#include <iostream> // Entrada/salida

#include <array> // Arreglos

#include <cstdlib> // Random

#include <ctime> // Inicializar random

#include <cctype> // Funciones para caracteres

#include <string> // Funciones para cadenas

using namespace std;

const int ELEMENTOS = 25;

// Algunos de las franquicias de videojuegos mas vendidas

array<string, ELEMENTOS> franquicias = {

"Assassins Creed", "Battlefield", "Call of Duty",

"Diablo", "Dragon Quest", "Fallout", "Final Fantasy",

"Gran Turismo", "Grand Theft Auto", "Halo", "Just Dance",

"Metal Gear", "Minecraft", "Need for Speed", "Resident Evil",

"Star Wars", "Street Fighter", "Super Mario", "Super Smash Bros",

"Tetris", "The Elder Scrolls", "The Legend of Zelda",

"The Sims", "Tomb Raider", "Uncharted"

};

// Elige al azar uno de los elementos del arreglo

string elegirElemento(array<string, ELEMENTOS> &arr);

// Inicializa la palabra del jugador con guiones bajos

// Devuelve la cantidad de letras a adivinar en la palabra

int inicializarRespuesta(char respuesta[], string palabra);

// Solicita al usuario que ingrese una letra

// y la devuelve a la función desde donde se invoca

char pedirLetra(char respuesta[]);

// Valida si la letra ingresada es parte de la palabra

int validarLetra(char respuesta[], string palabra, char letra);

// Dibuja la figura de un ahorcado

void dibujarAhorcado(int errores, int letrasRestantes, string palabra);

// Jugar al ahorcado

void jugar();

// Elegir una opcion de menu

int elegirOpcion();

// Realizar busqueda binaria en un arreglo

int busquedaBinaria(array<string, ELEMENTOS> &arr, string valor);

// Busca una franquicia ingresada por el usuario en el arreglo de franquicias

void buscarFranquicia();

int main() {

int opcion;

opcion = elegirOpcion();

while (opcion != 3) {

switch (opcion) {

case 1:

jugar();

break;

case 2:

buscarFranquicia();

break;

}

cout << endl;

opcion = elegirOpcion();

}

cout << "Que tengas un buen dia." << endl;

return 0;

}

int elegirOpcion() {

int opcion;

cout << "Que desea hacer?" << endl;

cout << "1- Jugar al ahorcado" << endl;

cout << "2- Buscar mi franquicia favorita" << endl;

cout << "3- Salir" << endl;

cout << "Su eleccion: ";

cin >> opcion;

while (cin.fail() || !(opcion >= 1 && opcion <= 3)) {

cin.clear();

cin.ignore(1000, '\n');

cout << "Opcion no valida. " << endl << " Su eleccion: ";

cin >> opcion;

}

cout << endl;

return opcion;

}

void jugar() {

string palabra = elegirElemento(franquicias);

char respuesta[25];

char letra;

int letrasRestantes = inicializarRespuesta(respuesta, palabra);

int intentosFallidos = 0;

int letrasCorrectas = 0;

cout << "Jugar al ahorcado" << endl;

while (letrasRestantes > 0 && intentosFallidos < 7) {

letra = pedirLetra(respuesta);

letrasCorrectas = validarLetra(respuesta, palabra, letra);

if (letrasCorrectas > 0) {

letrasRestantes -= letrasCorrectas;

}

else {

intentosFallidos++;

}

dibujarAhorcado(intentosFallidos, letrasRestantes, palabra);

}

}

// Inicializa la palabra del jugador con guiones bajos

// Devuelve la cantidad de letras a adivinar en la palabra

int inicializarRespuesta(char respuesta[], string palabra) {

int longitudPalabra = palabra.length();

int cantidadLetras = 0;

for (int i = 0; i < longitudPalabra; i++) {

char caracter = palabra.at(i);

if (isalpha(caracter)) {

respuesta[i] = '\_';

cantidadLetras++;

}

else {

respuesta[i] = caracter;

}

}

respuesta[longitudPalabra] = '\0'; // caracter de fin de cadena

return cantidadLetras;

}

// Elige al azar uno de los elementos del arreglo

string elegirElemento(array<string, ELEMENTOS> &arr) {

// variable estática, "recuerda" su valor

// se usa para plantar la semilla en el generador

// de numeros aleatorios solo una vez

static bool semillaPlantada = false;

if (!semillaPlantada) {

srand(time(NULL));

semillaPlantada = true;

}

// Genera un número aleatorio entre 0 y 24

// y devuelve el elemento en ese indice

int indice = rand() % arr.size();

return arr[indice];

}

char pedirLetra(char respuesta[]) {

// Escriba una función que lea una letra y la devuelva a la función main

string frase = respuesta;

char letra;

cout << "Frase: " << frase << endl;

cout << "Ingrese una letra: ";

cin >> letra;

while (cin.fail() ) {

cin.clear();

cin.ignore(1000, '\n');

cout << "Letra no valida. " << endl << " Ingrese una letra: ";

cin >> letra;

}

cout << endl;

return letra;

}

int validarLetra(char respuesta[], string palabra, char letra) {

int veces = 0;

for (int i = 0; i < palabra.length(); i++) {

if (tolower(palabra.at(i)) == tolower(letra) &&

respuesta[i] != palabra.at(i)) {

respuesta[i] = palabra.at(i);

veces++;

}

}

return veces;

}

void dibujarAhorcado(int errores, int letrasRestantes, string palabra) {

cout << "Intentos fallidos: " << errores << endl;

cout << " \_\_\_\_\_" << endl;

cout << " | |" << endl;

cout << " | " << (errores > 0 ? "o" : "") << endl;

cout << " | " << (errores > 1 ? "/" : "") << (errores > 3 ? "|"

: " ") << (errores > 2 ? "\\" : "") << endl;

cout << " | " << (errores > 4 ? "|" : "") << endl;

cout << " | " << (errores > 5 ? "/ " : " ") << (errores > 6 ? "\\"

: "") << endl;

cout << " \_|\_" << endl;

cout << "| |\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_|" << endl << endl;

if (errores > 6) {

cout << "Te has ahorcado." << endl;

cout << "La franquicia era " << palabra << endl;

}

else if (letrasRestantes == 0) {

cout << "Felicidades! Has adivinado." << endl;

}

}

int busquedaBinaria(array<string, ELEMENTOS> &arr, string valor) {

// Implemente la función búsqueda binaria, para devolver el índice en

// el que se encuentra el valor o -1 si este no existe en el arreglo

int inf = 0;

int sup = (int) arr.size() - 1;

int med;

int indice = -1;

while (inf <= sup){

med = inf + (sup - inf) / 2;

if (valor < arr[med]) sup = med -1;

else if (valor > arr[med]) inf = med +1;

else{

indice = med;

break;

}

}

return indice;

}

void buscarFranquicia() {

string franquicia;

int posicion;

cin.clear();

cin.ignore(1000, '\n');

cout << "Buscar una franquicia en el top 25 de ventas" << endl;

cout << "Nombre de la franquicia: ";

getline(cin, franquicia);

posicion = busquedaBinaria(franquicias, franquicia);

if (posicion != -1)

cout << "Genial! Su franquicia esta en el top 25" << endl;

else

cout << "Su franquicia no es tan comercial, sus gustos son refinados." << endl;

}