1. CIRCULAR QUEUE:  To allot the room number to the customer.( the room number have been stored in circular queue and used in pop and push function)

DATA STRUCTURES USED FOR THE PROGRAM

1. STACK: To compute the restaurant bill (the prices of the food item were pushed in stack)
2. FILES:  The files restaurant.dat , customer.dat  and room.dat

* Room.dat- file to store the circular queue, front and rear of the queue after every allotting of room or room checkout
* Restaurant.dat- storing the customer detail and the details of the food ordered. So that it is easier to read at the time of bill display
* Customer.dat- storing customer details (first name,last name, email id, phone number, room number)

FUNCTIONS USED IN THE PROJECT

**Void push(int):**

The function is used to implement the circular queue, when the room is empty the room is inserted back in the queue using the push function. The room number that the needs to be pushed back in queue is passed as the argument

**Void pop()**

The function has been used to get the room number for the customer while booking for the hotel. The function returns the room number value from the circular queue and writes the ne value of queue, rear and front in the file.

**struct rest searchid(struct dish a[], int x)**

The function has been used to search the item that the customer has ordered corresponding to the food id (x) that the customer has entered. the function searches for the food item id (x) in the array of structure storing the dish name, prizes and food id and returns the food name and price of food by storing these details in r(struct rest).

**Void order(struct rest k)**

 The function has been used to place proper quantity of the food item selected. The customer is asked to enter the quantity and then the total price is computed. The total price computed is pushed in the stack.

**Void disp()**

The function has been used to display the total price to be payed by popping prices out from the stack and asking user to enter the details like- room number, first name and last name and storing these details in a file (restaurant.dat)

**Void namedisp()**

The function displays the customer on what order he has placed and how much he has paid. It gives him a date-wise overview of what all he had ordered during his stay and how much he has spent

**Void billdisp() and void lilmenu()**

The function displays the headlines for the restaurant menu like-food item name, price, quantity

**Void rdisp(struct rest x)**

The function displays the details of the order places stored in struct rest like- food name, food price, quantity etc.

**Void custdisp()**

The function displays the headlines for customer details

**Void crdisp(cust a)**

The function displays the details of the customer(last name, first name, email, date) in cust(struct ccdet).these details are read from the file using the function void detaildisp()

**Void bookroom(int day, int month, int year)**

The function has been used to book room(the room number have been stored in a circular queue) and enter customer details. the room number tis given by calling the pop function within this function.

**Void checkyear(int year)**

The function has been used to check if the booking year is a leap year or not

**int famount(int a, int b, int c, int x, int y, int z**)

the function has been used to calculate the total money to be paid by the customer for the stay in the hotel. The function takes booking date(day, month, year) and leaving date(day, month, year) as arguments

**void checkOut(int day, int month, int year)**

 the function has been used to push the empty room back into the queue after the customer leaves the room and also write it into the file. The function takes the leaving date(day, month, year) as parameters.

ALGORITHM FOR THE PROJECT

**Bookroom()**

* The booking date, month and year have been passed as arguments for the function,
* A pointer ‘a’of type struct cdet is declared and a file “customer.dat” is opened in append mode.
* The details corresponding to data members of struct cdet is entered, the status of room is changed to “STAYING” and the details are entered into the file.
* The value returned from the pop function [pop ()] is stored in a->rno
* The booking date, day and month are given the values of the function parameters respectively and leaving date, day and month are initialised as 0.
* Crdisp() function is called to display the details of the customer,.
* The details stored in the pointer ‘a’ are written in file “customer.dat”

**Checkyear()**

* The year of the leaving date is passed as the argument
* If year when divided by 4 and 400 leaves no remainder then year is a leap year and return 1 . if year is divisible by 100 and leaves no remainder then not a leap year and return 0.

**Famount ()**

* The booking date (day, month, year) and the leaving date(day, month, year) have been passed as arguments as a, b, c, x, y and z.
* An array “arr” is initialised to store the total number of days of each month of year (the year is assumed to be not a leap year)
* Days is a variable used to store x-a(numeric difference between booking and leaving day)
* The month of the booking date and leaving date is checked, if they are not the same then the value at the index position indicating the booking month is added to find the number of total days of stay.

Days =days+ arr [mm-1], mm is booking month

* To check whether the year is a leap year or not the check year function is called, if year is a leap year and booking month is February (y=2), then days++
* “t” variable is declared to store the total cost of staying by multiplying days with perday cost
* “t” is returned

**Checkout ()**

* The leaving date (day, month, year) have been passed as arguments
* The staying status of the room no entered by the user is read from the file (customer.dat ), if the status is “STAYING”, then the customer details are displayed (using function crdisp()) an the room is inserted in queue using ( push function) . the staying status of the room is changed to left and the details of the room are written in another file (temp.dat).
* If the room no entered by customer is empty, then simply write the detail is file

( temp.dat)

* The file customer.dat is removed and the new file (temp.dat) is renamed as (customer.dat)