PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Au[redacted]onomous Institute)

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

Name: Sonawane Prachi Mahendra. [redacted] **PRN:** 124B2B018

Department : Computer Engineering **Division:** B

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

Date: 11/10/24

Assignment – 9

- **Aim:**

1. 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.

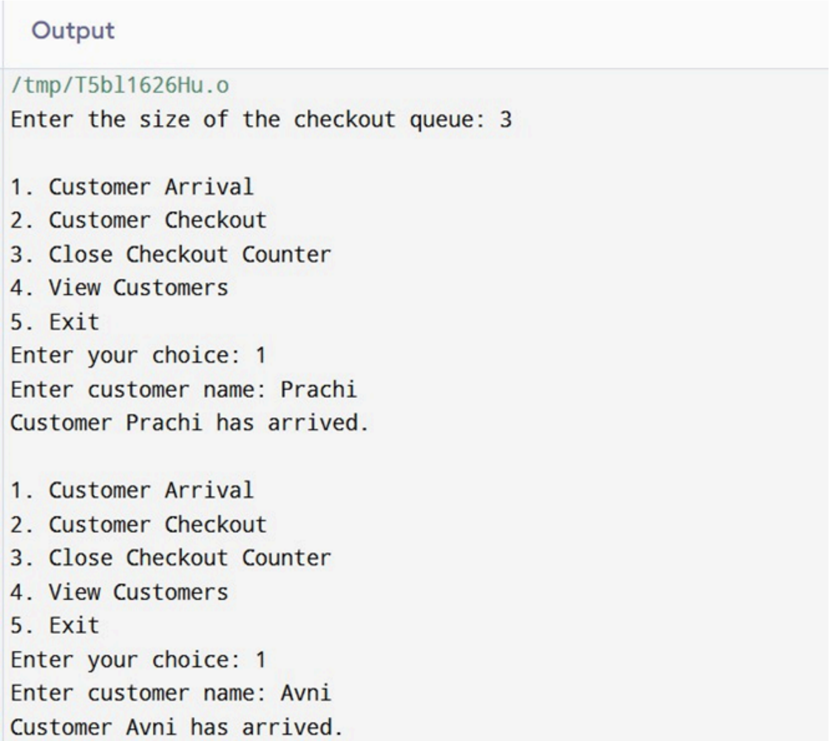
- **Source Code:**

```
#include <iostream>

#include <string>


using namespace std;

class CircularQueue { private:

    string* queue;

    int front, rear, capacity;
  public:

        CircularQueue(int size) {

        capacity = size;
    queue = new string[capacity]; front =
    rear = -1;

}




    ~CircularQueue() { delete[]
        queue;
}



void enqueue(string customer) {
    if ((rear + 1) % capacity == front) {
        cout << "Queue is full! Cannot add customer: " << customer << endl; return;

    }
        if (front == -1) { front =
            rear = 0;


    } else {
        rear = (rear + 1) % capacity;
    }
    queue[rear] = customer;
    cout << "Customer " << customer << " has arrived." << endl;
}
```

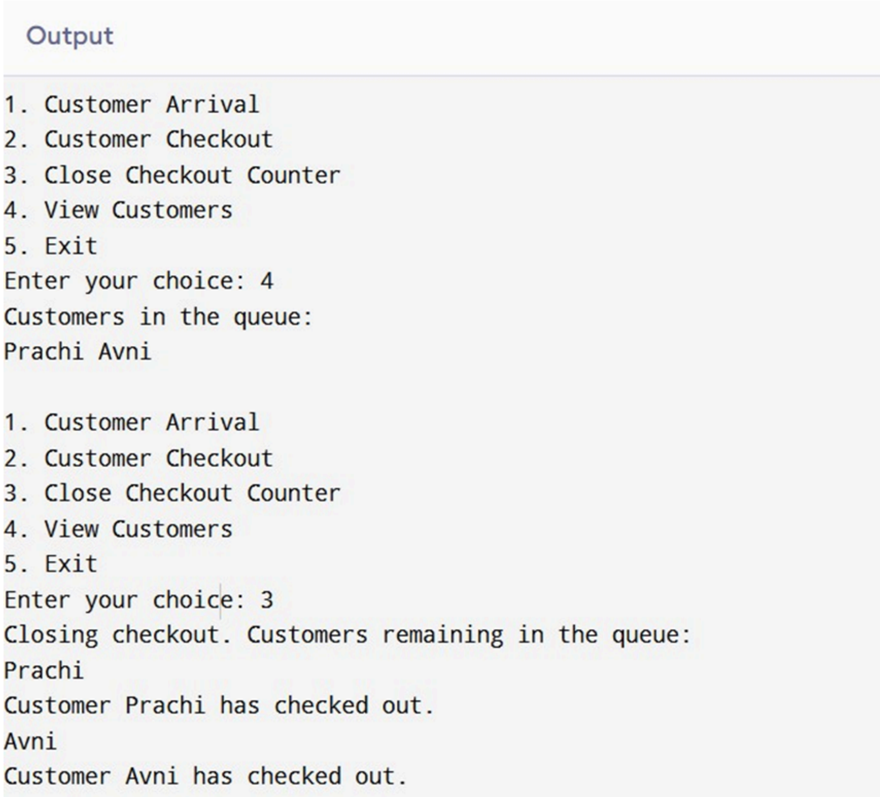
```
void dequeue() {
    if (front == -1) {
        cout << "No customers to checkout!" << endl; return;
    }
    cout << "Customer " << queue[front] << " has checked out." << endl; if (front
    == rear) {
        front = rear = -1;
    } else {
        front = (front + 1) % capacity;
    }
}



    void closeCheckout() { if
        (front == -1) {
        cout << "No customers in the queue to close the checkout." << endl; return;

    }

    cout << "Closing checkout. Customers remaining in the queue:" << endl; while (front
    != -1) {
        cout << queue[front] << endl; dequeue();


    }
}
```

```
        void viewQueue() { if
            (front == -1) {
            cout << "The queue is empty." << endl; return;
        }
        cout << "Customers in the queue:" << endl; int i =
        front;
        while (true) {
            cout << queue[i] << " "; if (i ==
            rear) break;
            i = (i + 1) % capacity;
        }
        cout << endl;
    }
};



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


    CircularQueue checkoutQueue(size);
    int choice; string
    customer;


    do {
        cout << "\n1. Customer Arrival\n2. Customer Checkout\n3. Close Checkout Counter\n4.
View Customers\n5. Exit\n";
        cout << "Enter your choice: "; cin >>
        choice;


            switch (choice) { case 1:
                cout << "Enter customer name: "; cin >>
                customer;
                checkoutQueue.enqueue(customer); break;
            case 2:
                checkoutQueue.dequeue(); break;
            case 3:
                checkoutQueue.closeCheckout(); break;
            case 4:
                checkoutQueue.viewQueue(); break;
            case 5:
                cout << "Exiting system." << endl; break;

            default:

                cout << "Invalid choice! Please try again." << endl;
        }
    } while (choice != 5);
    return 0;
}
```

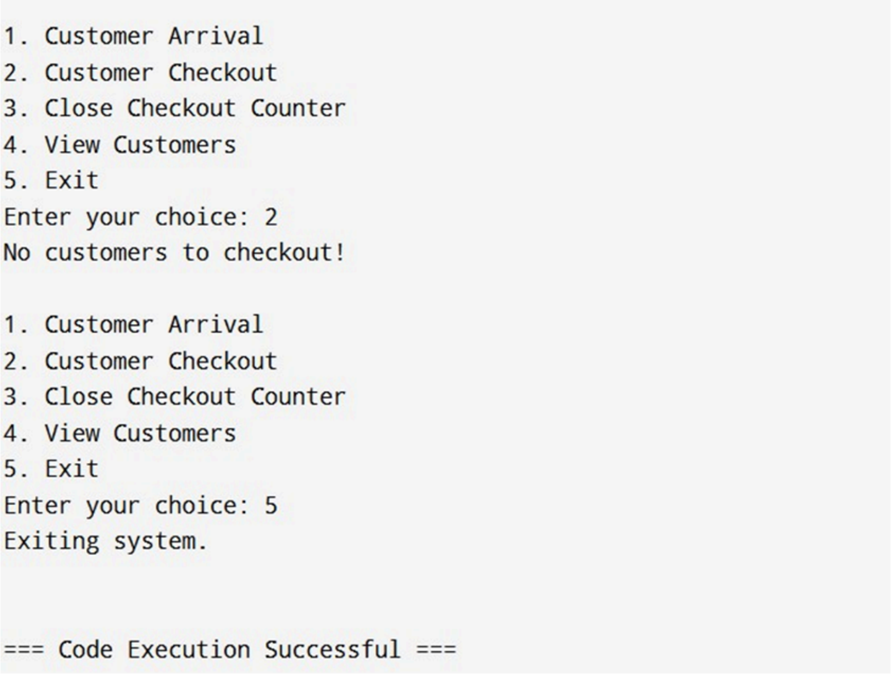
- **Screen shots of Output:**



```
/tmp/T5bl1626Hu.o
Enter the size of the checkout queue: 3

1. Customer Arrival
2. Customer Checkout
3. Close Checkout Counter
4. View Customers
5. Exit
Enter your choice: 1
Enter customer name: Prachi
Customer Prachi has arrived.

1. Customer Arrival
2. Customer Checkout
3. Close Checkout Counter
4. View Customers
5. Exit
Enter your choice: 1
Enter customer name: Avni
Customer Avni has arrived.
```



```
1. Customer Arrival
2. Customer Checkout
3. Close Checkout Counter
4. View Customers
5. Exit
Enter your choice: 4
Customers in the queue:
Prachi Avni

1. Customer Arrival
2. Customer Checkout
3. Close Checkout Counter
4. View Customers
5. Exit
Enter your choice: 3
Closing checkout. Customers remaining in the queue:
Prachi
Customer Prachi has checked out.
Avni
Customer Avni has checked out.
```



```
1. Customer Arrival
2. Customer Checkout
3. Close Checkout Counter
4. View Customers
5. Exit
Enter your choice: 2
No customers to checkout!

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


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

- **Conclusion:**

  Hence, we studied about Circular Queue and its operations.