Pseudo Code:

Start

Include System Libraries

Include my Structures

Include function prototypes

Void Instructions(fstream, Changes)

Menu()

knwDinners(fstream)

addDinner(fstream, changes)

newDinner(string)

rmvDinner(fstream)

modify();

dlDinList(fstream)

randomDin(Dinners\*, int)

Dinners addDinList(Dinners, int, int)

Begin main body of program

Declare an initialize variables for main

Declare structures & array Structure

Do

Begin Switch

Call function menu() to display the menu and get user choice

Request user's menu choice

Choice equals case 1

Calls instructions() to Read in game instructions from a file and display to user

Choice equals case 2

Calls kwnDinners() to Read in dinners from a file, display to user

Choice equals case 3

Calls addDinner() to Add a dinner to the known dinners

Choice equals case 4

Call rmvDinner() to Remove a dinner from the known dinners

Choice equals case 5

Call dlDinner list to delete current list of dinners

And get the input for how many dinners on the new list

Choice equals case 6

Loops to fill the new dinner list

Call addDinList() to Request dinners for the new list through manual input

Choice equals case 7

Loops to fill the new dinner list with random meals

Call randomDin() to Request dinners for the list through random selection

Choice equals case 8

View the current dinner list

Choice equals 9

Export the current dinner list to a file to print/email

Choice equals anything else

Ends the program

While

choice isn't equal to 1-9

End program

void instructions(){

Pull in instructions from a file display them to user

}

short menu(){

Open file to diplay menu to user

Get input from user to make a decision

Return choice

}

void kwnDinners(){

Open file to display a current list of known dinners

}

void addDinner(){

Open file to Add a new dinner to your known dinners

Request name of recipe and needed ingredients from user

Saves input to variables adds new input and exports/updates list of known dinners

Calls newDinner() to do this step

}

void newDinner(){

Creates a new file with the added dinner name

Requests the ingredients from the user needed to make dinner

Exports those ingredients to a file that can then be accessed for new dinners

}

void rmvDinner(){

Used to remove a dinner from the known dinner list

Able to remove from specific categories

Calls modify() to remove specified dinner from the known dinner list

Uses file input and output to modify required fields

}

char modify(){

Called on by rmvDinner()

Opens file, pulls out the unwanted dinner

Then exports modified information back into the file w/o the unwanted dinner

}

void dlDinList(){

Deletes the current dinner list

Requests how many dinners will be wanted in the new dinner list

}

dinners addDinList(){

Loops until input needed has been achieved

Used to manually add dinners to the dinners list

}

void randomDin(){

Fills the dinner list with randomly chosen meals

Unfortunately manually added dinners cannot be accessed under the random function

Selects random meal

Loops until finished

}

END OF CODE