

CHAPTER 9: AIRPORT STATUS

The program given here records airplane arrivals and gate processing at an airport. The user is first asked

- The number of gates available
- The nominal duration for servicing an airplane

Three types of airplane, a B747, an Airbus 300, and a Dash are considered as possible arriving planes. These three planes have service duration of 3 times, 2 times and 1 time of the nominal duration given, respectively. If the gates are fully occupied, a new arrival is placed in a waiting queue.

For input, each minute the user is prompted for possible arrivals.

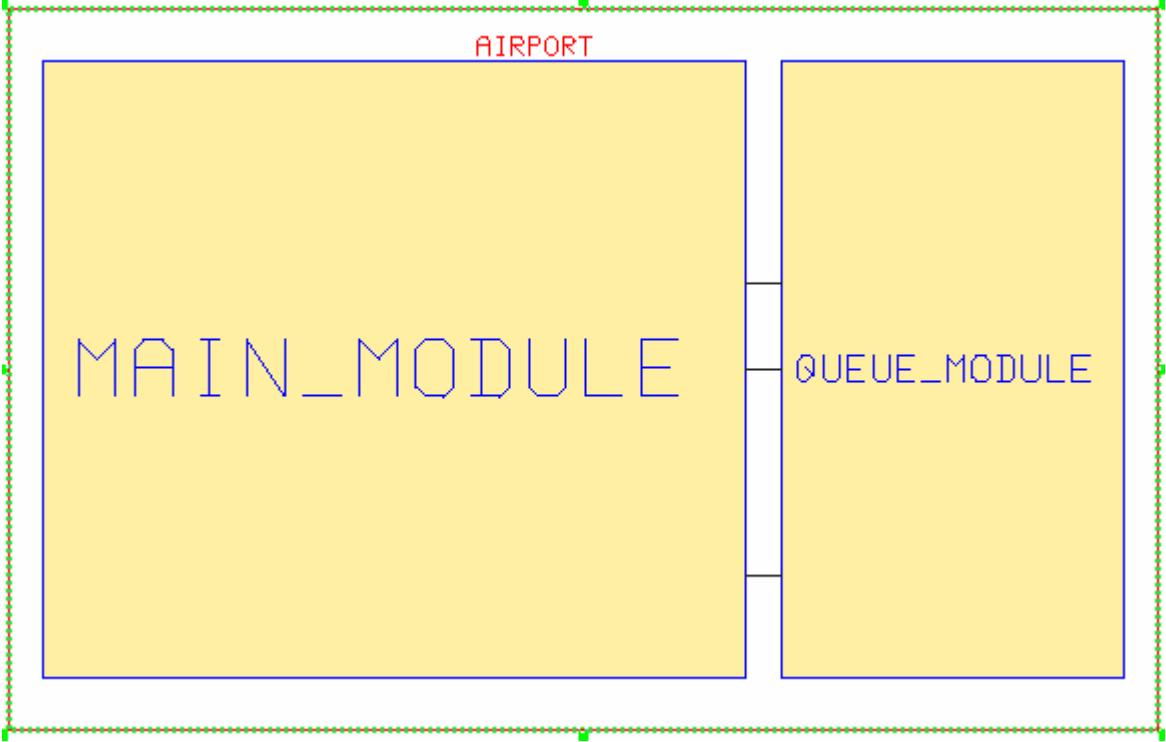
- N – indicates no arrival
- A – Airbus 300
- B – B747
- D – DASH
- X – indicates exit

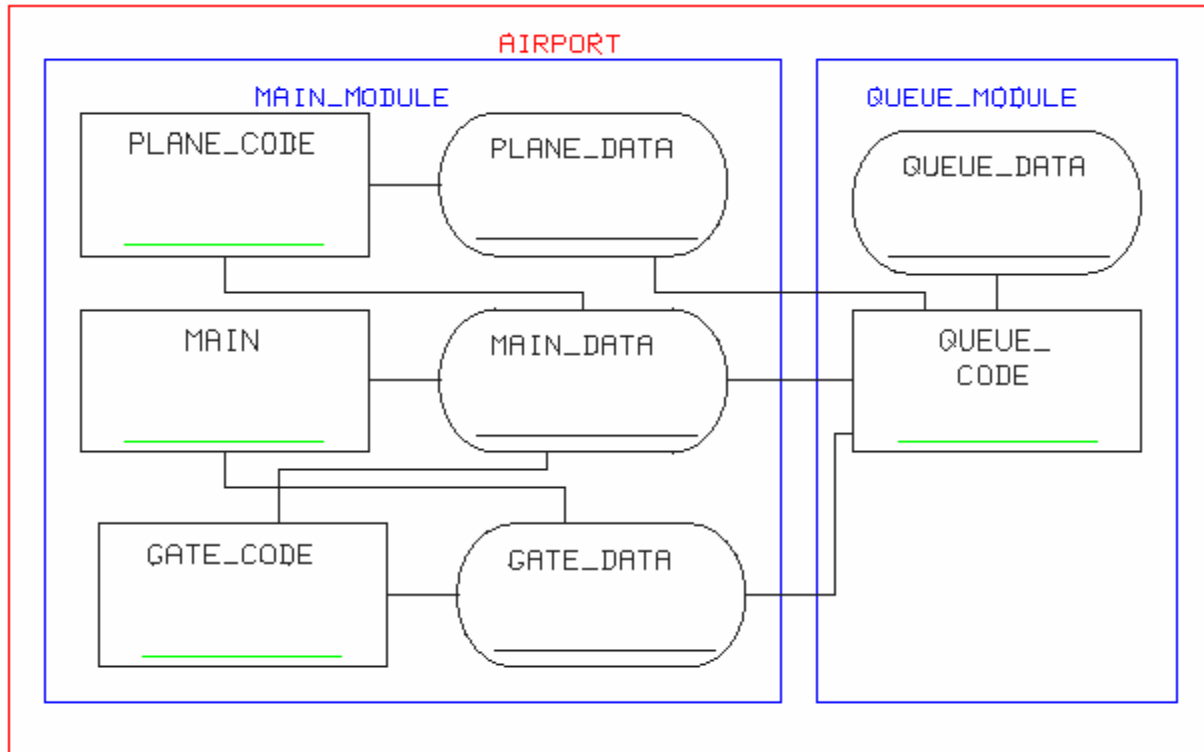
Every time the user enters a value 'A', 'B' or 'D', it represents the arrival of a plane the next minute, a value of 'N' indicates that no plane arrived the next minute.

The output is a series of statistics about the time each plane spends before service is completed, the final states of the queue and gates, and the maximum waiting period.

The arrival and servicing of different kinds of planes is recorded using various functions and logic in the program. A key item is a program counter which functions like a real clock. The counter, an integer, is incremented like the minutes in a clock. Each increment represents one minute. A real time atmosphere is created as if the planes can arrive at each minute of time. Then, given the number of available gates, a plane is sent to a gate for servicing or put into a waiting queue.

The picture below describes the program structure at a glance. This is the VSE architecture:





Architecture of Airport Program

This program involves four resources and four processes, as shown above.

The four resources are:

1. MAIN_DATA
2. PLANE_DATA
3. GATE_DATA
4. QUEUE_DATA

The four processes are:

1. MAIN
2. PLANE_CODE
3. GATE_CODE
4. QUEUE_CODE

The QUEUE_DATA and QUEUE_CODE form an independent part of the program. The QUEUE_DATA and QUEUE_CODE perform the function of a real-life queue, a kind of “abstract data type” in technical terms.

The data related to planes and gates are declared as arrays of size 40 in PLANE_DATA and GATE_DATA respectively. In a similar manner the data related to QUEUE is declared as array in QUEUE_DATA.

Execution starts with the INITIALIZE process, where initial values are set and the user is prompted for the number of gates and the nominal service duration. Control is passed to RUN_PLANE_STATUS, and eventually SHOW_FINAL_STATE and SHOW_STATISTICS at the end of the session.

In the PLANE_CODE process, we first check for the arrival of new plane, and if there is a new arrival, we note the kind of plane that has arrived and calculate the duration of service that is required.

The GATE_CODE module checks which gates are open, and changes a plane from the queue into the gates opened. First it checks if a plane at a particular gate has completed its service duration; if it has, that particular gate is set to open. Every time a plane is moved to a gate, that gate is set to closed. Then this module calls the QUEUE_CODE and updates MAX_QUEUE_TIME and TOTAL_NO_WAIT if appropriate.

In the QUEUE_CODE module, every plane that arrives is first placed in the queue and the plane is later moved from the queue to an open gate, and every time a plane is placed in the queue, QUEUE_COUNT is incremented by 1. This module adds the planes to the queue as and when they arrive. According as the gates are open or not, the planes are either sent to a gate or kept in the queue. Whenever a plane enters a gate from the queue, the queue count is decreased by 1. Each plane is sent to a gate from the information obtained from CHECK_GATES. Thus in this module, at the end of the session we have all planes which still remain in the queue.

A sample run is given below:

```
AIRPORT
ENTERING VSE TASK      AIRPORT...
Enter Number of Gates:
> 4
Enter Nominal Service Duration:
> 1
Enter N to indicate there is no arriving plane
Enter B for B747
Enter A for Airbus 300
Enter D for DASH
Enter X to end the input
Enter N,B,A,D,or X:
> A
Enter N,B,A,D,or X:
> B
Enter N,B,A,D,or X:
> D
Enter N,B,A,D,or X:
> N
Enter N,B,A,D,or X:
> X
Total number of arriving planes:          3
There are no aircraft in the queue.
The following aircraft are still at a gate:
      2
Average service duration in minutes
      2.3
Maximum waiting time
      0
AIRPORT COMPLETED
RESULTS IN AIRPORT.LIS
HIT ENTER TO CONTINUE
□
```

*****Data for the Resource: MAIN_DATA**

0001	MAIN_DATA		
0002	1	PROG_CLOCK	INTEGER
0003	1	NUM_ARRIVAL	INTEGER
0004	1	PLANE_COUNT	INTEGER
0005	1	SERVICE_DURATION	INTEGER
0006	1	TOTAL_SERV_DURATION	INTEGER
0007	1	AVE_SERV_DURATION	DECIMAL Z(2)9.99
0008	1	MAX_GATES	INTEGER
0009	1	QUEUE_TIME	INTEGER
0010	1	MAX_QUEUE_TIME	INTEGER
0011	1	TOTAL_NO_WAIT	INTEGER
0012	1	NUM_OPEN_GATES	INTEGER
0013	1	NUM_OF_PLANES	INTEGER
0014	1	QUEUE_COUNT	INTEGER
0015	1	PLANE_MOVED	INTEGER
0016	1	GATE_NUMBER	INTEGER
0017	1	PLANE_SERVICED	INTEGER
0018	1	DURATION	INTEGER
0019	1	AVERAGE_DURATION	REAL
0020	1	I	INTEGER
0021	1	J	INTEGER
0022	1	QUEUE_ID	INTEGER
0023	1	PLANE_NUMBER	INTEGER
0024	1	PLANES_ARRIVED	INTEGER
0025	1	ARRIVAL_PLANE	CHAR QUANTITY(50)
0026	1	PLANE_NAME	CHAR 11
0027	1	CALL_FROM_ARRIVE	STATUS TRUE,FALSE
0028	1	CALL_FROM_GATES	STATUS TRUE,FALSE
0029	1	CALL_FROM_FINAL_STATE	STATUS TRUE,FALSE
0030	1	NEW_ARRIVAL	STATUS TRUE,FALSE

*****Data for Resource: AIRPLANE_INFO**

0031	AIRPLANE_INFO		
0032	1	AIRPLANE	QUANTITY(40)
0033	2	PLANE_ID	INTEGER
0034	2	TIME_ENTER_QUEUE	INTEGER
0035	2	TIME_LEAVE_QUEUE	INTEGER
0036	2	DURATION_SERVED	INTEGER
0037	2	KIND	CHAR 11

*****Data for Resource: GATE_DATA**

0038	GATE_INFO		
0039	1	GATE	QUANTITY(40)
0040	2	PLANE_ID	INTEGER

0041	2	TIME_ENTER_QUEUE	INTEGER
0042	2	TIME_ARRIVE_GATE	INTEGER
0043	2	DURATION_SERVED	INTEGER
0044	2	TIME_COMPLETED	INTEGER
0045	2	MINS_UNUSED	INTEGER
0046	2	OPENED	STATUS OPEN,CLOSED

*****Data for Resource: QUEUE_DATA**

0047	QUEUE_INFO		
0048	1	QUEUE	QUANTITY(40)
0049	2	PLANE_ID	INTEGER
0050	2	TIME_ENTER_QUEUE	INTEGER
0051	2	DURATION_SERVED	INTEGER
0052	2	TIME_ARRIVE	INTEGER

*****Data for Process: MAIN**

```

0003 MAIN
0004     EXECUTE INITIALIZE
0005     EXECUTE RUN_PLANE_STATUS
0006     EXECUTE SHOW_FINAL_STATE
0007     EXECUTE SHOW_STATISTICS.

0008 INITIALIZE
0009     PROG_CLOCK      = 0
0010     PLANE_COUNT    = 0
0011     QUEUE_TIME     = 0
0012     MAX_QUEUE_TIME = 0
0013     TOTAL_NO_WAIT  = 0
0014     QUEUE_COUNT    = 0
0015     PLANES_ARRIVED = 0
0016     QUEUE_ID = 0
0017     I = 1
0018     J = 1
0019     NUM_OF_PLANES = 0
0020     PLANE_NUMBER = 1
0021     SET CALL_FROM_ARRIVE      TO FALSE
0022     SET CALL_FROM_GATES      TO FALSE
0023     SET CALL_FROM_FINAL_STATE TO FALSE
0024     SET NEW_ARRIVAL          TO FALSE
0025     DISPLAY 'Enter the Number of Gates Open: '
0026     ACCEPT NUM_OPEN_GATES
0027     DISPLAY 'Enter Nominal Duration: '
0028     ACCEPT SERVICE_DURATION
0029     EXECUTE VALID_INPUT.
0030     EXECUTE INITIAL_LOOP NUM_OPEN_GATES TIMES.
0031     I = 1.
0032     EXECUTE GET_INPUT UNTIL ARRIVAL_PLANE(I) IS EQUAL TO 'X'.

0033 GET_INPUT
0034     DISPLAY 'Enter N to indicate there is no arriving plane'
0035     DISPLAY 'Enter B for B747'
0036     DISPLAY 'Enter A for Airbus 300'
0037     DISPLAY 'Enter D for DASH '
0038     DISPLAY 'Enter X to end the input'
0039     DISPLAY ' '
0040     DISPLAY 'Enter N,A,B,D,OR X:'

0041 INITIAL_LOOP
0042     GATE PLANE_ID(I) = 0
0043     GATE TIME_ENTER_QUEUE(I) = 0
0044     GATE TIME_ARRIVE_GATE(I) = 0
0045     GATE DURATION_SERVED(I) = 0

```

```

0046     GATE TIME_COMPLETED(I)  = 0
0047     SET  GATE OPENED(I) TO OPEN
0048     INCREMENT I

0049 INPUT_PLANES
0050     ACCEPT ARRIVAL_PLANE(I)

0051     IF ARRIVAL_PLANE(I) IS EQUAL TO 'A' OR
0052     ARRIVAL_PLANE(I) IS EQUAL TO 'B' OR
0053     ARRIVAL_PLANE(I) IS EQUAL TO 'D'
0054     INCREMENT PLANES_ARRIVED.

0055     IF ARRIVAL_PLANE(I) IS EQUAL TO 'A' OR
0056     ARRIVAL_PLANE(I) IS EQUAL TO 'B' OR
0057     ARRIVAL_PLANE(I) IS EQUAL TO 'D' OR
0058     ARRIVAL_PLANE(I) IS EQUAL TO 'N'
0059     INCREMENT I
0060     ELSE IF (ARRIVAL_PLANE(I) NOT EQUAL TO 'X'
0061     AND ARRIVAL_PLANE(I) NOT EQUAL TO 'N')
0062     DISPLAY 'Not a Valid Plane. Please enter N,A,B,D,OR X:'.

0063     IF ARRIVAL_PLANE(I) NOT EQUAL TO 'X'
0064     DISPLAY 'Enter N,A,B,D,OR X:'.

0065 RUN_PLANE_STATUS
0066     DECREMENT I.
0067     NUM_OF_PLANES = I.
0068     EXECUTE START_RUN NUM_OF_PLANES TIMES.

0069 START_RUN
0070     CALL PLANE_CODE.
0071     CALL GATE_CODE.
0072     INCREMENT PROG_CLOCK.
0073     INCREMENT PLANE_NUMBER.

0074     IF PROG_CLOCK IS GREATER THAN 0
0075     AND MOD(PROG_CLOCK,10) IS EQUAL TO 0
0076     DISPLAY PROG_CLOCK 'Minute(s) Passed. ' .

0077 SHOW_FINAL_STATE
0078     DISPLAY 'Total number of planes arrived:' PLANES_ARRIVED
0079     IF QUEUE_COUNT IS GREATER THAN 0
0080     DISPLAY 'The following aircraft are in queue:'
0081     SET CALL_FROM_FINAL_STATE TO TRUE
0082     CALL QUEUE_CODE
0083     ELSE

0084     DISPLAY 'There are no aircraft in the queue.'.

0085     DISPLAY 'The following aircraft are still at a gate: '
0086     I = 1
0087     EXECUTE DISPLAY_GATE_INFO NUM_OPEN_GATES TIMES.

0088 DISPLAY_GATE_INFO
0089     IF GATE OPENED(I) IS CLOSED
0090     DISPLAY GATE PLANE_ID(I).
0091     INCREMENT I

0092 SHOW_STATISTICS
0093     AVERAGE_DURATION = TOTAL_SERV_DURATION/PLANES_ARRIVED.
0094     MOVE AVERAGE_DURATION TO AVE_SERV_DURATION.
0095     DISPLAY 'Average service duration in minutes'
0096     DISPLAY AVE_SERV_DURATION.
0097     DISPLAY ' '.
0098     DISPLAY 'Maximum waiting time'
0099     DISPLAY MAX_QUEUE_TIME .

```

***Data for Process: PLANE_CODE

```

0102 PLANE_CODE
0103 EXECUTE NEW_ARRIVAL

0104 IF NEW_ARRIVAL IS TRUE
0105 INCREMENT PLANE_COUNT
0106 AIRPLANE PLANE_ID(PLANE_COUNT) = PLANE_COUNT
0107 AIRPLANE TIME_ENTER_QUEUE(PLANE_COUNT) = PROG_CLOCK

0108 IF ARRIVAL_PLANE(PLANE_NUMBER) IS EQUAL TO 'B'
0109 MOVE 'B747' TO AIRPLANE KIND(PLANE_COUNT)
0110 AIRPLANE DURATION_SERVED(PLANE_COUNT) = SERVICE_DURATION * 3
0111 TOT_SERV_DURATION = TOT_SERV_DURATION +
0112 AIRPLANE DURATION SERVED(PLANE COUNT)
0113 ELSE IF ARRIVAL_PLANE(PLANE_NUMBER) IS EQUAL TO 'A'
0114 MOVE 'Airbus 300' TO AIRPLANE KIND(PLANE_COUNT)
0115 AIRPLANE DURATION_SERVED(PLANE_COUNT) = SERVICE_DURATION * 2
0116 TOT_SERV_DURATION = TOT_SERV_DURATION +
0117 AIRPLANE DURATION SERVED(PLANE COUNT)
0118 ELSE IF ARRIVAL_PLANE(PLANE_NUMBER) IS EQUAL TO 'D'
0119 MOVE 'DASH' TO AIRPLANE KIND(PLANE_COUNT)
0120 AIRPLANE DURATION_SERVED(PLANE_COUNT) = SERVICE_DURATION.
0121 TOT_SERV_DURATION = TOT_SERV_DURATION +
0122 AIRPLANE DURATION_SERVED(PLANE_COUNT)

0123 IF NEW_ARRIVAL IS TRUE
0124 MOVE AIRPLANE KIND(PLANE_COUNT) TO PLANE_NAME
0125 DURATION = AIRPLANE DURATION_SERVED(PLANE_COUNT)
0126 INCREMENT QUEUE_COUNT
0127 SET CALL_FROM_ARRIVE TO TRUE
0128 CALL QUEUE_CODE.

0129 NEW_ARRIVAL
0130 IF ARRIVAL_PLANE(PLANE_NUMBER) IS NOT EQUAL TO 'N'
0131 SET NEW_ARRIVAL TO TRUE
0132 ELSE
0133 TOT_SERV_DURATION = TOT_SERV_DURATION + 1
0134 SET NEW_ARRIVAL TO FALSE.

```

*****Data for Process: CHECK_GATES**

```

0137 GATE_CODE
0138 EXECUTE CHECK_GATES.
0139 EXECUTE ADD_TO_GATE.

0140 CHECK_GATES.
0141 I = 1
0142 EXECUTE COMPLETE_SERVICE NUM_OPEN_GATES TIMES.

0143 COMPLETE_SERVICE
0144 IF GATE TIME_COMPLETED(I) IS EQUAL TO PROG_CLOCK
0145 SET GATE OPENED(I) TO OPEN
0146 IF PROG_CLOCK IS GREATER THAN 0
0147 PLANE_SERVICED = GATE PLANE_ID(I).
0148 INCREMENT I

0149 ADD_TO_GATE
0150 GATE_NUMBER = 1
0151 EXECUTE GO_GATE NUM_OPEN_GATES TIMES.

0152 GO_GATE
0153 IF GATE OPENED(GATE_NUMBER) IS OPEN
0154 AND QUEUE_COUNT IS GREATER THAN 0
0155 SET CALL_FROM_GATES TO TRUE
0156 CALL QUEUE_CODE
0157 INCREMENT J.

0158 IF GATE OPENED(GATE_NUMBER) IS OPEN

```

```

0159     AND QUEUE_COUNT IS GREATER THAN 0
0160     IF QUEUE_TIME GREATER THAN MAX_QUEUE_TIME
0161         MAX_QUEUE_TIME = QUEUE_TIME.

0162     IF GATE_OPENED(GATE_NUMBER) IS OPEN
0163     AND QUEUE_COUNT IS GREATER THAN 0
0164     DECREMENT QUEUE_COUNT
0165     SET GATE_OPENED(GATE_NUMBER) TO CLOSED
0166     IF QUEUE_TIME EQUALS ZERO
0167         INCREMENT TOTAL_NO_WAIT.

0168     INCREMENT GATE_NUMBER

```

*****Data for Process: QUEUE_CODE**

```

0171 QUEUE_CODE
0172     EXECUTE PROCESS_QUEUE

0173 PROCESS_QUEUE
0174     IF CALL_FROM_ARRIVE IS TRUE
0175     EXECUTE PLANE_QUEUE
0176     SET CALL_FROM_ARRIVE TO FALSE.

0177     IF CALL_FROM_GATES IS TRUE
0178     EXECUTE QUEUE_TO_GATE
0179     SET CALL_FROM_GATES TO FALSE.

0180     IF CALL_FROM_FINAL_STATE IS TRUE
0181     EXECUTE QUEUE_DATA QUEUE_COUNT TIMES.

0182 PLANE_QUEUE
0183     INCREMENT QUEUE_ID
0184     QUEUE_PLANE_ID(QUEUE_ID) = AIRPLANE_PLANE_ID(PLANE_COUNT)
0185     QUEUE_DURATION_SERVED(QUEUE_ID) = DURATION
0186     QUEUE_TIME_ENTER_QUEUE(QUEUE_ID) =
0187         AIRPLANE_TIME_ENTER_QUEUE(PLANE_COUNT).

0188 QUEUE_TO_GATE
0189     PLANE_MOVED = QUEUE_PLANE_ID(J)
0190     GATE_PLANE_ID(GATE_NUMBER) = QUEUE_PLANE_ID(J)
0191     GATE_TIME_ENTER_QUEUE(GATE_NUMBER) = QUEUE_TIME_ENTER_QUEUE(J)
0192     GATE_TIME_ARRIVE_GATE(GATE_NUMBER) = PROG_CLOCK
0193     GATE_TIME_COMPLETED(GATE_NUMBER) =
0194         GATE_TIME_ARRIVE_GATE(GATE_NUMBER)
0195         + QUEUE_DURATION_SERVED(J)
0196     QUEUE_TIME = GATE_TIME_ARRIVE_GATE(GATE_NUMBER)
0197         - GATE_TIME_ENTER_QUEUE(GATE_NUMBER)

0198 QUEUE_DATA

```

```
0199   DISPLAY QUEUE PLANE_ID(QUEUE_ID)
0200   DECREMENT QUEUE_COUNT
0201   DECREMENT QUEUE_ID.
```