GPSS (General Purpose System Simulation)

Introduction:

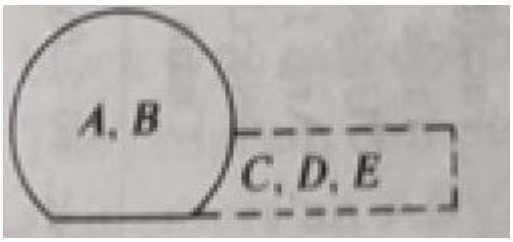
GPSS is a discrete time simulation language, where a simulation clock advances in discrete steps. A system is modeled as transactions (processes) that; Enter the system and are passed from one service (represented by blocks) to another. User translates his problem into a conceptual model, which is a block diagram. Then GPSS software package: Processes this block diagram, Executes the simulation run, and Produces statistics.

Transaction

Transaction in GPSS is a process that represents the real-world system you are modeling which is executed by moving from block to block. Each transaction in the model is contained in exactly one block, but one block may contain many transactions. The Various Blocks of a GPSS are as follows:

1. GENERATE BLOCK:

This block will produce a flow of transactions with inter-arrival times determined by the attribute values. The Label is optional. The distribution of inter-arrival times follows a uniform probability distribution.



SYNTAX:

Line Number Label GENERATE A, B, C, D, E *Line Number and Label Optional*

EXAMPLE: GENERATE 5, 2

ATTRIBUTES:

A = Average value of uniform distribution (mean)

B = Modifier

C = time delay before first transaction is generated

D = maximum Number of transactions generated

E = priority allocated to transactions

Generally C, D and E are rarely used

2. QUEUE BLOCK:

This block will instruct GPSS to start gathering queuing statistics on the queue named in its attribute value. The Label is optional but may be necessary if you have to refer to this Line from somewhere else in the program

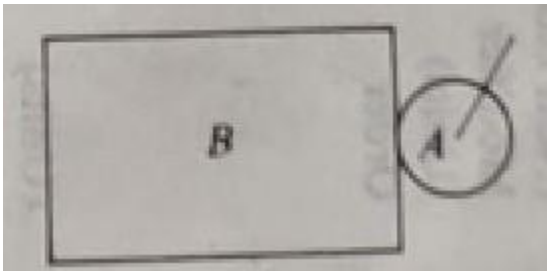
SYNTAX:

Line Number Label QUEUE A

ATTRIBUTES:

GPSS (General Purpose System Simulation)

A = name of queue (For Example: Garage)



EXAMPLE: QUEUE Garage

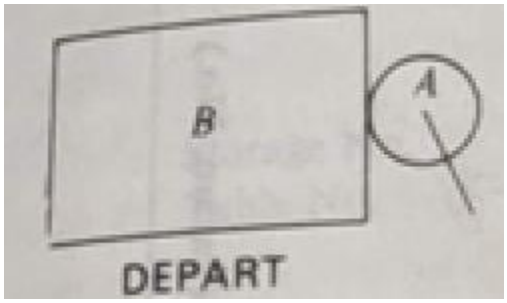
If a transaction arriving at the queue block cannot proceed because it is blocked by the next stage, then it will stay in the queue block until it can gain entry to the next stage.

3. DEPART BLOCK:

This block instructs GPSS that a transaction is leaving the queue named in its attribute value. This is necessary in order to compile the statistics on the queue. The Label is optional.

SYNTAX:

Line Number Label DEPART A



ATTRIBUTES:

A = name of the queue (For Example: Garage) **EXAMPLE**: DEPART Garage

*Note: If **QUEUE** block is used in the model then the corresponding **DEPART** block should follow it.*

4. SEIZE BLOCK:

This block allows the transaction to seize a **Facility** if it is free. Thus it may be a car "seizing" a "facility" such as a petrol pump or a customer in a supermarket "seizing" a "facility" such as the checkout assistant. When the car or customer is being serviced by the facility, then it is said to "own the facility". The Label is optional.

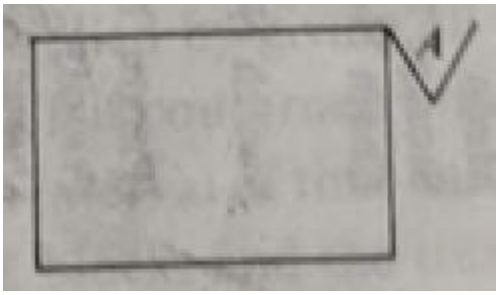
SYNTAX:

Line Number Label SEIZE A

GPSS (General Purpose System Simulation)

ATTRIBUTES:
A = Name of **Facility** (For Example: Pump)
EXAMPLE: SEIZE PUMP
Note: A transaction can only seize a facility if it is free or else wait until the owning transaction releases it.

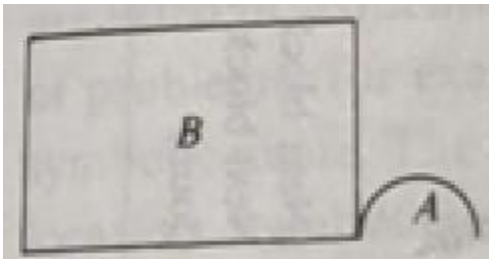
5. RELEASE BLOCK:
A transaction entering this block informs GPSS that it is giving up ownership of the **Facility** named in its attribute value. The Label is optional.



SYNTAX:
Line Number Label RELEASE A
ATTRIBUTES:
A = Name of Facility (For Example: Pump)
Note: By giving up ownership of the facility, the transaction makes it available for another transaction that may be waiting to use it.
*Note: If **SEIZE** block is used in the model then the corresponding **RELEASE** block should follow it.*

6. ENTER BLOCK:
This Block instructs GPSS that a transaction has entered STORAGE. The name of storage **(Storage Contains Multiple Service Points)** is given by the first attribute value. The second attribute value gives the amount the storage will be incremented by, when the transaction enters the ENTER block. STORAGE must be declared at the beginning of a program. For Example:
100 Warehouse STORAGE 25

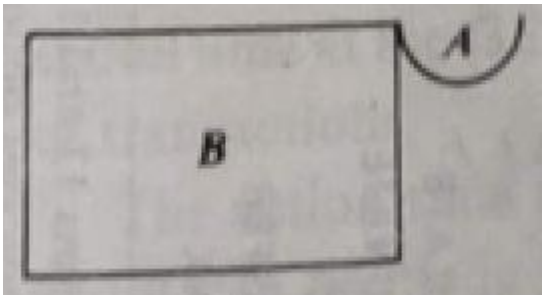
In the 'Label' section you must give the STORAGE a name so that the ENTER block can refer to it. The "verb" is STORAGE and the attribute value A, which is 25 in this Example, states the maximum capacity of the Warehouse.
SYNTAX:
Line Number Label ENTER A, B



GPSS (General Purpose System Simulation)

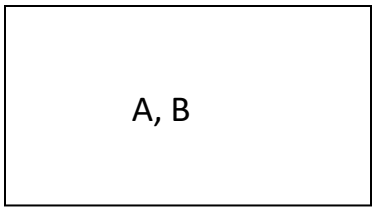
ATTRIBUTES:
A = name of the storage (for Example: warehouse)
B = increment storage by this value

7. LEAVE BLOCK:
This block instructs GPSS that a transaction is leaving a STORAGE. The first attribute gives the name of the STORAGE and the second attribute decrements the storage by the value of the attribute.



SYNTAX:
Line Number Label LEAVE A, B
ATTRIBUTES:
A = name of the storage (for Example: warehouse)
B = Decrement storage by the value
*Note: If **ENTER** block is used in the model then the corresponding **LEAVE** block should follow it.*

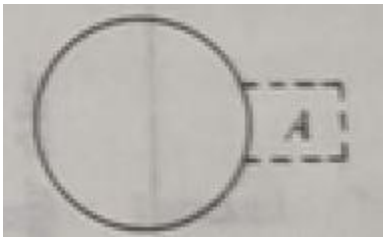
8. ADVANCE BLOCK:
This block represents the servicing of a transaction. The servicing times follow a uniform probability distribution. The Label is optional.
SYNTAX:
Line Number Label ADVANCE A, B



ATTRIBUTES:
A = average value of uniform distribution (mean)
B = Modifier
A transaction entering this block will be delayed by a time interval chosen at random from the specified probability distribution.

8. TERMINATE BLOCK:

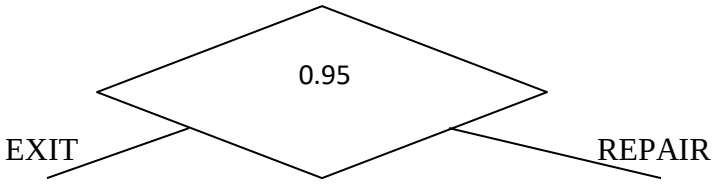
This block destroys any transaction entering it and removes it from computer memory. Each time a transaction enters this block it decrements a counter by an amount equal to its attribute value. The counter is set by the user upon starting the simulation.



SYNTAX:
Line Number Label TERMINATE A
ATTRIBUTES:
A = decrements simulation counter by this amount
When the counter, set at the beginning of the simulation, reaches zero then the simulation is complete and a statistical report is produced on the outcome of the simulation

9. TRANSFER BLOCK:

This block will take transactions entering it and transfer them to each of two different destinations according to laid down proportions. For Example:



TRANSFER 0.95, EXIT, REPAIR

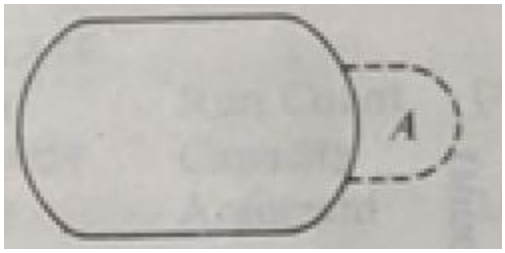
In this case 95% of all transactions entering the TRANSFER block will go to the program Line Labelled REPAIR and 5% will go to the program Line Labelled EXIT. If the second attribute "EXIT" is replaced by a "comma", then the 5% will go to the next block in the program.

SYNTAX:
Line Number Label TRANSFER A,B,C
ATTRIBUTES:
A = probability value (0 to 1)
B = proportion of (1-A) transactions transferred to this Labelled location
C = proportion A transactions transferred to this Labelled location

GPSS (General Purpose System Simulation)

10. MARK

It marks the absolute clock time that the Transaction first entered the simulation or entered a MARK Block.



SYNTAX: MARK (Param No)

TABLE IN GPSS

A Table Entity is a set of integers used to accumulate data for a histogram. Each integer represents a frequency class in a histogram. A Table Entity is defined by a TABLE Command.

11. TABULATE: TABULATE Blocks update the histogram data accumulated in a Table Entity.

SYNTAX: TABULATE Table No

Format of GPSS Program

In GPSS each separate Line of the program will either be a statement or a block. (In a few rare situations the GPSS statement will be continued for two or more Lines). A general format of a GPSS block consists of four separate items. These are:

1. Label or location
2. Operation or block statement
3. Operands
4. Comments

Controlling the Length of a Simulation Run(Time Driven)

The terminate block can contain an attribute value that will make the simulation stop when the Number of transactions entering the TERMINATE block equals the attribute value.

GPSS (General Purpose System Simulation)

In a GENERATE statement it is possible to specify the maximum Number of transactions that the GENERATE block will produce. Clearly the model will cease to function once this limit has been reached and the last transaction has been terminated.

It is possible to have a separate timing section in your program that will determine the exact time of a simulation. For Example:

```
TIMER    GENERATE    2000
         TERMINATE   1
```

This will cause the GENERATE block in Line 500 to produce a single transaction at 2000 time units after the simulation starts and decrement the termination counter to zero. This will immediately stop the simulation run. You must ensure there is no attribute value in any other TERMINATE block in the simulation model.

Example of Program in GPSS: Joe's Barbershop (Time Driven)

Suppose we want to simulate a typical day of nine hours' work. Since we have decided that one time unit is equal to one minute then we must convert nine hours into minutes. That is 9 x 60 minutes = 540 minutes.

```
MEN      GENERATE    18,6
         QUEUE       SEAT
         SEIZE       JOE
         DEPART      SEAT
         ADVANCE     15,3
         RELEASE     JOE
         TERMINATE
TIMER    GENERATE    540
         TERMINATE   1
```

To run the simulation, we simply use the command START 1.

Note: the first TERMINATE block does not have an attribute. In the program above, after the command START 1 is issued, the simulation begins to run. Due to the fact that the TERMINATE block at Line 800 does not have an attribute, the termination counter is not decremented and the simulation continues. At 540 time units after the simulation starts, however, a single transaction will be generated by Line 900 which will decrement the termination counter to zero and stop the program.

GPSS (General Purpose System Simulation)

Example of Program in GPSS: Joe's Barbershop (Event Driven)

Suppose we want to simulate the Barbershop for 50 Transactions.

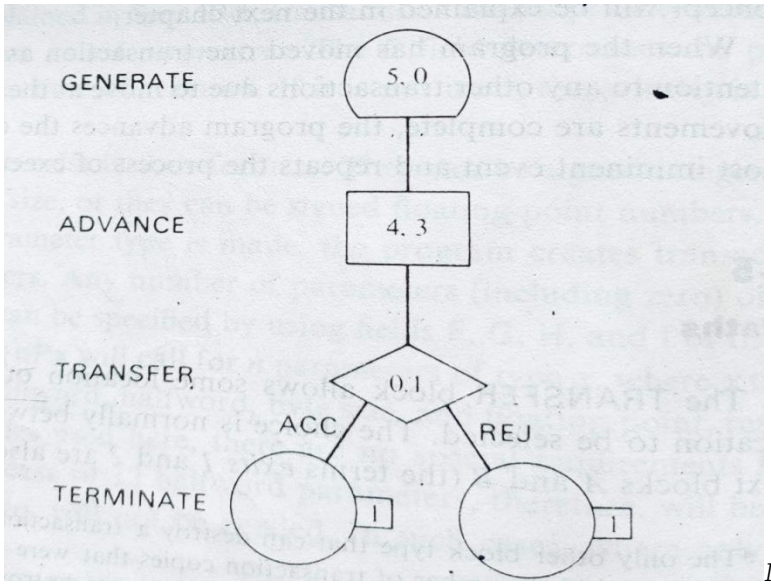
```
MEN     GENERATE    18,6
        QUEUE       SEAT
        SEIZE       JOE
        DEPART      SEAT
        ADVANCE     15,3
        RELEASE     JOE

        TERMINATE   1
```

To run the simulation, we simply use the command START 50.

GPSS EXAMPLES

1. A machine tool in a manufacturing shop is turning out parts at the rate of one every 5 minutes. As they are finished the parts go to a inspector who takes 4±3 minutes to examine each one and rejects 10% of the parts. Represent the system in GPSS
   Write GPSS program for1000 parts.

```
GENERATE 5
ADVANCE 4,3
TRANSFER 0.1 ACC,REJ
ACC TERMINATE 1
REJ TERMINATE 1
START 1000
```



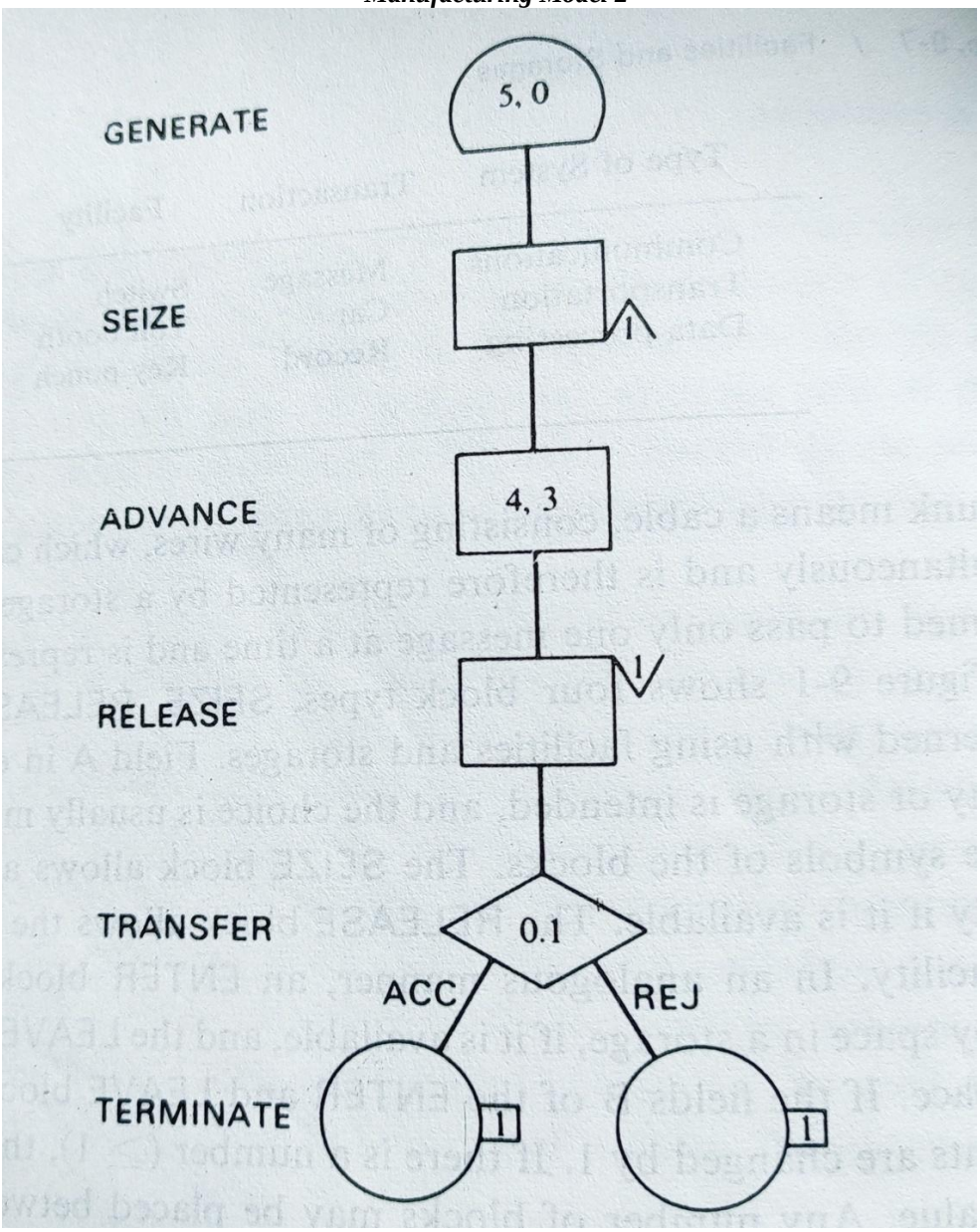
Manufacturing model 1

GPSS (General Purpose System Simulation)

2. Implement Q.No. 1 using facility.

```
GENERATE 5
SEIZE INSPECTOR
ADVANCE 4,3
RELEASE INSPECTOR
TRANSFER 0.1 ACC,REJ
ACC TERMINATE 1
REJ TERMINATE 1
START 1000
```

Manufacturing Model 2

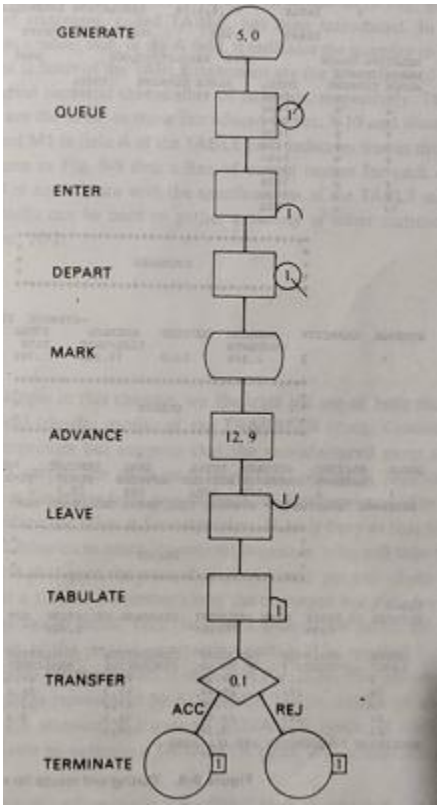


3. Implement Q.No. 1 using storage.(Here storage have 3 servers and timing parameters are changed)

```
GENERATE 5
ENTER STO,1
ADVANCE 12,9
LEAVE STO,1
TRANSFER 0.1,ACC,REJ
ACC TERMINATE 1
REJ TERMINATE 1
STO STORAGE 3
START 1000
```

Manufacturing Model 3

4. Model 4(Use of Mark and Tabulate)



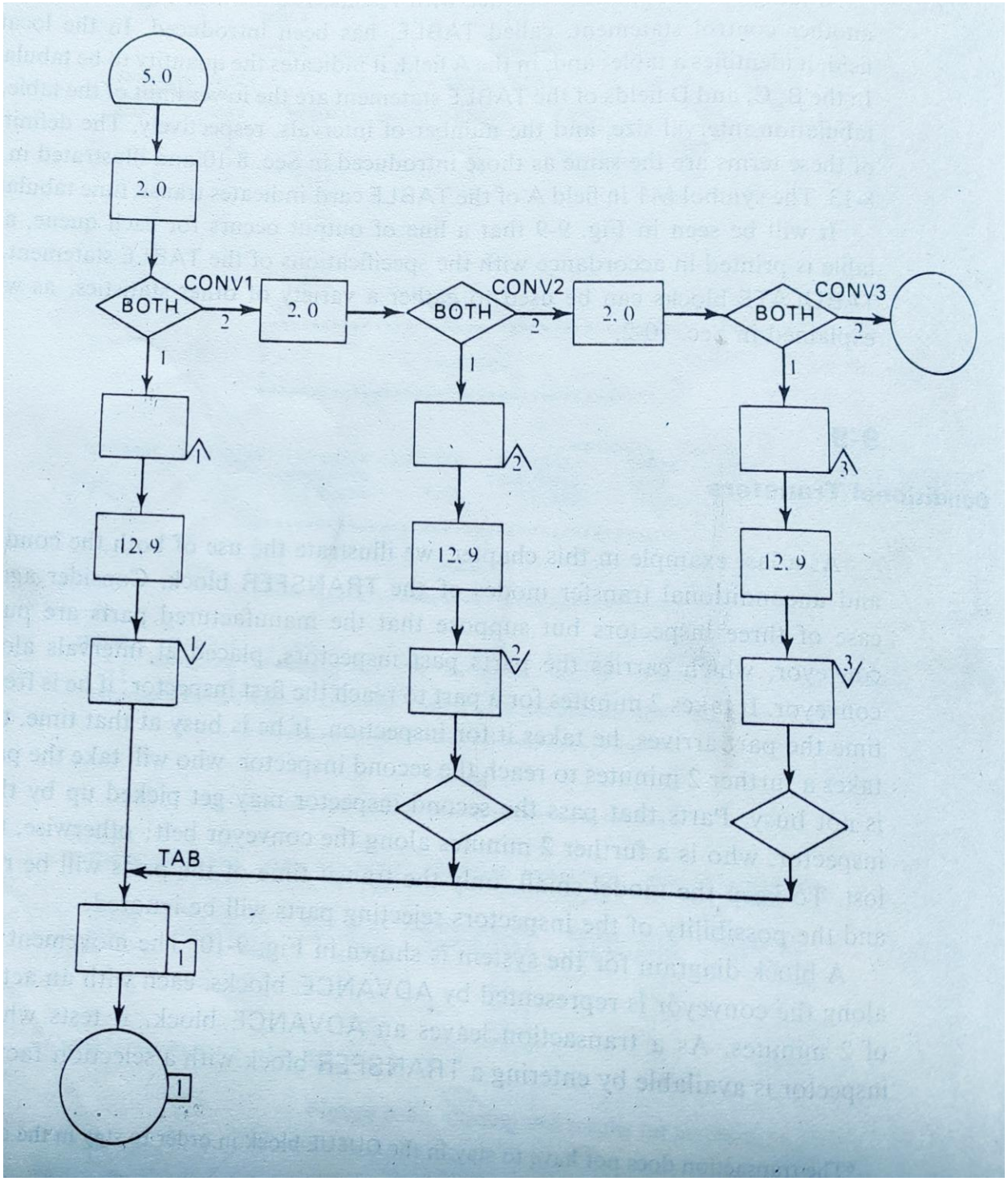

```
*MANUFACTURING SHOP MODEL 4
*
GENERATE 5
QUEUE SEAT
ENTER STO
DEPART SEAT
MARK
ADVANCE 12,9
LEAVE STO
```

GPSS (General Purpose System Simulation)

```
TABULATE TAB
TRANSFER .1 ACC,REJ
ACC TERMINATE 1
REJ TERMINATE 1
*
STO STORAGE 4
TAB TABLE M1,5,5,10
*
START 1000
```

5. Suppose that the manufactured parts are put on a conveyor, which carries the parts to part inspectors. It takes 2 minutes for a part to reach first inspector, if he is busy the pars reaches to second inspector after 2 minutes, again if the second inspector is busy it reaches to third inspector after 2 minutes and if the third is also busy the part is lost.

GPSS (General Purpose System Simulation)




```
*MANUFACTURING SHOP MODEL 5
GENERATE 5
ADVANCE 2
TRANSFER BOTH,,INS1
SEIZE 1
ADVANCE 12,9
RELEASE 1
```

GPSS (General Purpose System Simulation)

```
*MANUFACTURING SHOP
 MODEL 5
GENERATE 3
ADVANCE 2
TRANSFER BOTH,,INS1
SEIZE 1
ADVANCE 15,9
RELEASE 1
TT TABULATE TAB
TERMINATE 1
*
INS1 ADVANCE 2
TRANSFER BOTH,,INS2
SEIZE 2
ADVANCE 15,9
RELEASE 2
TRANSFER ,TT
*
INS2 ADVANCE 2
TRANSFER BOTH,,INS3
SEIZE 3
ADVANCE 15,9
RELEASE 3
TRANSFER ,TT
*
INS3 TERMINATE 1
*
TAB TABLE M1,5,5,10
*
```

6. Fuel Station Simulation: One vehicle arrives every 2±2 minutes. It takes 5±2 minutes to fuel one vehicle. Number of nozzle = 2. Fuel station operates 10 hours a day. Simulate operation of fuel station for entire day.

```
GENERATE 2,2
QUEUE LINE
ENTER STO,1
DEPART LINE
ADVANCE 5,2
LEAVE STO,1
TERMINATE
TIMER GENERATE 600
TERMINATE 1
STO STORAGE 2
START 1
```