

Question 23 (20 marks) Use a SEPARATE writing booklet.

Use the information below and the algorithm on page 16 to answer Question 23.

An event ticket agency sells tickets for theatre, cinema and pop concerts. You have been employed to rectify errors in their ticketing system.

The goal of the ticketing system is to accept messages from remote terminals that include transaction details relating to events and ticket sales. The system collects the transactions and stores the relevant data in TICKET and EVENT files.

Following a ticket sale, tickets are printed at the remote terminal with a unique ticket number and the date and time of printing.

At the end of the day a supervisor concludes the processing by entering SystemStatus equals 'OFF'.

Question 23 continues on page 16

Question 23 (continued)

The algorithm for the ticketing system is:

```
1   BEGIN TicketSystem
2     SystemStatus = 'ON'
3     READ LastEventNumber
4     READ SystemDate, SystemTime
5     WHILE SystemStatus = 'ON'
6       READ MessageHeader
7       IF MessageHeader = 'EVENT' THEN
8         ProcessEvent
9       ENDIF
10      IF MessageHeader = 'TICKET' THEN
11        ProcessTicket
12      ELSE
13        SystemStatus = 'OFF'
14      ENDIF
15      READ SystemDate, SystemTime
16    END WHILE
17  END TicketSystem

18  BEGIN ProcessEvent
19    EventNumber = LastEventNumber + 1
20    LastEventNumber = EventNumber
21    STORE LastEventNumber
22    OPEN EVENT
23    STORE (EventName, EventDate, EventTime, EventLocation,
24          EventTicketsAvailable, EventNumber)
25    CLOSE EVENT
26  END ProcessEvent

27  BEGIN ProcessTicket
28    OPEN EVENT
29    IF EventTicketsAvailable < > 0 THEN
30      EventTicketsAvailable = EventTicketsAvailable – NumberTicketsSold
31    ENDIF
32    CLOSE EVENT
33    OPEN TICKET
34    STORE (TicketEventNumber, TicketEventName, NumberTicketsSold,
35          TicketEventDate, TicketEventTime)
36    FOR PrintTicket = 1 to NumberTicketsSold
37      PRINT (TicketCurrentTicketNumber, TicketEventName,
38            TicketEventDate, TicketEventTime, SystemDate, SystemTime)
39    ENDFOR
40    CLOSE TICKET
41  END ProcessTicket
```

Question 23 continues on page 17

Question 23 (Continued)

- (a) There are a number of errors in this algorithm. One error is as follows:

If the program reads an unrecognisable message, then the program does not perform as expected. This error occurs between lines 7 and 16 of the algorithm.

- (i) Identify this error in the algorithm. **2**
 - (ii) Modify the appropriate line(s) of the algorithm to solve this error. **3**
 - (iii) There are three classes of messages involved with the operation of this ticketing system. These messages are passed from the remote terminals to the main module. **2**
 What is the data type of each message class?
 - (iv) Using a desk check, identify TWO further errors in the algorithm. **3**
- (b) The users of the remote terminals produce end-of-day reports listing the number of tickets sold for each event. The TICKET and EVENT files will be numerically sorted by ticket number and event number prior to processing. **6**

Write an algorithm to produce a report. The algorithm must:

- create an array of records (TicketArray) and insert the number of tickets sold, and the name and number of each event;
- sort the array by event number; and
- calculate and print the number of tickets sold and the event name for each event for the day.

- (c) The ticket agency is currently reviewing the manner in which it provides documentation for the users of the remote terminals. **4**

After a study of the users it has been determined that they need support to:

- learn the application;
- learn specific functions in the application; and
- understand the application and its function and purpose.

The ticket agency wants to ensure that its user documentation meets these needs and is considering three alternative formats: paper, online documentation and video documentation.

Briefly describe TWO of these user documentation formats and evaluate the suitability of ONE of the formats described for the ticket agency.

End of Question 23



Q23 (a) (i)

The error is found in the "IF" statements ^{header} between lines 7 and 16. If the ~~input~~ is not 'event' or 'ticket', then the ~~sys~~ system turns off (system status = 'OFF').

This process is only meant to happen when the supervisor inputs "system status equals 'OFF'". Hence, the algorithm ~~en~~ does not function as expected. This can be fixed by validating the data (Q 23 a ii).



(23)(0) (ii)

Change the IF^{selectin} structure to a multi-way structure:

6 READ MessageHeader

7 CASEWHEN MessageHeader =

8 "EVENT" : ProcessEvent

9 "TICKET" : ProcessTicket

10 "systemstatus equals off" : SystemStatus = "OFF"

11 OTHERWISE : OUTPUT "Inappropriate header. Please
re-input header"

12 ENDCASE

13 Read SystemDate, SystemTime

etc.



23(a) (iii)

The "systemstatus=OFF" message is a string
The "EVENT" and "TICKET" messages are records

(iv) Desk-checking lines 29 to 31 give us:

Number Tickets Sold	Event Tickets Available
2	1
2	-1 ... continues infinitely!
2	2
	0 stops!
2	3
	1 ... can continue!

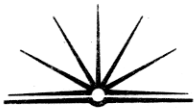
From this, we can see that if Event Tickets Available is less than 0, the processes can still occur ^{but} (it is impossible to sell tickets if there are none left!). Hence, it is an error. This can be fixed by ensuring Number Tickets Sold is ~~always~~ ^{never} larger than Event Tickets Available and changing line 29 to "IF Event Tickets Available > 0 THEN"



Deskchecking line ~~32~~ 36-38 gives us:

Print Ticket	Number Tickets Sold	Ticket Current Ticket Number
1	4	1000
2		1000
3		1000
4		1000

As shown above, the ticket number does not change (Ticket Current Ticket Number) remains as the same number. Hence, every ticket will NOT be unique, resulting in another error. This can be fixed by incrementing the Ticket Current Ticket Number variable at the end of the "For" loop. (between lines 38 and 39).



223

(b) BEGIN. Report.

Length is the total number of events.
Ticket Array is an Array of Ticket Records,
indexed from 1 to ~~last event number~~. ~~Number tickets~~ (E) Length sold!

Ticket is a record, containing,

Number ~~tickets~~ sold, an integer.

Event Name, a string

Event Number, an integer

END

FOR index goes from 1 to ^{Length} ~~Number tickets sold~~

OPEN TICKET

STORE ~~Event~~ Ticket Event Number in ~~Array~~ Ticket Array (index). Event Number

STORE Ticket Event Name in Ticket Array (index). Event Name

STORE Number tickets sold in Ticket Array (index). Number sold

CLOSE ~~ticket~~ TICKET.

ENDFOR.

Sort Array (Ticket Array)

Display Array (Ticket Array)

END Report.



(b)

```
BEGIN SortArray (Ticket Array)
```

```
Set low count to length of Ticket Array.
```

```
FOR index goes from 2 to count.
```

```
FOR index goes from 2 to count
```

```
IF Ticket Array (index).eventnumber >
```

```
Ticket Array (index (index+1)) .eventnumber THEN
```

```
Swap (Ticket Array (index), Ticket Array (index+1))
```

```
ENDIF.
```

```
ENDFOR
```

```
ENDFOR
```

```
Return Ticket Array
```

```
END SortArray
```

```
BEGIN Swap (Ticket (index), Ticket Array (index+1)).
```

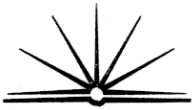
```
Temp = Ticket (index).
```

```
Ticket (index) = Ticket (index+1)
```

```
Ticket (index+1) = Temp
```

```
Return (Ticket (index), Ticket (index+1))
```

```
END Swap
```



(6)

BEGIN Display Array (^{Ticket}Array)

~~FOR~~ Set Count to Length of ^{Ticket} Array

FOR index goes from 1 to Count

~~Number~~ Tickets = 0.

~~Ticket~~ = ^{Ticket}Array(index). Numbr ~~red~~ sold + Tickets.

IF ^{Ticket}Array(index). event number = ^{Ticket}Array(index+1). event
number THEN
Tickets = ^{Ticket}Array(index+1). Numbr sold + Tickets

ELSE

○ DISPLAY ^{Ticket}Array(index). event name, ^{Ticket}Array(index). event
number, ~~Array~~ Tickets

ENDIF

ENDFOR.

END Display Array



23

(c) Briefly describe one video and paper documentation ~~of paper~~ ~~down~~ Paper documentation involves hardcopy documents (i.e. printed on paper). This includes ^{or} manuals, ~~low~~ system manuals, troubleshooting guides, installation guides and introductory manuals). Paper documentation lacks a dynamic nature as it must be reprinted & redistributed if it is to be updated.

Video documentation is in the next booklet...

Online documentation involves softcopy documents (i.e. viewed on the screen). These would include HTML documents. They allow for ~~interactivity~~ interactivity (using hypertext), a variety of media types (use of audio, video, images and text) to promote ~~understanding~~ understanding and maintain interest. They have a very ~~dynamic~~ dynamic nature and allow for fast data retrieval. (Every word is indexed and it stores previous search paths and keywords). It is convenient because it allows the person to view the documentation whilst at the computer.

Next booklet...

(c).

They can be easily updated in the sense that the newer version can easily replace the older version and it is very portable. Text from online documentation can be copied into a word file and stored for later reference. Balloon text provides information about screen elements and tutorial-type assistance help users to complete information processes. However, despite the benefits, there are costs ~~there~~ there must be more storage space and faster processing and the development process is lengthened. Hence, the benefits ~~outweigh~~ outweigh the costs and ~~can~~ thus online documentation is very suitable.

Next page for video documentation...



(23) (C)

Video documentation involves documents in a VHS format that can be played by a VCR. (It is ~~a~~ in an analogue video format). They help to explain concepts easily by providing a graphical ~~way~~ representation of various features. However, they are sequential in order of access and hence lack interactivity.