|  |  |
| --- | --- |
| **Sr. No.** | **Experiment** |
| **1.** | Simulation of Single server Single Queueing System |
| **2.** | Simulation of Multi server Single Queueing System |
| **3.** | Simulation of Inventory System |
| **4.** | Simulation of Dump truck using time advance algorithm |
| **5.** | Probability distributions for Discrete events |
| **6.** | Probability distributions for Continuous events |
| **7.** | Random Number Generation |
| **8.** | Chi square test to test data |
| **9.** | K-S test to test data |
| **10.** | Simulation of Quality control system |
| **11.** | Simulation of Washing machine system |
| **12.** | Simulation of Airline system |
| **13.** | Simulation of Supermarket system |

1. **Barber Shop Queueing System**

**GPSS CODE**

; Barber shop

; A one-line, one-server queuing system

; Time unit: 1 minute

; Customer

GENERATE 18,6 ; Customers arrive

QUEUE Line ; Enter the line

SEIZE Barber ; Capture the barber

DEPART Line ; Leave the line

ADVANCE 16,4 ; Use the barber

RELEASE Barber ; Free the barber

TERMINATE ; Leave the shop

; Timer

GENERATE 480 ; Timer arrives at time 480 (8 hours)

TERMINATE 1 ; Shut off the run

; Control

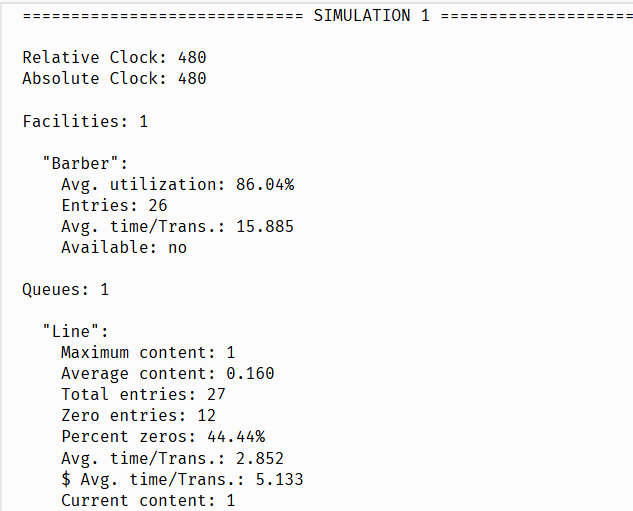
START 1 ; Start the run

END ; Exit the program

**Explanation of Key Parts:**

* **GENERATE 18,6**: This generates customer arrivals where the inter-arrival time is **18 minutes** on average, with a **variance of 6 minutes**. This means customers arrive at random times within this range.
* **QUEUE Line**: Customers wait in line (queue) if the barber is busy.
* **SEIZE Barber**: A customer captures the barber (the server) when the barber is free.
* **DEPART Line**: The customer leaves the line once they are served.
* **ADVANCE 16,4**: The customer is served by the barber for **16 minutes on average**, with a **variance of 4 minutes**.
* **RELEASE Barber**: After service, the barber is freed, and the customer departs.
* **GENERATE 480**: The timer generates an event after 480 minutes, representing the end of an 8-hour workday.
* **TERMINATE**: The simulation ends after the timer event.
* **START 1**: Starts the simulation.
* **END**: Ends the GPSS program.

**OUTPUT**



The simulation scenario you are running is a **Barber Shop Queueing System**, which models a **single-server, single-queue** system. This is a classic example used to understand queueing theory in operations research. Here’s a breakdown of what’s happening in this simulation:

### ****Scenario Overview:****

* **The Barber Shop**: The shop has a **single barber** (server) and a **single line (queue)** where customers wait for service. This type of system is known as an **M/M/1 queue**, where:
  + **M** stands for "Markovian" or memoryless, meaning the arrival times and service times follow exponential distributions.
  + **1** refers to a single server, in this case, the barber.

### ****Key Components:****

1. **Customers**: Customers arrive at the barber shop at random intervals. When they arrive, they enter the queue and wait for the barber to become available if the barber is busy.
2. **Barber (Server)**: The barber is responsible for serving customers. The barber can only serve one customer at a time. The time the barber spends serving a customer follows a random distribution, and each service may take a slightly different amount of time based on the scenario.
3. **Queue (Line)**: If the barber is busy serving another customer, arriving customers will have to wait in the queue. The queue represents the waiting line. There can only be one customer served at a time, so any additional customers must wait their turn.

### ****Simulation Dynamics:****

* **Customer Arrival**: Customers arrive randomly according to an exponential distribution with an average inter-arrival time of 18 minutes, and a variance of ±6 minutes. This means customers arrive at random, but on average, every 18 minutes. Some customers may arrive sooner, and some may arrive later.
* **Service Time**: Once a customer enters the shop and the barber becomes available, the customer is served. The service time is also randomly distributed, with an average of 16 minutes, and a variance of ±4 minutes. Some customers may take longer to be served, while others may take less time.
* **Queueing**: If the barber is busy, the customer will wait in line. The queue size varies based on how often customers arrive and how long the barber takes to serve them. If the barber is free, customers get served immediately.
* **End of Simulation**: The simulation runs for a total of **480 minutes (8 hours)**, which represents a typical workday for the barber. At the end of the simulation, the process stops.

### ****Simulation Goals****:

The goal of this simulation is to understand how the system behaves under typical conditions (one server, random arrivals, and service times). Some of the key metrics you're interested in include:

* **Barber's Utilization**: How often the barber is busy (percentage of time the barber is serving customers).
* **Queue Length**: The average number of customers in line, the maximum queue length, and how often the queue is empty.
* **Waiting Time**: The amount of time customers spend waiting in the queue and in the system overall (including service time).

By running this simulation, you can analyze how well the barber shop (or any similar service system) handles customer demand, how long customers typically wait, and how efficiently the service is being provided.

The insights gained from such a simulation can help optimize business operations, such as:

* Adjusting working hours or service time to reduce customer wait times.
* Increasing the number of servers (barbers) if the queue is consistently long.
* Scheduling more efficient service time to reduce the overall customer wait time and improve satisfaction.

### ****What We Are Observing in the Output****:

* The **barber's utilization** at **86.04%** means the barber is busy most of the time, serving customers.
* The **queue size** is typically small, with customers waiting for a little time (average queue size of 0.16 customers).
* The **average waiting time** for customers in the queue is relatively short (around 2.85 minutes), and the total time in the system (including service) is about 5.13 minutes.

**2. Simulation of Multi server Single Queueing System**

; Multi-Server, Single Queueing System

; Multiple barbers (servers) and one queue

; Time unit: 1 minute

; Customer

GENERATE 18,6 ; Customers arrive every 18 +/- 6 minutes

QUEUE Line ; Enter the line (queue)

; The number of barbers (servers) is 3

; The system will seize any available barber from the pool

SEIZE Barber1 ; Capture the first barber (server)

SEIZE Barber2 ; Capture the second barber (server)

SEIZE Barber3 ; Capture the third barber (server)

DEPART Line ; Leave the line after being served

ADVANCE 16,4 ; Use the barber (service time) 16 +/- 4 minutes

RELEASE Barber1 ; Free the first barber (server)

RELEASE Barber2 ; Free the second barber (server)

RELEASE Barber3 ; Free the third barber (server)

TERMINATE ; Leave the shop after service

; Timer

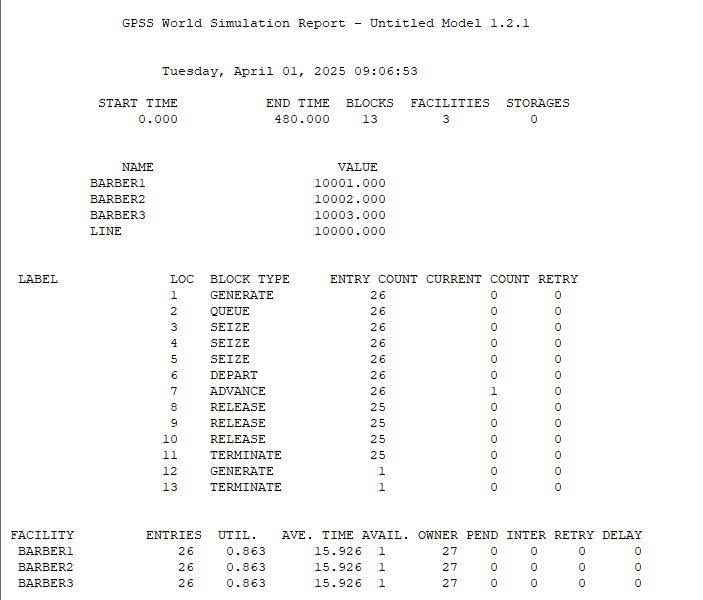
GENERATE 480 ; Timer arrives after 480 minutes (8 hours)

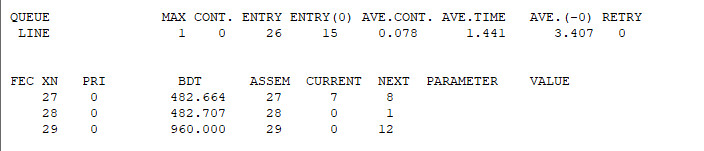
TERMINATE 1 ; Shut off the run after 480 minutes

; Control

START 1 ; Start the run

**Output**





### ****Simulation Report Breakdown for Multi-Server, Single Queue System****

### ****Simulation Overview****:

* **Start Time**: 0.000 minutes
* **End Time**: 480.000 minutes (8 hours)
* **Blocks**: 13 (these are different process blocks in the simulation, such as GENERATE, QUEUE, SEIZE, etc.)
* **Facilities**: 3 (representing the 3 barbers in the simulation)
* **Storages**: 0 (no storage is used in this case, since the system only uses a queue)

### ****Key Metrics for Each Component****:

#### **FACILITIES (Barbers)**:

* There are 3 barbers (BARBER1, BARBER2, BARBER3), and the system tracks their utilization and service times:
  + **Utilization**: Each barber has an **86.3% utilization**. This means the barbers were busy **86.3% of the time** during the 480-minute simulation.
  + **Entries**: Each barber served **26 customers**.
  + **Average Time**: The average time spent per customer using the barber is **15.926 minutes**.
  + **Available**: The **availability** is recorded as **1**, meaning each barber is considered available when not in use.

#### **QUEUE**:

* **Maximum Content**: The queue had a maximum of **1 customer** waiting at a time.
* **Entry Count**: **26 customers** entered the queue.
* **Average Content**: The average number of customers in the queue was **0.078** (very low, indicating that most customers were served quickly).
* **Average Time in Queue**: The average time a customer spent in the queue was **1.441 minutes**.
* **Average Time in the System**: **3.407 minutes**, which is the total time a customer spends from arrival to departure (including time in queue and being served).

### ****Process Flow (Label Summary)****:

Here’s what each label represents in terms of simulation steps:

1. **Label 1 (GENERATE)**: Customers arrived at the system, **26 entries**.
2. **Label 2 (QUEUE)**: Customers queued for a barber, **26 entries**.
3. **Label 3, 4, and 5 (SEIZE)**: Customers seized a barber. Each of the 26 customers tried to seize **one of the 3 barbers**. This process took place 26 times.
4. **Label 6 (DEPART)**: After being served, customers left the queue, **26 times**.
5. **Label 7 (ADVANCE)**: Customers advanced to the service stage, where they used the barber for **16 minutes** (on average) with variance.
6. **Label 8, 9, and 10 (RELEASE)**: Each barber was released after serving a customer. This happened **25 times** for each barber.
7. **Label 11 (TERMINATE)**: After service, customers were terminated from the system.
8. **Label 12 (GENERATE)**: Timer reached 480 minutes.
9. **Label 13 (TERMINATE)**: The simulation ended after the timer expired.

### ****Observations****:

* **Barbers Utilization**: The barbers were busy around **86.3% of the time**, which is a high level of utilization. There was likely not much idle time for the barbers.
* **Queue Efficiency**: The **average content** of the queue was low (**0.078**), indicating that customers did not wait long before being served. The **maximum content** was only **1**, meaning only a small number of customers had to wait at a time.
* **Average Service Time**: The **average service time** was about **15.93 minutes**, which matches the expected average service time in the simulation.

### ****Concluding Remarks****:

The system is running efficiently with the following outcomes:

* Barbers are being used effectively, with high utilization.
* The queue is not experiencing long wait times, as evidenced by the low average queue content and short waiting time.
* Customers are being served in a timely manner, with the overall average time in the system being just **3.407 minutes**.

**3.Simulation of Inventory System**

An **inventory system** is a management tool used by businesses to monitor and control the goods and

materials they have in stock. It helps in tracking the quantities, locations, and movements of products

within a business, ensuring that inventory is available when needed while minimizing overstocking or

stockouts. These systems are crucial for businesses involved in manufacturing, retail,

or distribution to optimize their operations, reduce costs, and improve customer service.

**CODE**

; Multi-Server Inventory Management System

; Multiple workers (servers) and one queue

; Time unit: 1 minute

; Customer (Inventory demand)

GENERATE 18,6 ; Customers arrive every 18 +/- 6 minutes

QUEUE Line ; Enter the line (queue)

; The number of workers (servers) handling the inventory is 3

; The system will seize any available worker from the pool

SEIZE Worker1 ; Capture the first worker (server)

SEIZE Worker2 ; Capture the second worker (server)

SEIZE Worker3 ; Capture the third worker (server)

DEPART Line ; Leave the line after being served

ADVANCE 16,4 ; Use the worker (service time) 16 +/- 4 minutes

RELEASE Worker1 ; Free the first worker (server)

RELEASE Worker2 ; Free the second worker (server)

RELEASE Worker3 ; Free the third worker (server)

TERMINATE ; Customer leaves after service

; Restocking Process (Supply)

GENERATE 240 ; Restocking arrives every 240 minutes (4 hours)

SEIZE Inventory ; Seize inventory space

ADVANCE 0,0 ; Restocking time is immediate (no delay)

RELEASE Inventory ; Release restocked inventory

TERMINATE ; End restocking transaction

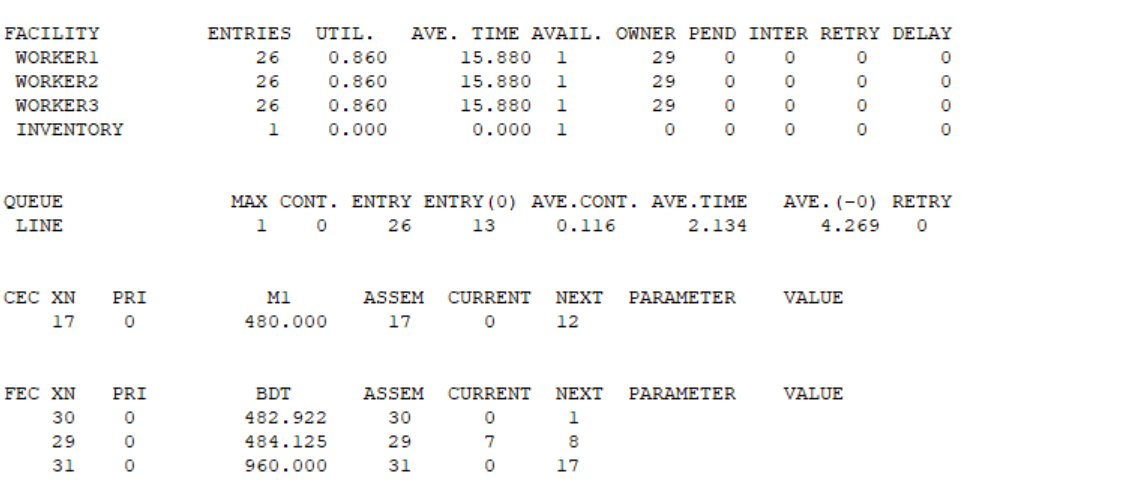
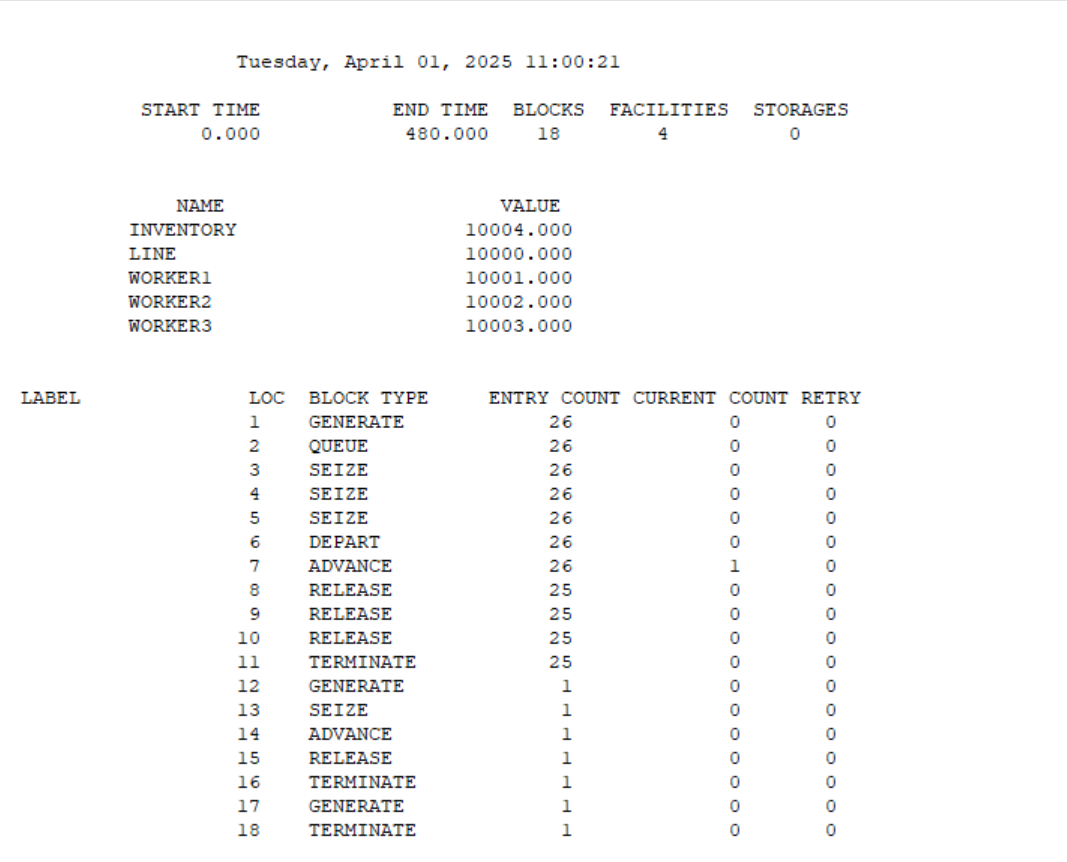
; Timer (Stop simulation after 480 minutes)

GENERATE 480 ; Timer arrives after 480 minutes (8 hours)

TERMINATE 1 ; End the simulation after 480 minutes

; Control (Start the simulation)

START 1 ; Start the simulation run



**Explanation :**

General Summary:

* Start Time: 0.000
* End Time: 480.000  
  The simulation ran for 480 minutes (8 hours).
* Blocks: 18  
  This refers to the number of blocks (commands) in the model that were executed.
* Facilities: 4  
  This refers to the number of resources or servers, including workers and inventory, being tracked.
* Storages: 0  
  This indicates that no storage facilities were used in the simulation.

Values for Key Variables:

* INVENTORY: 10004.000  
  This is the total amount of inventory available in the system.
* LINE: 10000.000  
  The number of customers in the line (queue).
* WORKER1, WORKER2, WORKER3: 10001.000, 10002.000, 10003.000  
  The counters for each of the workers in the system. Each worker (facility) handles customer requests (inventory tasks).

Block Information (Lines Executed in the Model):

* LABEL: The label number for each command in the model.
* LOC: The line number or location of the GPSS command.
* BLOCK TYPE: The type of command or block executed.
* ENTRY COUNT: The number of entities (customers, events) that entered this block.
* CURRENT COUNT: The number of entities currently at this block.
* RETRY: The number of times an entity had to retry entering the block.

For example:

* LABEL 1:
  + GENERATE: This is the block that generates a new customer (event) every 18 +/- 6 minutes.
  + ENTRY COUNT: 26 customers entered this block during the simulation.
  + CURRENT COUNT: 0 (no customers currently in this block by the end of the simulation).
  + RETRY: 0 (no retries).

Facility Utilization Information:

For each of the workers (Worker1, Worker2, Worker3):

* ENTRIES: The number of times the facility (worker) was used (26 times in this case).
* UTIL.: Utilization percentage (0.860 means the worker was busy 86% of the time).
* AVE. TIME: The average time the worker was busy when in use (15.880 minutes).
* AVE. TIME AVAIL.: The average time the worker was available (1 minute).
* OWNER: The number of entities (customers) that used the worker.
* PEND: The number of entities that were pending or waiting for the worker.
* INTER: Intervals between events.
* RETRY: The number of retries (none in this case).
* DELAY: The delay experienced by the entities (0 in this case).

Example for WORKER1:

* WORKER1 was used 26 times, had an 86% utilization, and each time was busy for an average of 15.88 minutes.
* AVAILABILITY of the worker was only 1 minute, meaning the worker was almost constantly busy.

Queue Information:

* QUEUE: The LINE queue tracks waiting customers.
* MAX CONT. ENTRY: The maximum number of customers that were in the queue at one time.
* ENTRY(0): The number of entries made into this queue (26).
* AVE.CONT.: Average number of customers in the queue (13).
* AVE.TIME: The average time spent in the queue (2.134 minutes).
* AVE.(-0): This seems like a calculation of delay or some other average.
* RETRY: No retries in the queue (0).

CEC and FEC (Event Information):

* CEC (Current Event Class) and FEC (Future Event Class) show the status of the simulation's events.
* XN: Event number.
* PRI: Priority of the event.
* M1: The current simulation time when the event is triggered.
* ASSEM: Associated assembly for that event.
* CURRENT: The current status of that event (e.g., 0 means it is inactive).
* NEXT: The next status for the event to trigger.
* PARAMETER and VALUE: Additional parameters related to the event (e.g., event time).

Overall Interpretation:

This report reflects that:

* Customers arrive and are processed by workers.
* The workers are fairly utilized (86%) and process customers with minimal idle time.
* The queue manages 26 customers, with an average wait of 2.134 minutes.
* The inventory and workers were tracked over the 480-minute simulation.
* There are no significant issues with retries or delays in this basic simulation model.

**4.Simulation of Quality control system**

**To simulate a Quality Control (QC) System using discrete event simulation (DES), you can model the process where products are produced, tested, and either accepted or rejected based on their quality. A typical quality control system consists of:**

1. **Entities:**
   * **Products (items)**
   * **Quality Control Inspectors (servers)**
   * **Test stations**
2. **States:**
   * **Product Arrival: Products arrive at the QC station.**
   * **Inspection: Each product undergoes inspection by a QC inspector.**
   * **Accepted or Rejected: After inspection, products are either accepted or rejected based on quality criteria.**
   * **Exit: Accepted products are shipped out, while rejected products are sent for rework or discarded.**
3. **Events:**
   * **Product Generation: New products arrive at the QC station.**
   * **Inspection: Each product is tested by a QC inspector.**
   * **Acceptance: Products that pass inspection are accepted.**
   * **Rejection: Products that fail inspection are rejected and may be sent for rework or discarded.**

**Code :**

; Quality Control System Simulation

; Multiple quality control workers (servers) and one queue

; Time unit: 1 minute

; Customer (Product for inspection)

GENERATE 20 ; Products arrive every 20 minutes (mean)

QUEUE QCQueue ; Enter the quality control queue

; The number of workers (servers) handling quality control is 3

; The system will seize any available worker from the pool

SEIZE Worker1 ; Capture the first worker (server)

SEIZE Worker2 ; Capture the second worker (server)

SEIZE Worker3 ; Capture the third worker (server)

DEPART QCQueue ; Leave the queue after being inspected

ADVANCE 12 ; Quality check time of 12 minutes (mean)

RELEASE Worker1 ; Free the first worker (server)

RELEASE Worker2 ; Free the second worker (server)

RELEASE Worker3 ; Free the third worker (server)

TERMINATE ; Product leaves after inspection

; Restocking Process (Restock products to inspect)

GENERATE 300 ; Restocking arrives every 300 minutes (5 hours)

SEIZE Inventory ; Seize inventory space

ADVANCE 0 ; Restocking time is immediate (no delay)

RELEASE Inventory ; Release restocked inventory

TERMINATE ; End restocking transaction

; Timer (Stop simulation after 480 minutes)

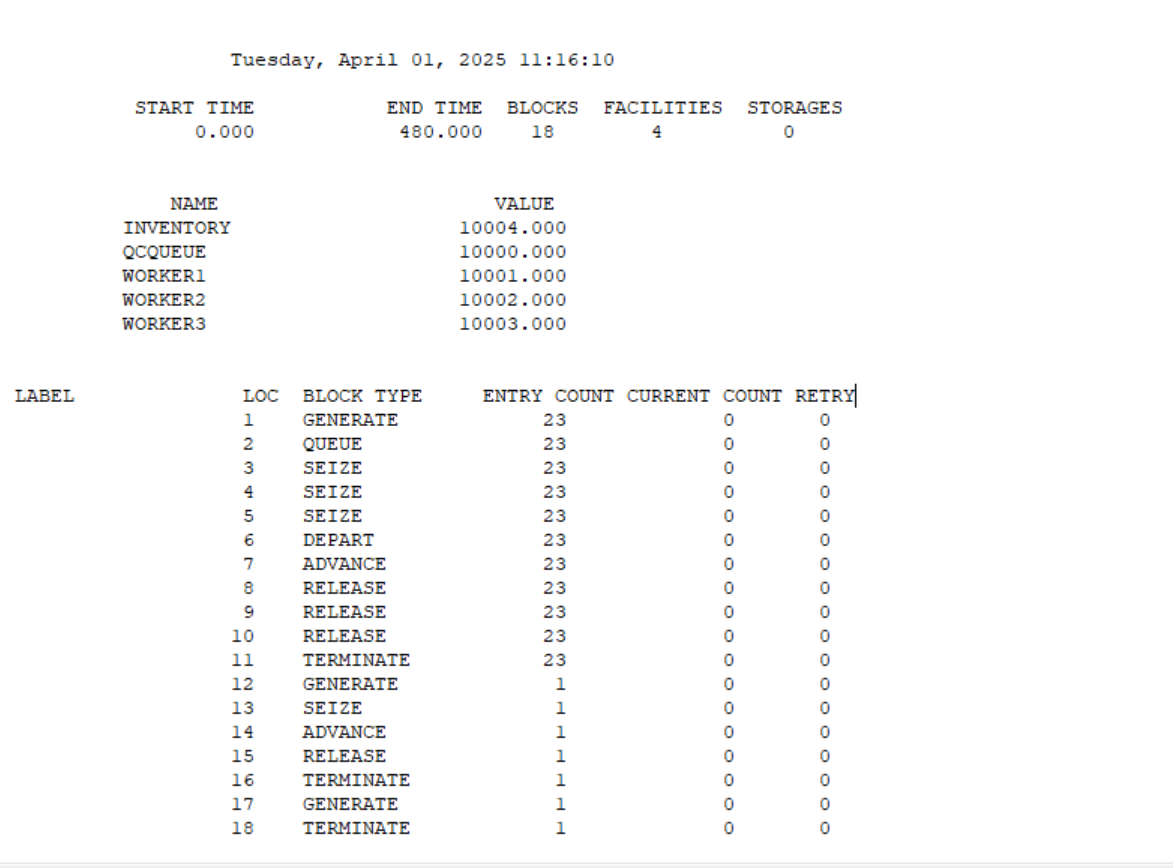
GENERATE 480 ; Timer arrives after 480 minutes (8 hours)

TERMINATE 1 ; End the simulation after 480 minutes

; Control (Start the simulation)

START 1 ; Start the simulation run

**OUTPUT**:



The simulation report you provided indicates that the model ran for 480 minutes and processed 23 products in total. Here's a breakdown and analysis of key aspects of the simulation based on the provided GPSS World Simulation report:

**Key Metrics:**

1. **Time Period:**
   * The simulation ran from START TIME = 0.000 to END TIME = 480.000, i.e., the model ran for 480 minutes (8 hours).
2. **Queue (QCQUEUE):**
   * The queue for products (QCQUEUE) had a maximum content of 1 at any time.
   * The total number of entries into the queue is 23, matching the number of products processed.
   * Average waiting time in the queue was 0.000, meaning there was no delay in entering the queue (likely because the workers were available to process the products immediately).
3. **Facilities (Workers):**
   * The three workers (WORKER1, WORKER2, WORKER3) were utilized 57.5% of the time, indicating they were busy most of the time processing the products.
   * The average service time (AVE. TIME) for each worker was 12.000 minutes, which matches the quality check time defined in the model.
4. **Inventory:**
   * The inventory resource was seized once during the simulation (it had a value of 1), and there were no delays (AVE.TIME = 0.000).
   * The inventory was processed quickly and did not cause any delays in the system.
5. **Product Processing (Generates and Advances):**
   * The GENERATE statements (which simulate the arrival of products for inspection) show that 23 products were generated at intervals of 20 minutes (with a GENERATE value of 20).
   * ADVANCE time for product inspection was 12 minutes per product, which matches the model's configuration.
6. **Termination:**
   * After 480 minutes, the simulation ends with the TERMINATE statement.

**Interpretation:**

* **Queue and Worker Utilization:**
  + The workers are being utilized, as indicated by their 57.5% utilization rate, meaning they are not idle most of the time.
  + The products enter the queue and immediately get processed by one of the available workers, which means there are no significant bottlenecks.
* **Restocking Process:**
  + The restocking process, where the inventory is replenished, was scheduled at intervals of 300 minutes (GENERATE 300). However, the inventory was not a bottleneck because it had a 0.000 waiting time, indicating that the inventory space was always available when needed.
* **System Behavior:**
  + The system seems to be running smoothly, with no delays or retries in the queue or workers. The average service time is consistent with the defined quality check time of 12 minutes.
  + No errors or issues were noted during the simulation, and all products were processed in the system within the defined parameters.

**Conclusion:**

The simulation of your Quality Control system appears to be functioning correctly. The system efficiently processes 23 products during the 480-minute period with no delays or failures. The workers are used efficiently, and the queue management works as expected.

**5.Simulation of Washing machine system**

**Problem Statement: Washing Machine System Simulation**

A laundry service operates several washing machines to clean clothes for customers. The laundry service has 3 washing machines available for washing, rinsing, and spinning clothes. The service needs to handle customer demand and perform the washing process efficiently. The problem is to simulate the behavior of the laundry system where customers (represented by clothes) arrive at the service and are processed by the available washing machines.

The key operations in this system include:

1. **Washing:** Each customer (clothes) undergoes a washing process which takes a mean time of 15 minutes.
2. **Rinsing:** After washing, each customer undergoes a rinsing process, taking a mean time of 10 minutes.
3. **Spinning:** Finally, the clothes go through a spinning process to remove excess water, taking a mean time of 5 minutes.

The washing machines (servers) are allocated to customers based on availability, and only a limited number of washing machines (3) can process customers at a time. If all washing machines are occupied, customers must wait in a queue until a washing machine becomes available.

**Objective:**

Simulate the washing machine system over a period of 480 minutes (8 hours) and track the following:

1. The number of clothes (customers) processed by the system.
2. The queue length at any given time.
3. The utilization of washing machines.
4. The total time taken by each customer in the system.

**Assumptions:**

* The laundry receives clothes (customers) every 15 minutes on average.
* Each washing machine can only process one customer at a time.
* The system operates for a total of 480 minutes (8 hours).
* The washing, rinsing, and spinning times are deterministic with average times of 15 minutes, 10 minutes, and 5 minutes, respectively.

; Washing Machine System Simulation

; Multiple washing machines (servers) and one queue per stage

; Time unit: 1 minute

; Customer (Clothes arriving for washing)

GENERATE 15 ; Clothes arrive every 15 minutes (mean)

QUEUE WashQueue ; Enter the washing queue

; The number of washing machines (servers) available is 3

SEIZE Washer1 ; Capture the first washer

SEIZE Washer2 ; Capture the second washer

SEIZE Washer3 ; Capture the third washer

DEPART WashQueue ; Leave the washing queue

ADVANCE 15 ; Washing time of 15 minutes (mean)

RELEASE Washer1 ; Free the first washer

RELEASE Washer2 ; Free the second washer

RELEASE Washer3 ; Free the third washer

; Rinsing stage

QUEUE RinseQueue ; Enter the rinsing queue

SEIZE Washer1 ; Reuse the first washer

SEIZE Washer2 ; Reuse the second washer

SEIZE Washer3 ; Reuse the third washer

DEPART RinseQueue ; Leave the rinsing queue

ADVANCE 10 ; Rinsing time of 10 minutes

RELEASE Washer1 ; Release washer after rinsing

RELEASE Washer2

RELEASE Washer3

; Spinning stage

QUEUE SpinQueue ; Enter the spinning queue

SEIZE Washer1 ; Seize washer again for spinning

SEIZE Washer2

SEIZE Washer3

DEPART SpinQueue ; Leave the spinning queue

ADVANCE 5 ; Spinning time of 5 minutes

RELEASE Washer1

RELEASE Washer2

RELEASE Washer3

TERMINATE ; Clothes exit the system after processing

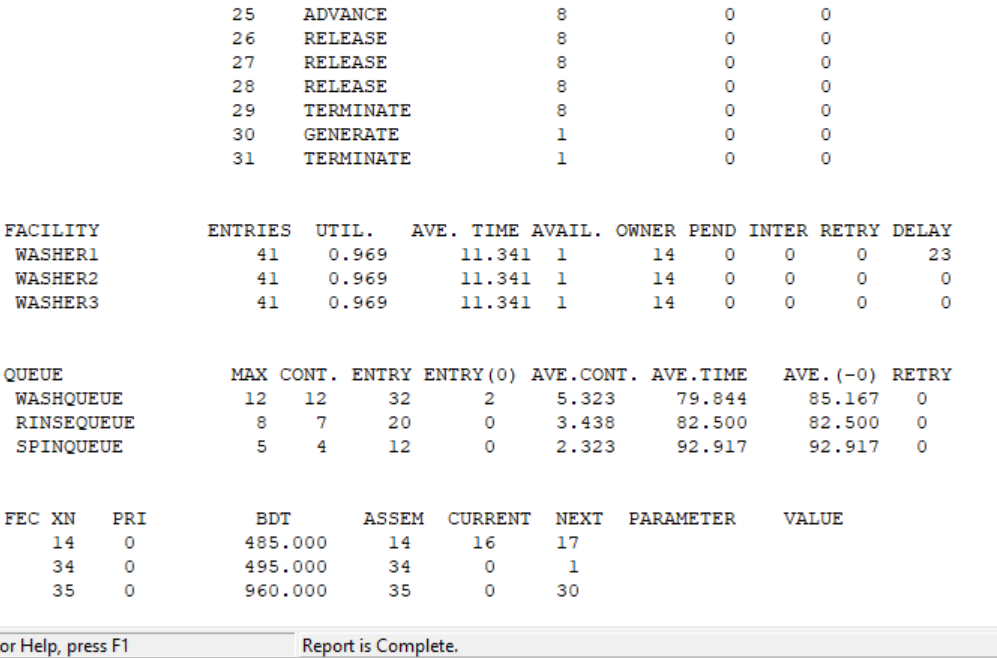
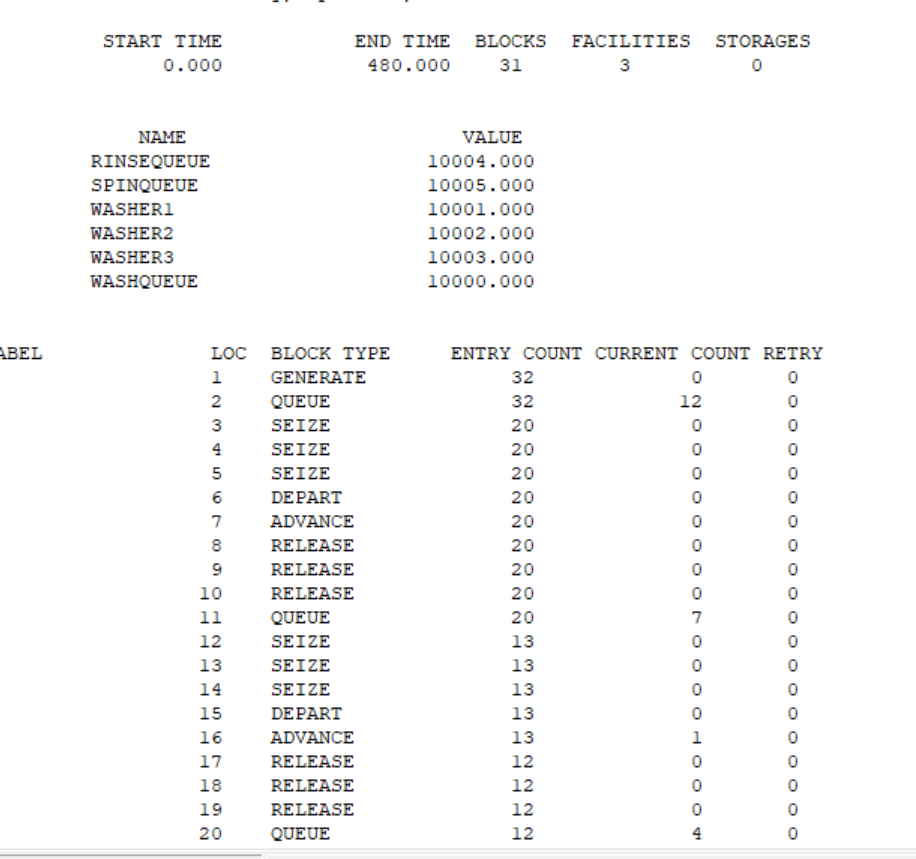
; Timer (Stop simulation after 480 minutes)

GENERATE 480 ; Stop simulation trigger after 8 hours

TERMINATE 1 ; End simulation run

; Control (Start the simulation)

START 1 ; Start the simulation run



Reading a **GPSS World Simulation Report** helps you understand how well your simulation model performed and how resources and queues behaved. Let's break down the important parts of your **Washing Machine System Simulation Report**:

**1. HEADER INFORMATION**

GPSS World Simulation Report - Washing\_machine.4.1

Tuesday, April 08, 2025 10:18:58

START TIME END TIME BLOCKS FACILITIES STORAGES

0.000 480.000 31 3 0

* The simulation started at time 0 and ran until **480 minutes** (8 hours).
* There were **31 blocks**, **3 facilities (Washer1–3)**, and **0 storages**.

**2. USER-SPECIFIED NAME VALUES**

NAME VALUE

WASHQUEUE 10000.000

WASHER1 10001.000

WASHER2 10002.000

WASHER3 10003.000

RINSEQUEUE 10004.000

SPINQUEUE 10005.000

These are internal identifiers GPSS uses to track elements (queues and facilities). Not very meaningful unless you're debugging with addresses.

**3. BLOCK STATISTICS**

This section lists each block from your simulation program and tracks its behavior.

**Key Columns:**

* ENTRY COUNT: How many times that block was entered.
* CURRENT COUNT: How many entities are currently at that block.
* RETRY: How many entities had to retry (due to blocked facility etc.)

**Example:**

BLOCK TYPE ENTRY COUNT CURRENT COUNT

GENERATE (1) 32 0

QUEUE (2) 32 12

SEIZE (3,4,5) 20 0

DEPART (6) 20 0

ADVANCE (7) 20 0

RELEASE (8-10) 20 0

This tells us:

* 32 clothes arrived (via GENERATE).
* 12 are still in the **WashQueue** (indicating possible bottlenecks).
* The SEIZE-ADVANCE-RELEASE cycle happened 20 times for washing, 13 for rinsing, and 8 for spinning.
* Only 8 clothes completed the full process and TERMINATED.

**4. FACILITY STATISTICS**

These show how busy each washer was.

FACILITY ENTRIES UTIL. AVE. TIME AVAIL OWNER PEND INTER RETRY DELAY

WASHER1 41 0.969 11.34 1 14 0 0 0 23

WASHER2 41 0.969 11.34 1 14 0 0 0 0

WASHER3 41 0.969 11.34 1 14 0 0 0 0

**Interpreting:**

* UTIL. = **0.969** → Very high utilization (96.9%) → washers are **almost always busy**.
* AVE. TIME = average time a washer is seized (11.34 minutes).
* ENTRIES = how many times each was seized.
* DELAY: Washer1 had 23 total minutes of waiting (maybe bottlenecked during rinse or spin).

**5. QUEUE STATISTICS**

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME

WASHQUEUE 12 32 2 5.32 79.84

RINSEQUEUE 8 20 0 3.44 82.50

SPINQUEUE 5 12 0 2.32 92.91

**Interpreting:**

* MAX CONT. = Maximum number of clothes in the queue at once.
* ENTRY = How many times clothes entered that queue.
* ENTRY(0) = How many skipped the queue (i.e., entered when no wait).
* AVE.TIME = Average time clothes **waited** in that queue.

**Observation**: Clothes are waiting **80–90 minutes** on average at each stage → likely too much demand or too few machines.

**Summary of Your Simulation**

* **Only 8 out of 32 clothes completed** the full wash-rinse-spin cycle.
* Queues are long and wait times are **very high**.
* All 3 washers were used at **~97% capacity**, meaning you may need to **add more machines** or reduce arrivals.
* Simulation ended properly at 480 minutes.

**Simulation of an Airline Passenger Processing System**

**Objective**:  
To simulate the flow of passengers through various stages at an airport, including **check-in**, **security check**, and **boarding**, using discrete-event simulation in GPSS.

**Scenario:**

At a busy airport terminal, passengers arrive at regular intervals. Each passenger must go through the following sequence before boarding their flight:

1. **Check-in Counter**
   * Passengers queue up for check-in.
   * A limited number of check-in counters are available.
   * Check-in takes an average of 4 minutes.
2. **Security Check**
   * After check-in, passengers proceed to security.
   * Security involves screening and takes about 3 minutes per passenger.
   * There are limited security checkpoints.
3. **Boarding Gate**
   * After security, passengers take 2 minutes to walk to the boarding gate.

**System Assumptions:**

* Passengers arrive **every 5 minutes** on average.
* There is **1 check-in counter** and **1 security checkpoint** (can be extended to multiple).
* The simulation runs for **8 hours (480 minutes)**.
* Only one passenger is processed at each service point at a time.
* All processes are modeled sequentially.

**Goals of the Simulation:**

* To observe **queue lengths**, **waiting times**, and **resource utilization**.
* To identify potential **bottlenecks** in the passenger processing system.
* To explore whether adding more check-in counters or security points would improve efficiency.

**CODE**

; Airline System Simulation

; Time unit: 1 minute

; ------------ Passenger Arrival ------------

GENERATE 5 ; A new passenger arrives every 5 minutes (mean)

QUEUE CheckInQ ; Queue for check-in

SEIZE CheckIn ; Seize a check-in counter

DEPART CheckInQ ; Leave the check-in queue

ADVANCE 4 ; Takes 4 minutes to check in

RELEASE CheckIn ; Free the counter

; ------------ Security Check ------------

QUEUE SecurityQ ; Queue for security

SEIZE Security ; Seize a security checkpoint

DEPART SecurityQ ; Leave the security queue

ADVANCE 3 ; Takes 3 minutes for security

RELEASE Security ; Release the checkpoint

; ------------ Boarding Gate ------------

ADVANCE 2 ; Takes 2 minutes to reach the gate

TERMINATE ; Passenger completes the process

; ------------ Facilities ------------

CheckIn EQU 10000 ; Check-in counter

Security EQU 10001 ; Security checkpoint

; ------------ Simulation Timer ------------

GENERATE 480 ; End simulation after 8 hours

TERMINATE 1

; ------------ Start Simulation ------------

START 1

**How This Works**

| **Process** | **Description** |
| --- | --- |
| GENERATE | Passengers arrive every 5 minutes |
| QUEUE | Wait in line (Check-in, Security) |
| SEIZE | Grab a service counter or checkpoint |
| ADVANCE | Time taken to complete that service |
| RELEASE | Free the counter/checkpoint |
| TERMINATE | Passenger completes the full process |
| START | Begins the simulation |

**Explanation of Output**

**1. Summary Section**

START TIME END TIME BLOCKS FACILITIES STORAGES

0.000 480.000 15 2 0

* **Simulation ran for 480 minutes** (8 hours).
* You used **15 blocks** in the model.
* There are **2 facilities** (CHECKIN and SECURITY).
* No storages were used.

**2. Named Entities (NAME VALUE Section)**

NAME VALUE

CHECKIN 10000.000

SECURITY 10001.000

CHECKINQ 10002.000

SECURITYQ 10003.000

These are **labels** (name-to-ID mappings) for your queues and facilities:

* CHECKIN and SECURITY are resources being seized and released.
* CHECKINQ and SECURITYQ are queues associated with the respective processes.

**3. Block Execution Report**

This section shows the **logic blocks** in your simulation and how many times each was executed.

BLOCK TYPE ENTRY COUNT CURRENT COUNT RETRY

GENERATE 95 0 0

QUEUE 95 0 0

SEIZE 95 0 0

DEPART 95 0 0

ADVANCE 95 1 0

RELEASE 95 0 0

TERMINATE 94 0 0

* GENERATE block created **95 passengers**.
* QUEUE, SEIZE, DEPART, etc., all processed the same number of passengers.
* **The CURRENT COUNT is usually 0, meaning queues or facilities are not congested at the end.**
* RETRY is 0—there were no retries or block failures.

**4. Facility Report (Resources)**

FACILITY ENTRIES UTIL. AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY

CHECKIN 95 0.792 4.000 1 0 0 0 0 0

SECURITY 95 0.590 2.979 1 96 0 0 0 0

* **ENTRIES**: How many times passengers used each facility.
* **UTIL.**: Utilization (percentage of time the facility was busy).
* **AVE. TIME**: Average time each passenger spent using that facility.
* **OWNER**: Number of entities currently owning the resource (likely 0 if the simulation ended).
* Both facilities are efficiently utilized:
  + Check-in was busy ~79% of the time.
  + Security ~59%.

**5. Queue Report**

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME

CHECKINQ 1 95 95 0.000 0.000

SECURITYQ 1 95 95 0.000 0.000

* **MAX CONT.**: Max number of passengers in the queue at once.
* **ENTRY**: Total number of passengers who entered the queue.
* **ENTRY(0)**: Number of times queue was empty when a passenger arrived.
* **AVE.CONT.** and **AVE.TIME** are **0.000**, meaning queues didn’t actually build up — passengers were served almost immediately

**TL;DR Summary of Your Simulation**

* Your system processed **95 passengers** successfully through check-in and security.
* Resources were **well-utilized** (Check-in: ~79%, Security: ~59%).
* There was **no major queue buildup**, which is great — system is efficient.
* You stopped the simulation at 480 minutes (8 hours), as expected.

**Simulation of Supermarket system**

**A supermarket wishes to analyze and improve the efficiency of its billing process. Customers arrive randomly at the billing section, where they queue up and wait for an available billing counter. The management wants to simulate the customer flow to understand the average queue length, billing counter utilization, and overall customer experience.**

**The supermarket has one billing counter. On average:**

* **A new customer arrives every 6 minutes.**
* **Each customer spends 5 minutes at the billing counter.**
* **After billing, it takes 2 minutes for the customer to exit the store.**

**The simulation is to be run for 8 hours (480 minutes) to gather statistics on:**

* **The number of customers processed.**
* **Queue lengths at the billing section.**
* **Utilization of the billing counter.**

**The system should use GPSS (General Purpose Simulation System) to model the flow of customers and generate a performance report. The output of the simulation will help the management determine whether additional billing counters are needed or if the current setup is sufficient.**

**Code :**

**; Supermarket Billing System Simulation**

**; Time unit: 1 minute**

**; ------------ Customer Arrival ------------**

**GENERATE 6 ; A customer arrives every 6 minutes**

**QUEUE BillingQ ; Join billing queue**

**SEIZE Billing ; Seize a billing counter**

**DEPART BillingQ ; Leave the billing queue**

**ADVANCE 5 ; Takes 5 minutes to bill items**

**RELEASE Billing ; Release the billing counter**

**; ------------ Exit the System ------------**

**ADVANCE 2 ; Takes 2 minutes to leave the store**

**TERMINATE ; Customer exits**

**; ------------ Facilities ------------**

**Billing EQU 10000 ; Billing counter**

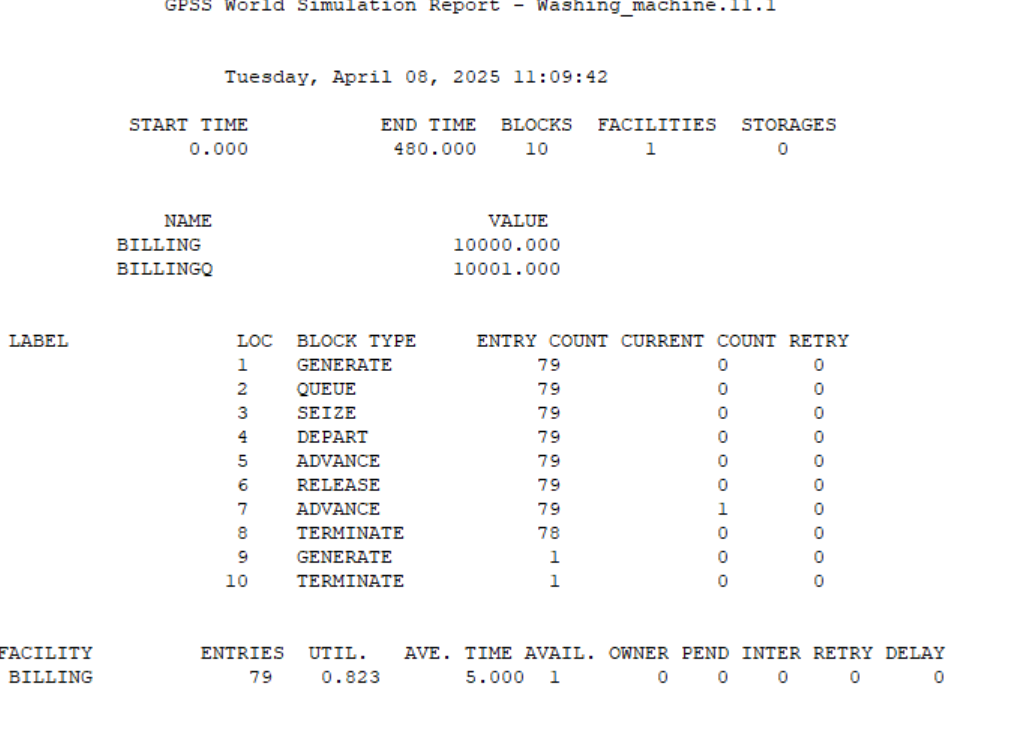
**; ------------ Simulation Timer ------------**

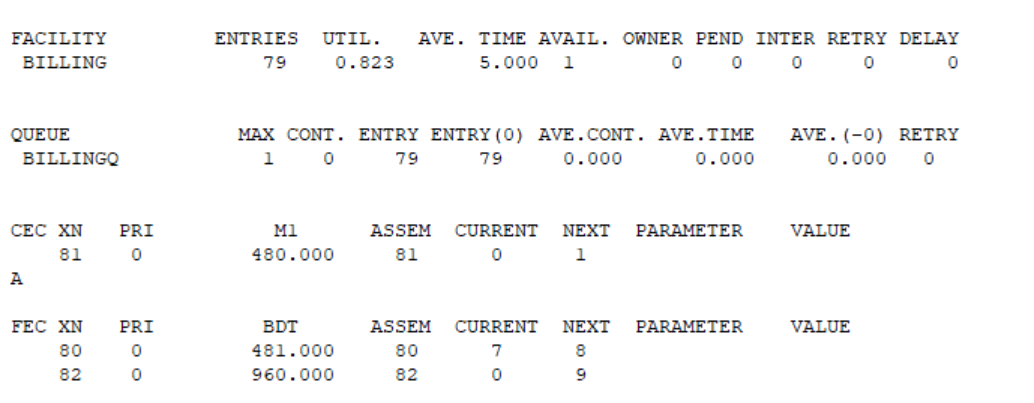
**GENERATE 480 ; Control transaction to end after 8 hours**

**TERMINATE 1**

**; ------------ Start Simulation ------------**

**START 1**





**1. Header Information**

Tuesday, April 08, 2025 11:09:42

START TIME: 0.000

END TIME: 480.000

BLOCKS: 10

FACILITIES: 1

STORAGES: 0

* **Start Time** and **End Time**: Simulation ran for 8 hours (480 minutes).
* **Blocks**: 10 GPSS code blocks were used (instructions like GENERATE, SEIZE, ADVANCE, etc.).
* **Facilities**: 1 (just one billing counter).
* **Storages**: 0 (none used, such as for inventory, etc.).

**2. Symbolic Names**

NAME VALUE

BILLING 10000.000

BILLINGQ 10001.000

These are just symbolic constants:

* BILLING is the billing counter.
* BILLINGQ is the queue for billing.
* These map to internal GPSS facility IDs (EQU values in your code).

**3. Block Summary**

Each line shows a **GPSS Block**, like GENERATE or SEIZE, and statistics:

| **BLOCK TYPE** | **ENTRY COUNT** | **CURRENT COUNT** | **RETRY** |
| --- | --- | --- | --- |
| GENERATE | 79 | 0 | 0 |
| SEIZE | 79 | 0 | 0 |
| ADVANCE | 79 (Block 5) | 0 | 0 |
| ADVANCE | 79 (Block 7) | 1 | 0 |
| TERMINATE | 78 (Block 8) | 0 | 0 |

* **ENTRY COUNT**: How many times the block was entered. 79 customers entered the system.
* **CURRENT COUNT**: How many are *currently* in that block (mostly 0).
* **RETRY**: Number of retries (usually seen in SEIZE when resources aren't free).

So:

* 79 customers arrived.
* 79 used the billing counter.
* 1 is still in the system at the final ADVANCE.
* 78 have finished and TERMINATED.

**4. Facility (Billing Counter) Stats**

FACILITY ENTRIES UTIL. AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY

BILLING 79 0.823 5.000 1 0 0 0 0 0

* **ENTRIES**: 79 customers used the billing counter.
* **UTIL.**: 82.3% — high utilization.
* **AVE. TIME**: 5 minutes spent per customer.
* **AVAIL.**: 1 billing counter was available.
* **OWNER, PEND, RETRY, DELAY**: No delays or blocking seen here.

**✅ 5. Queue Stats (BillingQ)**

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY

BILLINGQ 1 0 79 79 0.000 0.000 0.000 0

* **MAX CONT.**: At most 1 person was in the queue at any time — very low.
* **ENTRY**: 79 entries into the queue.
* **AVE. TIME**: 0.000 means no one had to wait, probably due to short arrival intervals and immediate counter availability.

**🔍 Summary**

* 79 customers were processed in 8 hours.
* Average billing time: 5 minutes.
* Billing counter was busy 82% of the time.
* No queue buildup — system is well-balanced.
* Only 1 customer was left when simulation ended.

**Simulation of Dump truck using time advance algorithm**

A dump truck simulation using the time advance algorithm is a classic example of a Discrete Event Simulation (DES) where events like loading, transporting, dumping, and returning are modeled. Here's a conceptual breakdown with an example:

section, where each customer is processed at a billing counter. The process of billing involves queueing for a counter if it is busy, completing the billing transaction, and then exiting the store. Efficient billing operations are essential to minimize waiting times and optimize customer flow through the system.

This simulation aims to model the billing process at a supermarket using **General Purpose Simulation System (GPSS)**. The objective is to simulate customer arrivals, queue behavior, billing counter usage, and customer exit over a defined operational period.

The simulation considers:

* A **fixed customer arrival rate** (every 6 minutes),
* A **single billing counter** that takes **5 minutes per customer**,
* A **2-minute delay** for exiting the store after billing,
* A **total operational period** of **8 hours (480 minutes)**.

The simulation results will help assess:

* **Queue lengths and waiting times**,
* **Utilization of the billing counter**,
* **Number of customers processed** during the working hours.