Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
HingnaRoad,Wanadongri, Nagpur - 441110
NAAC A++
Ph.: 07104-237919,234623,329249,329250Fax:07104-232376,Website: www.ycce.edu

DepartmentofComputerTechnologyB.TechinComputerScienceandEngineering(IOT)

Visionofthe Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session 2025-2026

Vision: Build a clear, hands-on understandingof concurrent programming by modeling real-world order handling with threads and synchronization, fostering confidence to design correct, deadlock-free producer–consumer systems in operating environments.	**Mission:** Implement a two-thread restaurant workflow using mutexes and condition variables, validate safe coordination for exactly five orders, analyze timing and output for correctness, and reflect on improvements (incremental signaling, buffer generalization) to strengthen OS-concepts mastery.

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-si-lL easy to recall**
PEO2 PEO3	**Core Competence** **Breadth**	**E: Environment (Learning Environment)**	
PEO4	**Professionalism**	**P: Professionalism**	
PEO5	**Learning**	**C: Core Competence**	
	Environment	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program) **Keywords of POs:**
Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

"I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life." *to contribute to the development of cutting-edge technologies and Research.*

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

Name and Signature of Student and Date
(Signature and Date in Handwritten)

Session	**2025-26(ODD)**	**CourseName**	**Operating System**
Semester	**5**	**CourseCode**	**23IOT1504**
RollNo	**52**	**Name of Student**	**Parth Bhedurkar**

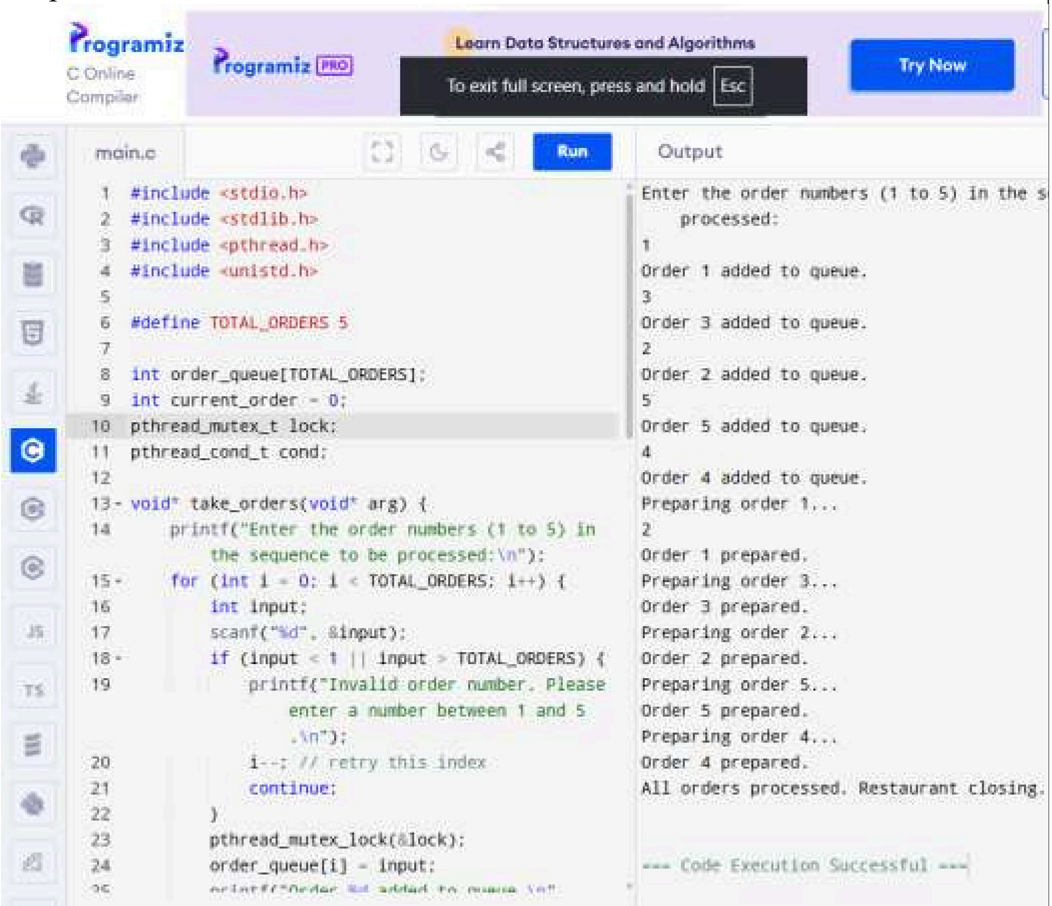
Practical Number	**3**
Course Outcome	• CO1: Use Linux toolchain and commands to compile, run, and observe thread behavior and system resource usage during the simulation, showing efficient use of the environment for OS experiments. CO2: Analyze • operating system functionalities by implementing threads, coordinating them with appropriate synchronization, and reasoning about scheduling/ordering of events in the order–prepare pipeline. CO3: Apply • synchronization primitives (e.g., mutexes/semaphores/condition variables) to ensure no deadlock or race conditions while both threads terminate cleanly after handling 5 orders. .
Aim	Restaurant Order System
Problem Definition	Create a thread that simulates taking orders and another that simulates preparing food. Both should terminate after handling 5 orders.
Theory (100 words)	The restaurant order system demonstrates the classic producer–consumer problem in operating systems using two threads that share a bounded buffer of orders and synchronize access to avoid races and deadlocks. One thread acts as the producer by taking customer orders and enqueuing them, while the other acts as the consumer by dequeuing and preparing food; mutual exclusion is enforced with a mutex, and progress is coordinated using counting semaphores or condition variables for "empty" and "full" slots in the queue. The producer must block when the queue is full, and the consumer must block when the queue is empty, ensuring correctness and preventing buffer overflows or underflows in a bounded-buffer setup. Proper ordering of wait/signal around the critical section guarantees that only one thread manipulates the shared queue at a time,

	eliminating race conditions and avoiding deadlock; the exercise typically terminates after processing a fixed number of items, here 5 orders, to validate clean shutdown and resource release in coordinated concurrency
Procedure and Execution (100 Words)	1. Initialize synchronization: create a mutex (lock) and a condition variable (cond) to protect and signal access to the shared counter orders. 2. Define limits: set MAX_ORDERS to 5 so both threads collectively handle exactly five orders before termination. 3. Create threads: launch two POSIX threads—take_orders as producer and prepare_food as consumer. 4. Producer logic: in a for-loop from 0 to MAX_ORDERS−1, sleep(1) to simulate order taking, lock the mutex, increment orders, print "Order X taken.", signal the condition to wake the consumer, then unlock. 5. Consumer logic: maintain prepared = 0; while prepared < MAX_ORDERS, lock the mutex; while orders == 0, wait on the condition (which atomically releases the mutex and sleeps); once awakened and orders > 0, print "Preparing order k...", unlock to avoid holding the lock during long work, sleep(2) to simulate cooking, re-lock, decrement orders, increment prepared, print "Order k prepared.", then unlock. 6. Thread teardown: join both threads to ensure completion; destroy the mutex and condition variable; print the final "Restaurant closing" message
	Code: #include <stdio.h> #include <stdlib.h> #include <pthread.h> #include <unistd.h> #define TOTAL_ORDERS 5 int order_queue[TOTAL_ORDERS]; int current_order = 0; pthread_mutex_t lock;

```
pthread_cond_t cond;
void* take_orders(void* arg) {

   printf("Enter the order numbers (1 to 5) in the sequence to be processed:\n");
   for (int i = 0; i < TOTAL_ORDERS; i++) {
      int input;
      scanf("%d", &input);
      if (input < 1 || input > TOTAL_ORDERS) {
         printf("Invalid order number. Please enter a number between 1 and
5.\n");  i--; // retry this index
         continue;

      }
      pthread_mutex_lock(&lock);
      order_queue[i] = input;
      printf("Order %d added to queue.\n", input);
      pthread_mutex_unlock(&lock);
   }
   pthread_cond_signal(&cond); // Notify chef that orders are ready
   return NULL;

}
void* prepare_food(void* arg) {

   pthread_mutex_lock(&lock);
   pthread_cond_wait(&cond, &lock); // Wait until orders are taken
   for (int i = 0; i < TOTAL_ORDERS; i++) {

      int order_num = order_queue[i];
      printf("Preparing order %d...\n", order_num);
      sleep(2); // Simulate preparation time
      printf("Order %d prepared.\n", order_num);
   }
   pthread_mutex_unlock(&lock);
   return NULL;

}
int main() {

   pthread_t order_thread, chef_thread;
   pthread_mutex_init(&lock, NULL);
   pthread_cond_init(&cond, NULL);
```

```
    pthread_create(&order_thread, NULL, take_orders, NULL);
    pthread_create(&chef_thread, NULL, prepare_food, NULL);

    pthread_join(order_thread, NULL);
    pthread_join(chef_thread, NULL);
    pthread_mutex_destroy(&lock);
    pthread_cond_destroy(&cond);
    printf("All orders processed. Restaurant closing.\n");
    return 0;

}
```

Output:



Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
HingnaRoad,Wanadongri, Nagpur - 441110
NAAC A++
Ph.: 07104-237919,234623,329249,329250Fax:07104-232376,Website: www.ycce.edu

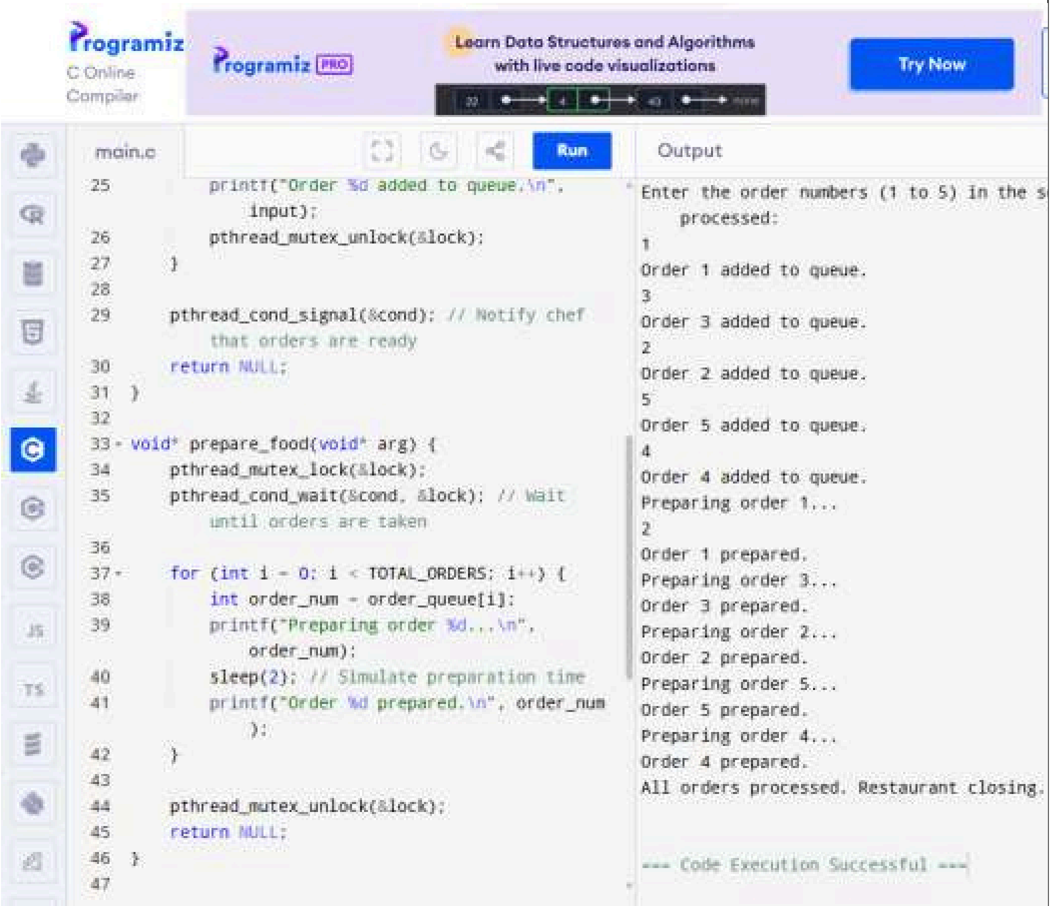
DepartmentofComputerTechnologyB.TechinComputerScienceandEngineering(IOT)

Visionofthe Department

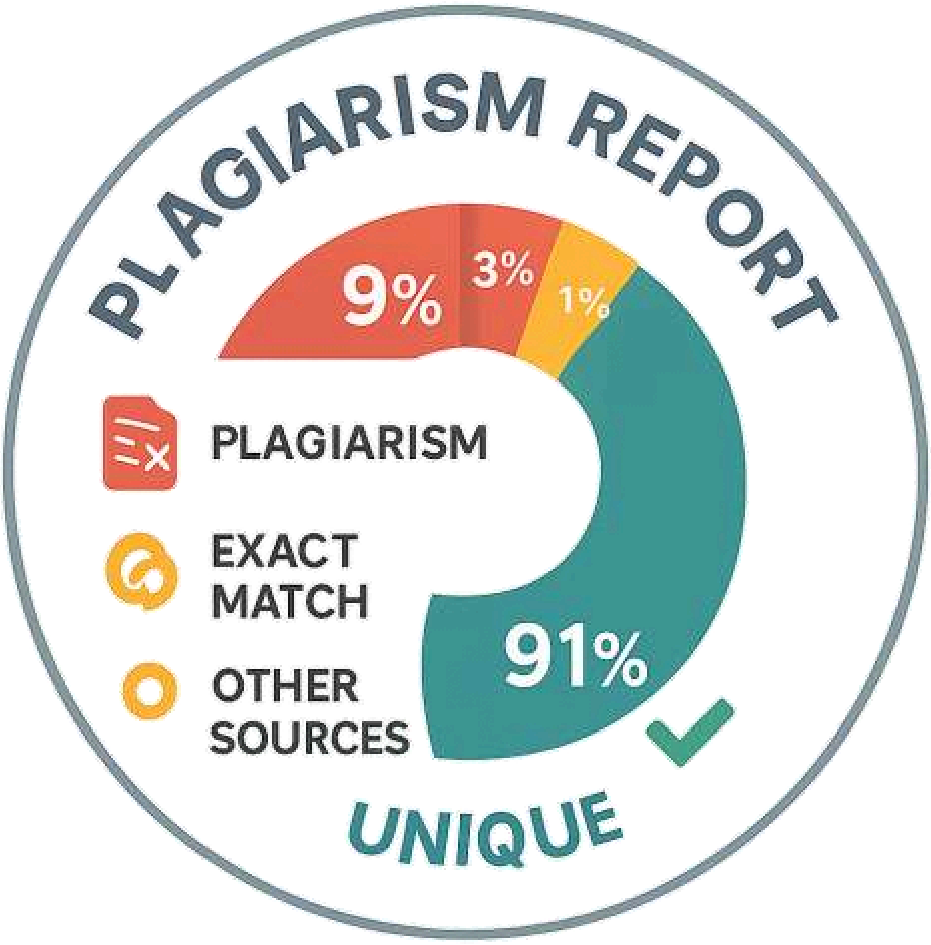
To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.



Output Analysis	The restaurant order system uses two threads with a mutex and condition variable to coordinate a bounded, five-item workflow: one thread collects exactly five order IDs from input and stores them in a queue, then signals once; the other thread sleeps until signaled, then prepares the queued orders sequentially, printing "Preparing/Prepared" for each item and exiting cleanly after the fifth, which demonstrates safe mutual exclusion, blocking instead of busy-waiting, deterministic FIFO processing of the entered sequence, and graceful termination with thread joins and resource cleanup.
Link of student Github	https://github.com/parthbhedurkar

profile where lab assignment has been uploaded	
Conclusion	This practical successfully demonstrates safe inter-thread coordination for a simple producer–consumer workflow using a mutex and a condition variable, achieving correct ordering, no busy-waiting, and clean termination after five orders. It reinforces core OS concepts—critical sections, signaling, and blocking waits—showing how proper synchronization ensures correctness and predictable output in concurrent programs while highlighting design choices such as batching versus incremental signaling for responsiveness.
Plag Report (Similarity index < 12%)	
Date	14/09/2025

Visionofthe Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.