CLINICAL IMPRESSION MACHINE PROJECT

SUBMITTED BY: NIÑO, JOHN KOVIE L. AND LARRAQUEL, REIGN ELAIZA D.

char userType()

It asks the user to enter a choice of either D, P, or E. If the user enters anything other than D, P, or E, the user is asked to try again.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User chose D for Doctor	choice = 'D'	return 'D'	return 'D'	P
2	User chose P for Patient	choice = 'P'	return 'P'	return 'P'	P
3	User input not in the choices	choice = 'y'	output: "Invalid choice. Please try again." *prompts user to input again*	output: "Invalid choice. Please try again." *prompts user to input again*	P

char doctorMenu()

It's a function that displays a menu for the doctor to use, and returns the doctor's choice.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User inputs not in the choices	choice = 'x'	output: "Invalid choice. Please try again." *prompts user to input again*	output: "Invalid choice. Please try again." *prompts user to input again*	P
2	User inputs correct choice	choice = 'U'	return U;	return U;	P

void assignSymptoms(pairImpression* masterListImpression, pairSymptom* masterListSymptom, int impressionIndex)

It asks the user for the number of symptoms present in the case, then asks the user to input the corresponding number of each symptom

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User inputs n of symptoms within the current amount of symptoms	presentSymptoms = 8	8 number of symptoms were assigned successfully	8 number of symptoms were assigned successfully	P
2	User inputs n of symptoms greater than the current amount of symptoms	presentSymptoms = 21	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	P
3	User inputs n of symptoms where n is negative or 0	presentSymptoms = -4	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	P

void inputSymptoms(pairSymptom* masterListSymptom)

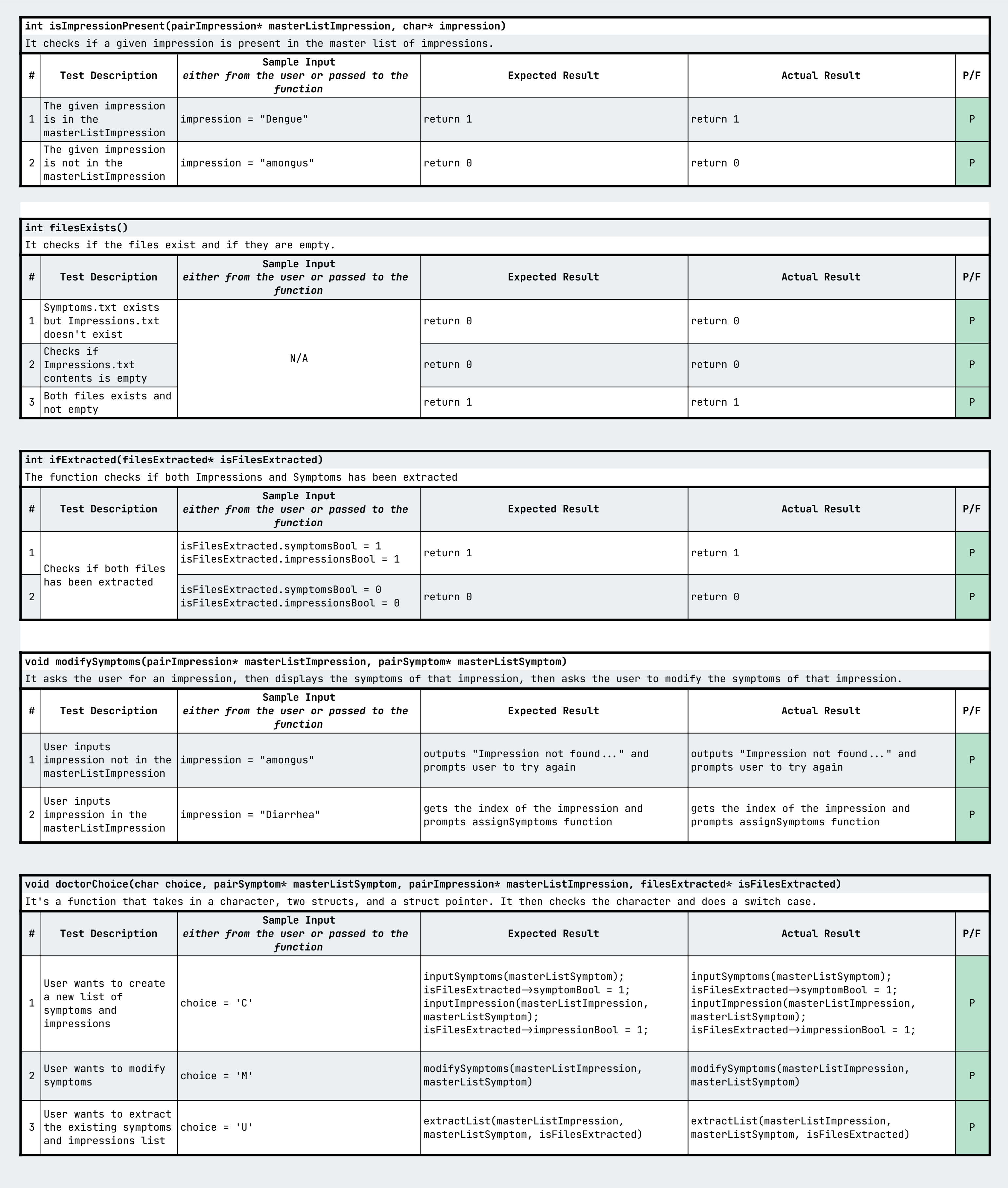
It asks the user for the number of symptoms they want to consider, then asks for the name of each symptom and its corresponding question.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User inputs n of symptoms within the maximum amount of symptoms	noSymptoms = 18	18 symptoms will be prompted by the program	18 symptoms will be prompted by the program	P
2	User inputs n of symptoms greater than the maximum amount of symptoms	noSymptoms = 34	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	P
3	User inputs n of symptoms where n is negative or 0	noSymptoms = 0	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	outputs "Invalid number of symptoms. Please try again." *prompts user to input again*	P

void inputImpression(pairImpression* masterListImpression, pairSymptom* masterListSymptom)

This function is used to gather information from the Doctor user, the name of each impression and the symptoms that are under it.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User inputs n of symptoms within the maximum amount of impressions	noImpression = 11	11 impressions will be prompted by the program	11 impressions will be prompted by the program	P
2	User inputs n of symptoms greater than the maximum amount of impressions	noImpression = 24	outputs "Invalid number of impressions. Please try again." *prompts user to input again*	outputs "Invalid number of impressions. Please try again." *prompts user to input again*	P
3	User inputs n of impressions where n is negative	noImpression = -23	outputs "Invalid number of impressions. Please try again." *prompts user to input again*	outputs "Invalid number of impressions. Please try again." *prompts user to input again*	P



int isImpressionPresent(pairImpression* masterListImpression, char* impression)

It checks if a given impression is present in the master list of impressions.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	The given impression is in the masterListImpression	impression = "Dengue"	return 1	return 1	P
2	The given impression is not in the masterListImpression	impression = "amongus"	return 0	return 0	P

int filesExists()

It checks if the files exist and if they are empty.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	Symptoms.txt exists but Impressions.txt doesn't exist	N/A	return 0	return 0	P
2	Checks if Impressions.txt contents is empty		return 0	return 0	P
3	Both files exists and not empty		return 1	return 1	P

int ifExtracted(filesExtracted* isFilesExtracted)

The function checks if both Impressions and Symptoms has been extracted

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	Checks if both files has been extracted	isFilesExtracted.symptomsBool = 1 isFilesExtracted.impressionsBool = 1	return 1	return 1	P
2		isFilesExtracted.symptomsBool = 0 isFilesExtracted.impressionsBool = 0	return 0	return 0	P

void modifySymptoms(pairImpression* masterListImpression, pairSymptom* masterListSymptom)

It asks the user for an impression, then displays the symptoms of that impression, then asks the user to modify the symptoms of that impression.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F
1	User inputs impression not in the masterListImpression	impression = "amongus"	outputs "Impression not found..." and prompts user to try again	outputs "Impression not found..." and prompts user to try again	P
2	User inputs impression in the masterListImpression	impression = "Diarrhea"	gets the index of the impression and prompts assignSymptoms function	gets the index of the impression and prompts assignSymptoms function	P

void doctorChoice(char choice, pairSymptom* masterListSymptom, pairImpression* masterListImpression, filesExtracted* isFilesExtracted)

It's a function that takes in a character, two structs, and a struct pointer. It then checks the character and does a switch case.

#	Test Description	Sample Input *either from the user or passed to the function*	Expected Result	Actual Result	P/F

1	User wants to create a new list of symptoms and impressions	choice = 'C'	inputSymptoms(masterListSymptom);
			isFilesExtracted->symptomBool = 1;
			inputImpression(masterListImpression, masterListSymptom);
			isFilesExtracted->impressionBool = 1;
				inputSymptoms(masterListSymptom);
				isFilesExtracted->symptomBool = 1;
				inputImpression(masterListImpression, masterListSymptom);
				isFilesExtracted->impressionBool = 1;
					P

2	User wants to modify symptoms	choice = 'M'	modifySymptoms(masterListImpression, masterListSymptom)	modifySymptoms(masterListImpression, masterListSymptom)	P
3	User wants to extract the existing symptoms and impressions list	choice = 'U'	extractList(masterListImpression, masterListSymptom, isFilesExtracted)	extractList(masterListImpression, masterListSymptom, isFilesExtracted)	P