PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH **Year:** 2024 – 25 **Semester:** I

Name: Abhishek Joshi **PRN: 123B1B150**

Department: Computer Engineering **Division:** C (C1)

Course: Data Structures Laboratory **Course Code:** BCE23PC02

Date:

Assignment – 9B

- **Aim:**

Implement a checkout system for a supermarket to efficiently manage customer queues during peak hours. The system should support the following operations using a circular queue data structure:

a. Customer arrival

b. Customer checkout

c. Close Checkout Counter

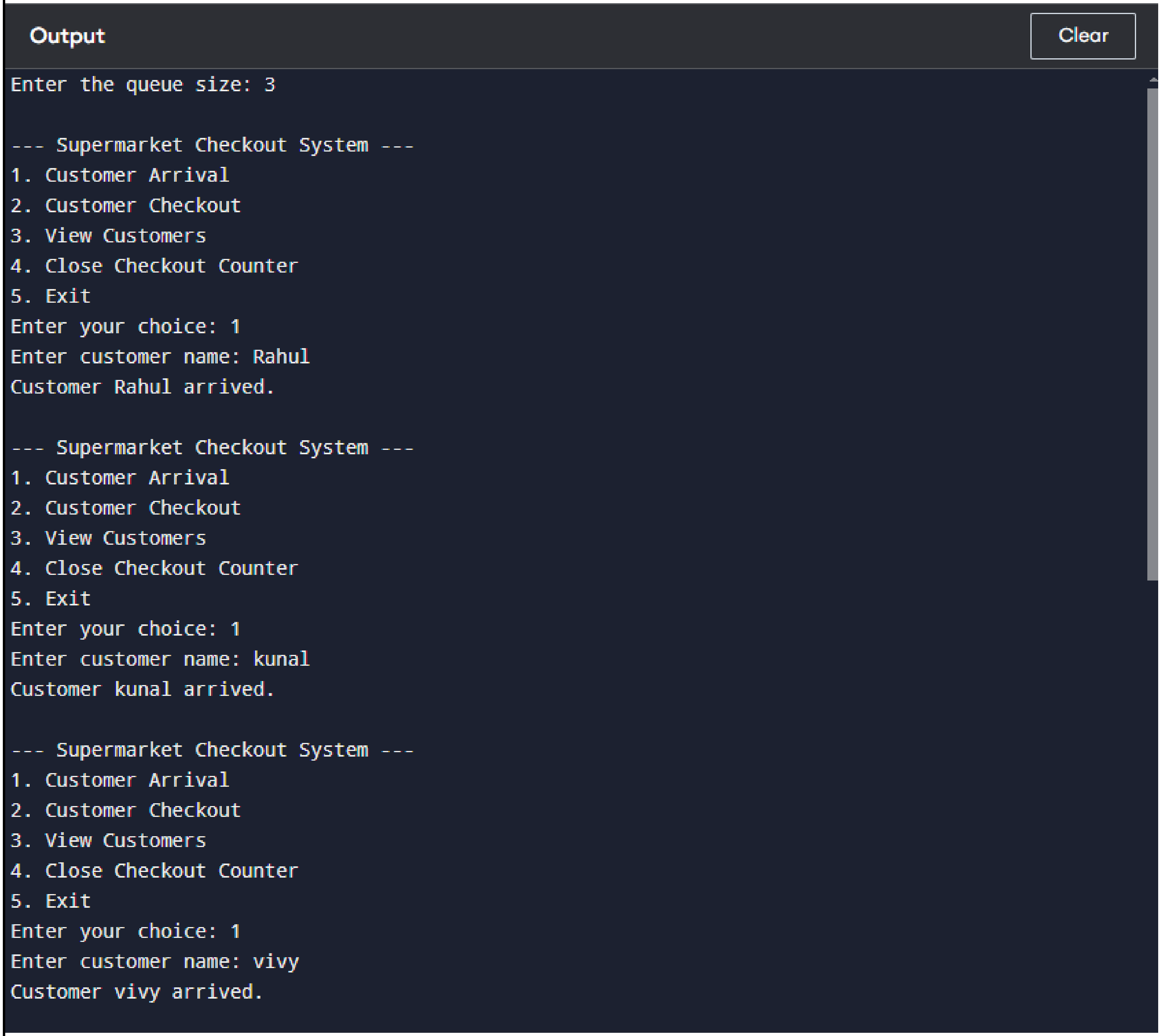
d. View customer

- **Source Code :**

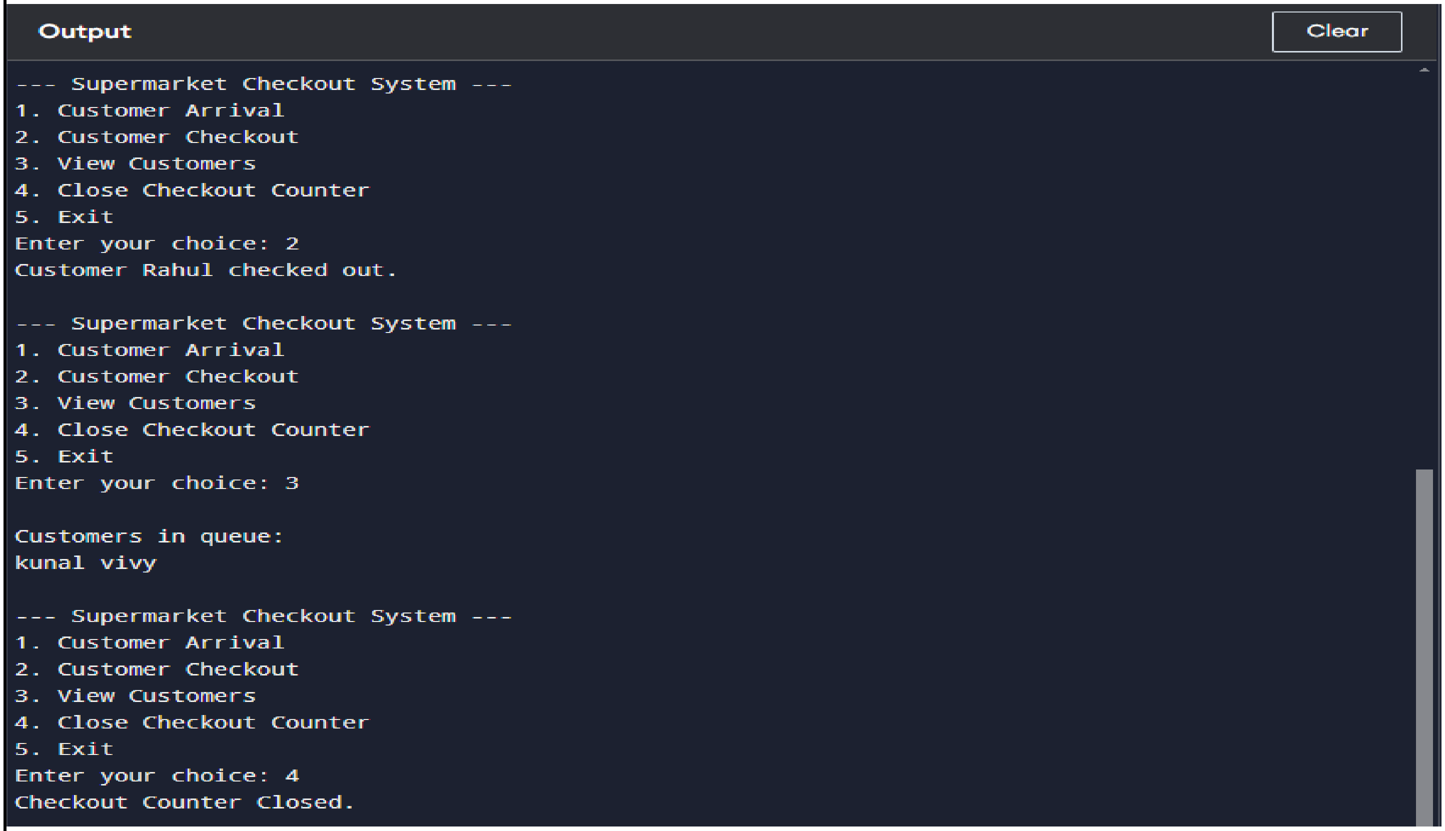
```
#include<iostream>
#include<string>
using namespace std;

class Queue{
   int front;
   int  rear;
   int size;
   string *arr;
public:
   Queue(int  s){
      front = rear = -1;
      size = s;
      arr = new string[s];
   }
   void Customer_arrival(string name);
   string Customer_checkout();
   void View_customer();
   void Close_Checkout_Counter();
```

```
};

void Queue::Customer_arrival(string name){
    if ((front == 0 && rear == size-1) || ((rear + 1) % size == front)){
        cout << "\nQueue is Full"<<endl;
        return;
    }
    if (front == -1){
        front = rear = 0;
    } else {
        rear = (rear + 1) % size;
    }
    arr[rear] = name;
    cout << "Customer " << name << " arrived."<<endl;
}

string Queue::Customer_checkout(){
    if (front == -1){
        cout << "\nQueue is Empty"<<endl;
        return "";
    }
    string val = arr[front];
    cout << "Customer " << val << " checked out."<<endl;

    if (front == rear){
        front = rear = -1;
    } else {
        front = (front + 1) % size;
    }
    return val;
}

void Queue::View_customer(){
    if (front == -1){
        cout << "\nNo customers in queue."<<endl;
        return;
    }
    cout << "\nCustomers in queue:"<<endl;
    int i = front;
    while (true) {
        cout << arr[i] << " ";
        if (i == rear) break;
        i = (i + 1) % size;
    }
    cout << endl;
}

void Queue::Close_Checkout_Counter(){
    cout << "Checkout Counter Closed."<<endl;
    front = rear = -1;
}

int main(){
    int size;
    cout << "Enter the queue size: ";
    cin >> size;
    Queue q(size);

    int choice;
    string name;

    do {
        cout << "\n--- Supermarket Checkout System ---"<<endl;
        cout << "1. Customer Arrival"<<endl;
        cout << "2. Customer Checkout"<<endl;
        cout << "3. View Customers"<<endl;
        cout << "4. Close Checkout Counter"<<endl;
        cout << "5. Exit"<<endl;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter customer name: ";
                cin >> name;
                q.Customer_arrival(name);
                break;
            case 2:
                q.Customer_checkout();
                break;
            case 3:
                q.View_customer();
                break;
            case 4:
                q.Close_Checkout_Counter();
                break;
            case 5:
                cout << "Exiting system."<<endl;
                break;
            default:
                cout << "Invalid choice! Please try again."<<endl;
        }
    } while (choice != 5);
    return 0;
}
```

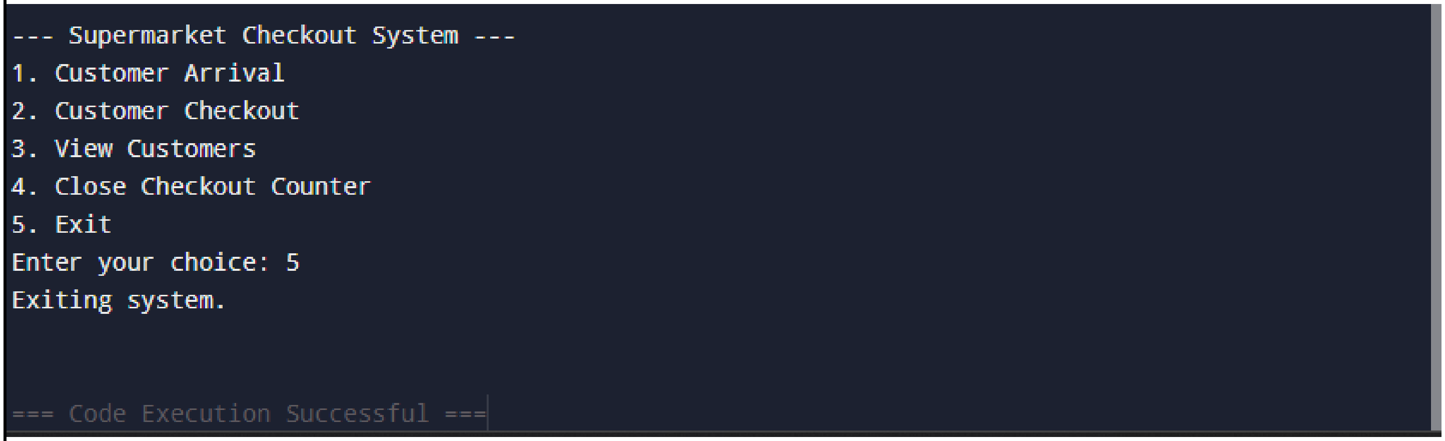
- **<u>Screen Shot of Output :</u>**



```
Enter the queue size: 3

--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 1
Enter customer name: Rahul
Customer Rahul arrived.

--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 1
Enter customer name: kunal
Customer kunal arrived.

--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 1
Enter customer name: vivy
Customer vivy arrived.
```



```
--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 2
Customer Rahul checked out.

--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 3

Customers in queue:
kunal vivy

--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 4
Checkout Counter Closed.
```



```
--- Supermarket Checkout System ---
1. Customer Arrival
2. Customer Checkout
3. View Customers
4. Close Checkout Counter
5. Exit
Enter your choice: 5
Exiting system.


=== Code Execution Successful ===
```

- **Conclusion:**

  In this assignment, we implemented a restaurant waitlist system using arrays to demonstrate queue operations for adding parties, seating them, and displaying the current waitlist. The program effectively handles basic queue functionalities while providing a user-friendly menu interface.