Warteg Arya Readme Documentation

This program is used to manage a restaurant menu using a linked list data structure in C programming language. The menu is stored in a file named "menu.txt" in the following format: dish_name#price#quantity. The program allows the user to insert new dishes to the menu, remove dishes from the menu, and view the menu.

Functions

struct Dish *createDish(char dish[], long long int prc, long long int quantity)
This function creates a new Dish with the provided name, price, and quantity and returns a pointer to it.

void insertLL(char dish[], long long int prc, long long int quantity)
This function creates a new dish using the createDish() function and inserts it into the linked list. The dishes are sorted in descending order by their quantity, so that the dishes with the highest quantity are at the top of the list. After inserting the new dish, the function updates the data in the "menu.txt" file.

void removeLL(char dish[])
This function removes a dish from the linked list based on its name. If the dish is not found in the linked list, the function prints an error message. After removing the dish, the function updates the data in the "menu.txt" file.

void readFromFileAndPushToList(const char *filename)
This function reads the menu data from the "menu.txt" file and creates a linked list of dishes. Each line in the file represents a dish in the format: dish_name#price#quantity. The function uses the insertLL() function to insert each dish into the linked list.

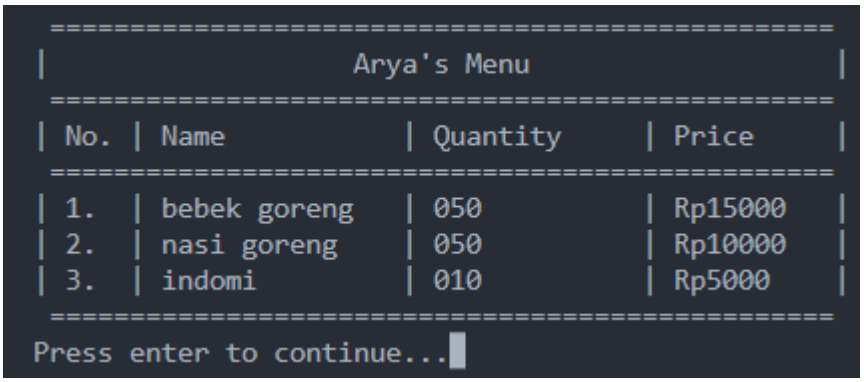
int searchMenu(char dish[])

This function searches for a dish in the linked list based on its name. If the dish is found, the function stores the dish's name, price, and quantity in the foundDish, foundPrc, and foundQty variables, respectively. If the dish is not found, the function returns a negative value.

void printLL()

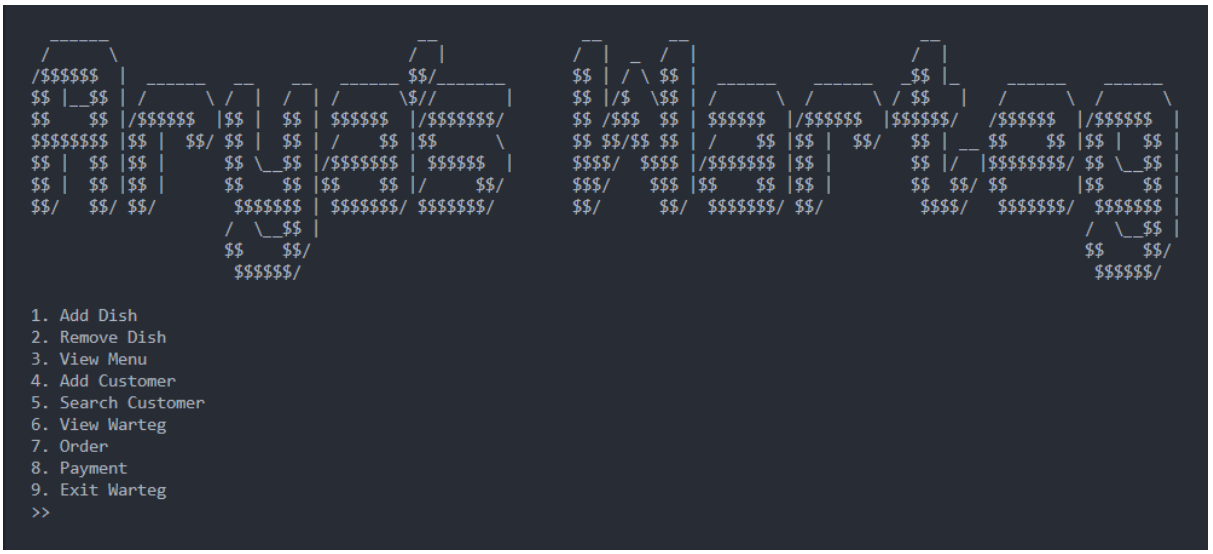
This function prints the menu items in a formatted table.

void viewMenu()



This function clears the screen and calls the printLL() function to display the menu items. It also waits for the user to press a key before continuing.

void mainMenu()



The mainMenu() function displays the main menu options and allows the user to select an option by inputting a number from 1 to 9. The input is

validated to ensure that the user inputs a valid number. The switch statement then executes the corresponding function based on the user's input.

unsigned long djb2(char name[])
This function implements the djb2 hash function which takes in a string and returns a hash value. The hash value is calculated using a loop which multiplies the previous hash value by 33 and then adds the current character of the string.

void linearProbing(int idx, char name[])
This function implements the linear probing technique which helps to resolve hash collisions. The function takes in an index and a string. It checks if the index in the hash table is already occupied or not. If it is occupied, it starts searching for the next available index in the hash table in a linear manner.

void insertCust(char name[])
This function inserts a customer into the hash table. It first calculates the hash value of the customer's name using the *djb2 function*. If the index corresponding to the hash value in the hash table is empty, it inserts the customer at that index. Otherwise, it uses *linearProbing* to find the next available index to insert the customer.

int searchCust(char name[])
This function searches for a customer in the hash table. It calculates the hash value of the customer's name using the djb2 function. If the index corresponding to the hash value in the hash table is empty, it returns -1. Otherwise, it searches for the customer by comparing the names of the customers at that index and the indices following it, until it finds the customer or reaches the starting index. If the customer is found, it returns the index in the hash table where the customer is stored. If the customer is not found, it returns -1.

void *insertOrder(int idx, char name[], long long int prc, long long int qty)

This function inserts an order into the linked list of orders of a customer. It creates a new order using the createDish function and then inserts it at the end of the linked list.

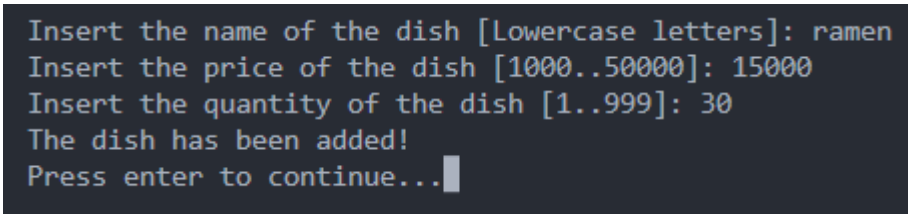
void removeCust(int idx)

This function removes a customer from the hash table. It first frees the memory allocated for the linked list of orders of the customer, and then frees the memory allocated for the customer itself. It also sets the corresponding index in the hash table to NULL.

void removeOrder(int idx)

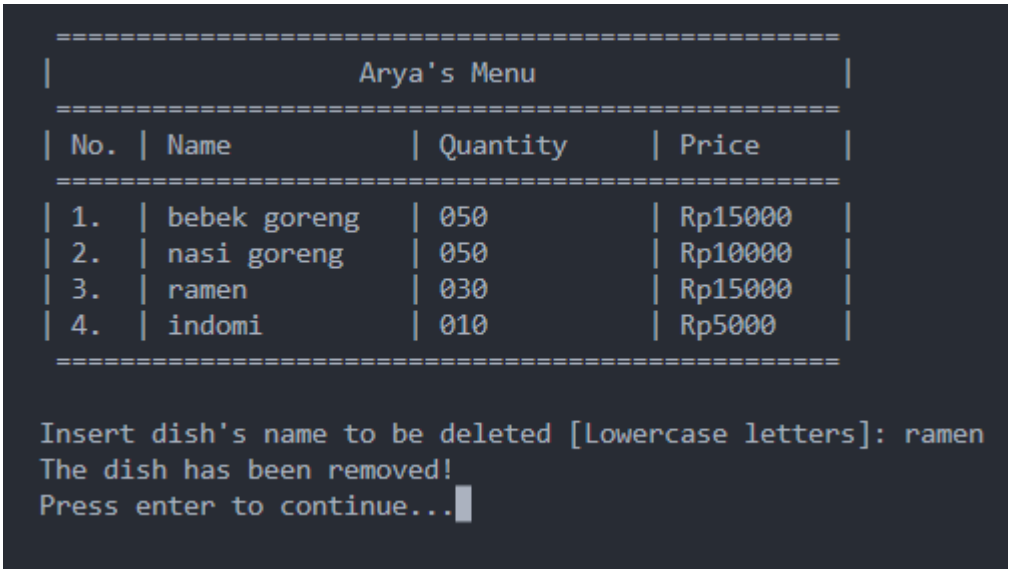
This function removes all orders of a customer. It frees the memory allocated for each order and sets the foodOrder pointer of the customer to NULL.

void addDish()



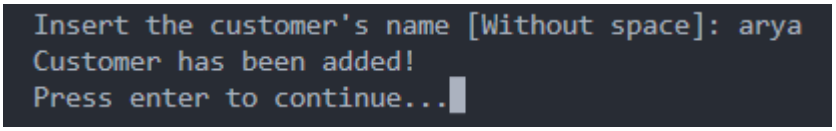
This function adds a new dish to the menu. It prompts the user to enter the name of the dish, its price, and its quantity. It checks the validity of the inputs and inserts the new dish into the linked list using the insertLL() function. It then displays a message indicating that the dish has been added, and prompts the user to press the enter key to continue.

void removeDish()



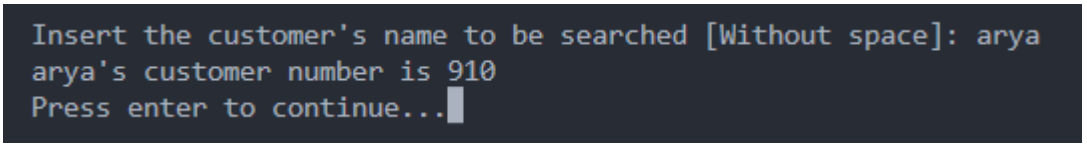
This function removes a dish from the menu. It displays the current menu using the printLL() function and prompts the user to enter the name of the dish to be deleted. It checks the validity of the input and removes the dish from the linked list using the removeLL() function. It then displays a message indicating that the dish has been deleted, and prompts the user to press the enter key to continue.

void addCustomers()



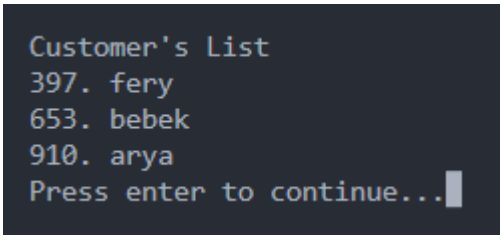
This function adds a new customer to the customer list. It prompts the user to enter the name of the customer and checks the validity of the input. It then inserts the new customer into the customer list using the insertCust() function. It displays a message indicating that the customer has been added, and prompts the user to press the enter key to continue.

void searchCostumer()



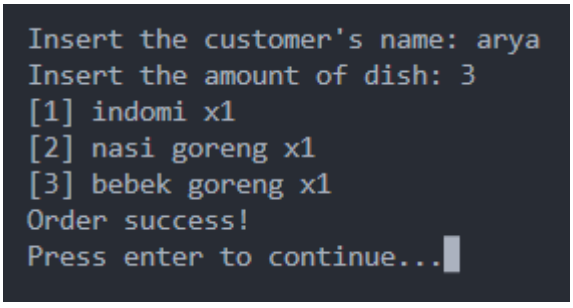
This function searches for a customer in the customer list. It prompts the user to enter the name of the customer to be searched, checks the validity of the input, and searches for the customer using the searchCust() function. If the customer is found, it displays the customer's number in the customer list. If the customer is not found, it displays a message indicating that the customer is not present. It then prompts the user to press the enter key to continue.

void viewWarteg()



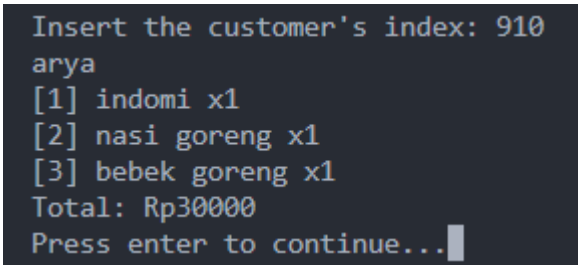
This function displays the list of customers in the customer list. It displays the customer's number and name using a for loop. If there are no customers in the list, it displays a message indicating that there are no customers and prompts the user to add a customer. It then prompts the user to press the enter key to continue.

void order()



This function allows a customer to place an order. It prompts the user to enter the customer's name and the name and quantity of the dish to be ordered. It checks the validity of the input and searches for the customer using the searchCust() function. If the customer is found, it adds the order to the customer's order list using the orderLL() function. If the customer is not found, it displays a message indicating that the customer is not present. It then prompts the user to press the enter key to continue.

void payment()



This function prompts the user to input the index of the customer and displays the order details and total payment amount. If the customer index is not found, an error message is displayed. Once the payment is made, the customer and their order are removed from the system.

Global Variables

int menuCounter
This variable keeps track of the number of dishes in the menu. It is incremented every time a new dish is added to the menu.

File Input/Output

The menu data is stored in a file named "menu.txt". The file is read when the program starts using the readFromFileAndPushToList() function, and it is updated every time a dish is inserted or removed using the insertLL() and removeLL() functions, respectively. The format of the data in the file is dish_name#price#quantity.

Data Structures

struct Dish

This is a structure that represents a dish in the menu. It contains the following fields:

char dish[255]: The name of the dish.
long long int prc: The price of the dish.
long long int quantity: The quantity of the dish.

struct Dish *prev: A pointer to the previous dish in the linked list.
struct Dish *next: A pointer to the next dish in the linked list.

struct Customers

This is a structure that holds the name of a customer and a pointer to a linked list of dishes that the customer has ordered. This structure is not used in the current program.

Main Function

The main() function reads the menu items from a file called "menu.txt" using the readFromFileAndPushToList() function and then calls the mainMenu() function to start the program.