Technical Overview

	Functions	Description
User-defined Functions	arrange_scores()	Sorts scores in descending order and rewrites them into the file.
	score(new_name, new_score)	Adds player names and scores based on certain conditions.
built-in functions	open(the file name , mode)	Opens the file in different modes such as r+ , w ,r ,and a+.
	close()	Close the file.

	readlines()
	read()
		Reads all lines from the file.

	writelines()	Writes a sequence of strings to a file.
	seek(0)	Sets the file's current position.
	split()	Splits a string into a list using a delimiter.
	strip()	Removes leading and trailing characters (by default, whitespace characters) from a string.
	lstrip()	Removes leading characters from a string.
	write():	Writes a string to a file.
	isalpha()	Checks if all characters in a string are alphabetic.
	isnumeric()	Checks if all characters in a string are numeric.
	input()	Reads a line from input, converts it to a string, and returns that.
	exit():	Exits from the Python script.
	print()	Prints the given object(s) to the standard output.
	sort()	Sort list ascending, and with descending order if "reserve" assign to true.
	replace()	Replace specific value with another specific value.
	append()	Add values to the list.

	variables	Description
In User-defined Functions	lined	Store the file data with excluding the header.
	header	Stores the header of the file.
	lines	Contains the lines read from the file to write the sorted scores lines with header.
	contents	Read the content of the file to check if it's empty. If the file is empty, add the header.
	parts	Split the lines by ‘|’ (User-defined Function and outside it).
	name	Contain the first elements in parts with remove the whitespace.
	new_name_stripped	Remove the new_name whitespace from the first only.
	option	Stores the option number user entered
	name	Store the name provided by the user to add to the file.
	user_score	Store the score provided by the user to add to the file.
	search_name	Stores the name provided by the user when searching for scores to update.
	update_name	Stores the name provided by the user to update the score.
	score_exists1	Stores a boolean to indicate if player's name exists during the update operation
	scored	A list to store lines that are not updated due not equal name in file or add to updated_lines then write in file
	updated_lines	stores the lines that are updated with new scores during the update operation.
	found	Stores a boolean to indicate if the player's name was found.
	lines	Read the lines from the file to splitting them, and check if the entered update_name exists in order to update the score.
	new_score	Stores the new score inputted by the user when updating to replace with old score.

data structures	Description
lists	To specify the contents of the file to sort , split and strip it and make empty lists to put the update score.
conditional statements	Used to handle the name if not found or duplicate name beside if score are not with numbers only.
loops	While used to handle display option menu until user write 4 (exit) for used to check all lines and split them
strings	used for displaying messages and prompts to the user.
int	used for displaying Integers and prompts to the user.
Boolean	Used to indicate if the player's name was found during the search and if player's name exists during the update operation