comando int J1, J2, num, i, j: Sirve para introducir las nuevas variables J1, J2, num, i, j:, y lo haremos con la palabra clave ‘int’ abreviatura de “integer”, entero en inglés.

-Comando struct figuras colección: Este comando sirve para introducir un registro de cada figura de nuestro juego programa asociando cada una de las 16 figuras un número, del 1 al 15.

**Líneas 18-33:**

-Comandos char: Los 16 comandos char de estas líneas son las presentaciones de cada uno de los 16 personajes que contiene el fuego.Cada char tienen un número asignado, son los que hemos asociado a cada figura del juego previamente.

Líneas 35-52: En estas líneas el jugador entra al menú y se le da 4 opciones, jugar contra el ordenador, acceder al ranking, las reglas del juego y salir del juego.

-Líneas 35-37: Estas 3 líneas son la primera parte del menú, en la que se te da la bienvenida y te preguntan lo que deseas hacer.

-Líneas 41-45: En estas líneas se te muestra el Menú, con sus respectivas opciones: jugar contra el ordenador, acceder al ranking, las reglas del juego y salir del juego, cada una de las opciones tiene asociado un numero del 1, 2, 3 y 4, en ese orden.

-Líneas 48-52: Aquí se le pide al jugador que introduzca la opción del Menú con la que quiere interactuar, pidiéndote un número del 1 al 4, siendo estos números las opciones del Menú.

**Línea 54 y línea 57:**

-Línea 54: El programa introduce un bucle. Los bucles son sentencias de control que sirven para realizar tareas repetitivas, se pueden introducir utilizando las instrucciones for, while, en este caso while. Estamos introduciendo el bucle para que en función del número introducido previamente, el programa haga una de las opciones del Menú.

-Línea 57: hasta ahora la ejecución de sentencias es totalmente secuencial, se van ejecutando unas detrás de otras, hasta llegar al final de la función main(). No obstante si queremos que el programa ejecute una función en función de una decisión, que son palabras claves de C que pueden alterar el ciclo de ejecución secuencial del programa. La sentencia condicional switch-case sirven para tomar decisiones en función de unas determinadas condiciones, para ejecutar unas sentencias u otras.

**Líneas 60-84:**

Como está explicado anteriormente, switch-case sirve para tomar decisiones en función de unas determinadas condiciones, para ejecutar unas sentencias u otras. Entre estas 24 líneas está hechas por si el jugador elige que quiere jugar contra el ordenador.

-Línea 60: De ahí “case 1”, (opción nº1). Una vez elegido la primera opción, el programa muestra todos los personajes con los que puede jugar, para así poder elegirlo.

-Línea 61: Se abre un bucle while, y como está indicado en el programa el nº1 indica el número de rondas que tiene cada partida.

-Líneas 62-70: El programa le pide al jugador de elegir un número, este está asignado a una figura, que quiere utilizar en la ronda. Abrimos un bucle for,( como se ha explicado anteriormente para hacer programas con tareas repetitivas o tomar decisiones, tareas que necesitan las denominadas sentencias de control, que son palabras claves de C que pueden alterar el ciclo de ejecución secuencial del programa. En este caso for para abrir un bucle para que el número elegido por el jugador sea asignado a una de las figuras del juego. Una vez acabado el bucle for, el número elegido por el jugador se guarda, “J1=numero” para después ser utilizado.

-Líneas 72-79: En estás líneas del “case 1” le toca al ordenador elegir una figura. El programa utiliza srand(…) para inicializar la semilla aleatoria y rand() para generar los números aleatorios entre1 y 16. La semilla de generación de números aleatorios se inicializa solo una vez, al principio del programa. Una vez el programa genera un número aleatorio, este se asocia a una figura. Entonces el programa utiliza el mismio bucle for que para el jugador, J1. Una vez acabado el bucle for, el número seleccionado por el ordenador, J2, se guarda para después

-Líneas 81-83 y línea 84:

-Estás líneas muestran las elecciones del J1 y J2, (jugador y ordenador).

-La sentencia break interrumpe un bucle en el momento en el que se ejecuta y continua el programa por la primera línea fuera del bucle.

**Líneas 86-219:**

Estas líneas sirven si el jugador elige la opción 2, “ranking”.

-Línea 86: La línea declara que se ha elegido la opción 2 y el programa lo muestra, con printf.

-Línea 87: La función fopen() abre un fichero para leer o escribir en él. Formalmente, su prototipo es el siguiente: FILE \*fopen (const char\*nombre, const char \*modo). El primer parámetro de la función es el nombre del fichero, el cuál tiene que respetar las normas del sistema operativo en el que se ejecute el programa. En los ejemplos se ha utilizado el formato de DOS (Disk Operating System), que es válido para la mayoría de los sistemas operativos, como Windows o UNIX. En DOS, el nombre del archivo puede tener de 1 a 8 caracteres y la extensión de 0 a 3. Sin embargo, en los ejemplos sólo se utiliza la extensión .txt, de manera que los archivos generados por los programas pueden abrirse con cualquier editor de texto. El segundo parámetro de la función fopen() indica el modo en el que se va a abrir el fichero, que puede ser para leer, escribir o añadir información (“r”).

Si no se produce un error al abrir el fichero, el puntero devuelto por fopen() se asigna a un puntero de tipo FILE que debemos haber declarado previamente en el programa. En este caso: FILE \*fentrada; fentrada=fopen(“entrada.txt", "r"); La primera instrucción declara un puntero de tipo FILE llamado “fentrada” y la segunda le asigna el puntero devuelto por la función fopen(), que en este ejemplo intenta abrir un fichero llamado entrada.txt para leer su contenido. Una vez realizada esta asignación, para referirnos al fichero (entrada.txt) utilizaremos el puntero asociado al mismo (fentrada).

-Líneas 89-91: Se introducen tres variables, dato\_ganar, dato\_empatar y dato\_perder. Las tres están inicializadas a cero.

-Líneas 94-102: Estas líneas son para abrir el fichero. Y como hemos dicho con anterioridad, el fopen de la función fichero es fentrada, intoducido en la línea 94,

-Línea 103: Esta línea sirve para leer el fichero, con el comando “fscanf” pasamos los datos del fichero a la memoria.

-Línea 104: Está línea del programa cierra el fichero con “fclose”, (la función fclose() cierra un fichero abierto previamente con fopen(). Su prototipo es el siguiente: int fclose (FILE \*fentrada); El argumento que hay que pasarle a la función es el puntero de tipo FILE que apunta al fichero. La función devuelve 0 si el fichero se cierra correctamente ..

-Líneas 109-199: En estas 89 líneas se escriben todas las condiciones para ganar, perder o empatar que tienes contra el ordenador, en función de que número elijas,(cada número del 1 al 15 tiene asociado una figura/avatar). Para que después, en función de si ganas pierdes o empates, estos datos se registren en el Ranking.

-Líneas 202-208: Estas líneas se abre el fichero para escritura.

-Línea 211: Esta función, “fprintf”, permite leer datos con formato y es idéntica a la función “printf()” salvo porque tienen un parámetro adicional, situado antes de la cadena de control que recibe el puntero asociado al fichero en el que se pretende leer la memoria de nuestro fichero.

-Línea 215: : Está línea del programa cierra el fichero con “fclose”, definido anteriormente.

-Líneas 217-219: Estos tres “printf” muestran el ranking del jugador, J1, es decir el número de partidas ganadas, empatadas y perdidas por el jugador contra el ordenador.

-Línea 220: La sentencia break interrumpe un bucle en el momento en el que se ejecuta y continua el programa por la primera línea fuera del bucle.

-**Líneas 223-225**:

Estas líneas sirven si el jugador elige la opción 2, “Reglas”, y se quiere saber como funciona el Ultimatum.

-Línea 223: La línea declara que se ha elegido la opción 3, y el programa muestra las reglas,(todas las figuras etc.), con “printf”.

-Línea 224La sentencia break interrumpe un bucle en el momento en el que se ejecuta y continua el programa por la primera línea fuera del bucle.

-**Líneas 227-229**:

Estas líneas sirven si el jugador decide salir del juego y elige la opción 4.

-Líneas 227-229: Este último comando indica que se salga del juego, si se el jugador escoge la opción 4, como se dice en la línea 227.