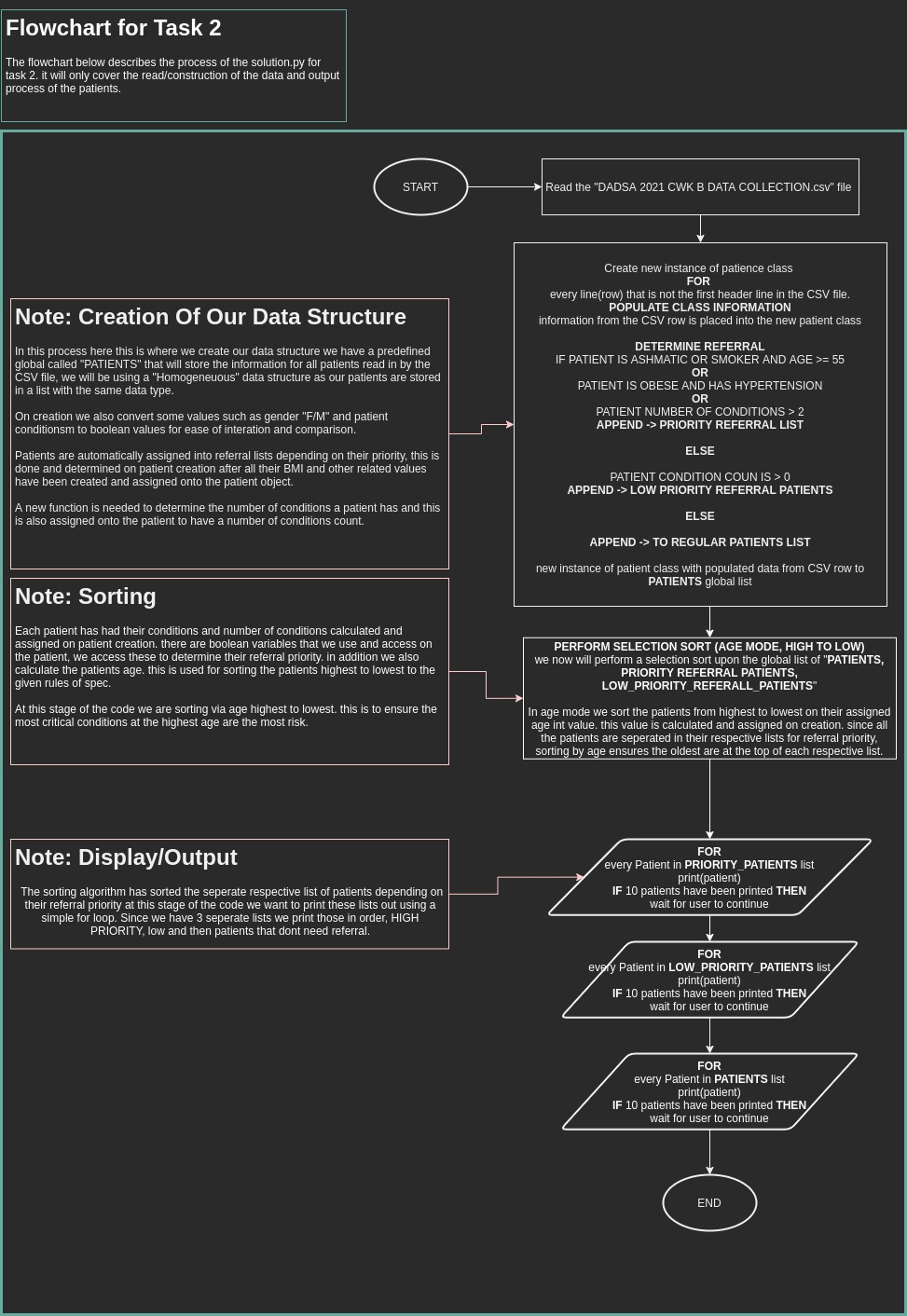
# FLOWCHART DESIGN DIAGRAM FOR TASK 2



In the flowchart diagram above you can see my design for task 2. In this example I am showing the general outline of how the solution.py should perform and react.

# PSEUDO CODE

## Read Datafile



In the provided text above you can see my pseudo code for task 2. In this example I am showing the read datafile method. This segment of code is responsible for reading the CSV file and converting it into a homogenous data structure. Each row of the CSV file will have its column values passed into a new instance of patient class. In addition, from the previous task this method has a new segment of code responsible for adding patients to different referral lists. These lists are then sorted by highest age to lowest.

## Patient Class



In the provided text above you can see my pseudo code for task 2. In this example I am showing the patient class. This class is responsible for holding all data about our patient. In addition, to the previous task the changes that needed to be made for this class was calculating and holding the number of conditions a patient has. We also have a Boolean value assigned to the patient if they need a dietitian referral.

## Determine Conditions Function



In the provided text above you can see my pseudo code for task 2. In this example I am showing determine condition’s function. This function is responsible for calculating the number of conditions a patient has, this is achieved by using the rules given in the spec. a patient object is passed in and we access its assigned variables to make the determination.

## Sorting Algorithm: Selection



In the provided text above you can see my pseudo code for task 2. In this example I am showing the sorting algorithm used to sort the patients. In addition to the earlier task this sorting algorithm has been designed with some extra upgrades. I have implemented a new Boolean argument value used to switch this sorting algorithm to sort the given list of patients via age. This will be needed as in the spec we want to referrer patients that are the oldest and with the most critical conditions.

## Calculate Body Mass Index Function



In the provided text above you can see my pseudo code for task 2. In this example I am showing the calculate body mass index function. This function is responsible for calculating the BMI of a given patient object, the patient object will store their weight and height as a public variable. We access these variables to calculate the BMI with the given equation in the spec. the result is then assigned upon the patient object as a new value called BMI. From the previous task there was no need to change this function.

## Determine patient classification Function:



In the provided text above you can see my pseudo code for task 2. In this example I am showing the determine patient classification. This function is responsible for calculating and assigning a patient classification name. The patient object will have variables stored upon it that contain their body build and bmi. We access these variables to calculate the patient classification to the requirements of the spec. the classification name is then returned and assigned on the patient object.

## Calculate Age



In the provided text above you can see my pseudo code for task 2. In this example I am showing the calculate dob to age function. This function is responsible for calculating and assigning a patient age. The patient object will have a variable stored upon it that contain date of birth. We access this variable to calculate the patient age by finding the difference with date time now. The returned value is assigned and stored on the patient.