2001 SPRING

Software Design and Development Engineer Examination (Afternoon, Part 1)

Questions must be answered in accordance with the following:

Question Nos.	Q1-Q6
Question Selection	All questions are compulsory
Examination Time	13:00-15:00 (120 minutes)

Instructions:

1. Use an HB pencil. If you need to change an answer, erase your previous answer completely and neatly. Wipe away any eraser debris.
2. Mark your examinee information and test answers in accordance with the instructions below. Your test will not be graded if you do not mark properly. Do not mark or write on the answer sheet outside of the prescribed places.
   (1)Examinee Number
   Write your examinee number in the space provided, and mark the appropriate space below each digit.
   (2)Date of Birth
   Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.
   (3) Write each answer in the space specified for that question.
   (4)Write your answers clearly and neatly. Answers that are difficult to read will receive a lower score.
3. After the test, you may take this question booklet home with you.

Do not open the exam booklet until instructed to do so.
Inquiries about the exam questions will not be answered.

Q1. Read the following description of a communications network and answer Sub-Questions 1 through 3.

The network in Company S is made up of three branch lines, LAN 1 through 3, and a primary trunk line and a subordinate. The branch lines operate at a transmission rate of 10 Mbps, with which PCs, servers, and printers, etc. are connected. Any of these will be hereinafter referred to as a PC. The trunk lines operate at a transmission rate of 100 Mbps and are connected to each branch via a router. The subordinate trunk line is a standby. Usually the primary trunk line is used. (See figure below.)

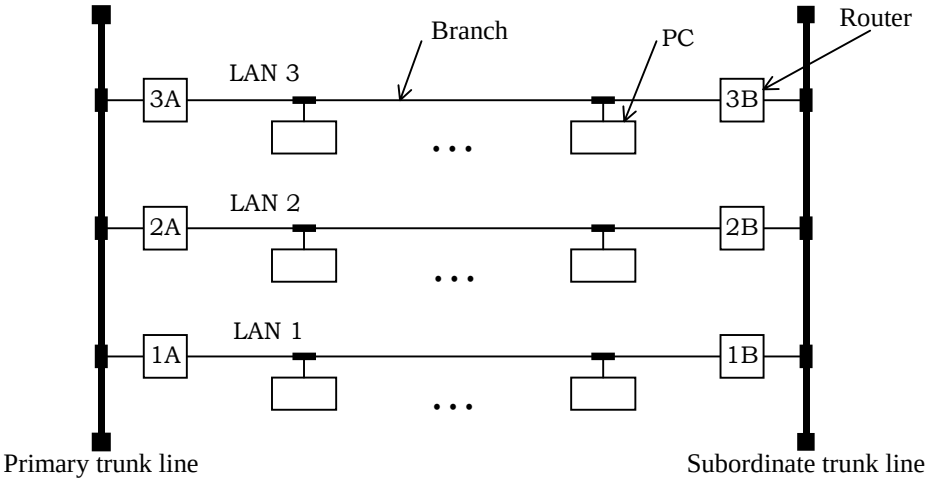


Fig. Network in Company S

Routers exchange routing information with each other, and each router constantly monitors routing information which is transmitted from other routers. If there is no notification from a router for a specified interval of time, that router is assumed to be out of order.

The hop count, which represents the distance on the network as integers, is included in the routing information exchanged between routers. The hop count is the least number of routers that the data must pass through after it first enters the router from the branch line until it reaches another branch line. The routers which the data is to pass through at this time are selected so that data cannot return through the branch line or trunk line through which it has already passed.

If a PC on a branch line communicates with a PC on another branch line, the PC selects the router that offers communications with the smallest hop count. For this reason, the router notifies routing information to the PC on the branch connected to that router. If all the routers are operating normally, only the primary routers (1A, 2A, and 3A) notify the PC of routing information, while the subordinate routers (1B, 2B, and 3B) make no notification. However, if the failure of any router on the primary line is detected, subordinate routers

begin notification.

Routing information that the router notifies to the PC is denoted in the form: “Name of Target Branch Line: Hop Count.” For example, if all the routers are operating normally, the routing information notified to a PC on LAN 1 by Router 1A is LAN2:2 and LAN3:2.

The average amount of data transferred between branch lines (and inside each branch line) is shown in Table 1.

Table 1 Average Amount of Data Between Branch Lines

Units: Mbps

Branch Line	To LAN 1	To LAN 2	To LAN 3
From LAN 1	0.3	1.0	0.6
From LAN 2	0.1	1.5	0.2
From LAN 3	0.5	1.5	0.2

Sub-Question 1

Table 2 below gives the average amount of data on each branch line. What are the correct values to insert in boxes [a] through [c]?

Table 2 Average Amount of Data on the Branch Line

Units: Mbps

Branch	Average Amount of Data
LAN 1	[a]
LAN 2	[b]
LAN 3	[c]

Sub-Question 2

What are the correct values to insert in boxes [d] through [g] in the following description regarding routing information?

If Router 1A fails, after a specified interval of time subordinate routers begin notifying routing information to PCs on each branch line.

At this time, the routing information notified to the PC on LAN 2 by Router 2A is LAN 1: [d] and LAN 3: [e]. Similarly, the routing information notified to the PC on LAN 2 by Router 2B is LAN 1: [f] and LAN 3: [g].

Sub-Question 3

At one time Router 1A and another router failed, and response worsened on some branch lines. The following description is part of the reason for the worsened response when this happened.

At the end of the following description about the worsened response, select the correct router names to place in boxes [h] and [i] and the correct branch line number to place in [j].

Note that response becomes suddenly worse on this network when the amount of data exceeds 5 Mbps on a branch line or 40 Mbps on a trunk line.

If Router 1A fails after a specified interval of time, the routing subordinate routers begin notifying routing information to PCs on the branch lines. The result is that subordinate trunk lines may be used. Therefore, even if routers such as 2A and 3A on the main line fail, no communication problems will occur because the subordinate system is operational.

When communication is performed from one branch line to another branch line as target, if only one router has failed, it is possible to communicate along a route made up of "sending branch line → trunk line→ target branch line." However, if a primary system router on a branch line and a subordinate router on another branch line fail, communication between some branch lines is handled along a route made up of "sending branch line → trunk line → bypass branch line→ trunk line → target branch line."

It is conceivable that , in addition to Router 1A, a router on the subordinate line also failed, which caused a large amount of data to flow over a branch line used as bypass, resulting in worsening the response time.

If Router [h] fails, all the data heading from LAN 1 to LAN 2 and from LAN 2 to LAN 1 bypasses LAN 3. Similarly, if Router [i] fails, all the data heading from LAN 1 to LAN 3 and from LAN 3 to LAN 1 bypasses LAN 2. Due to this fact, it is considered that the amount of data on LAN [j] exceeded 5 Mbps, thus worsening the response time.

Q2. Read the following description of object-oriented design and answer Sub-Questions 1 through 3.

There is a library which uses the lending form shown in Figure 1 below for the circulation of books and CDs. The circulation management system at this library was developed using an object-oriented approach. First, the books and CDs for lending were identified, and the classes "Book" and "CD," shown in Figure 2, were defined. Figure 2 shows that each class has attributes such as a title, etc., and the operations which reference them. The definitions of classes and the relationship between classes are shown in Figure 3.

Title _____

Issue date	Return date	User

Fig. 1 Lending Form

Book
title
author
keyword
Reference title
Reference author
Reference keyword

CD
title
composer
producer
performers
Reference title
Reference composer
Reference producer
Reference performers

Fig. 2 Definition of "Book" and "CD" Classes

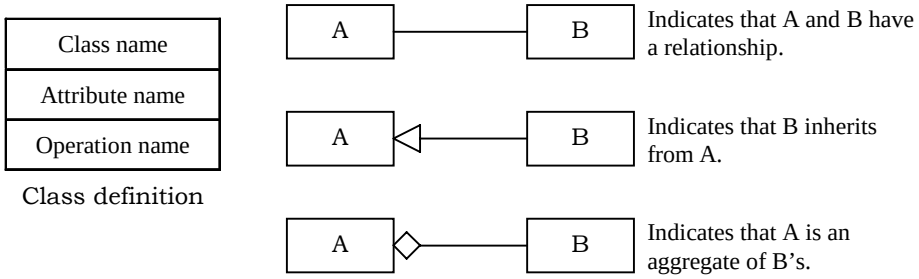


Fig. 3 Definition of a Class and Relationships between Classes

Sub-Question 1

A decision was made to make a superclass called "Items_on_loan" which is common to both "Book" and "CD." Complete Figure 4 below, which gives the structure between the classes. Assume that all the things which had better be defined in the superclass are defined in the superclass and not defined in the subclasses. Be sure to use the attribute names and operation names used in Figure 2.

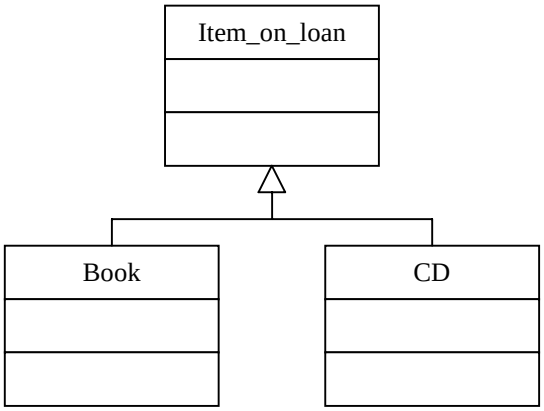
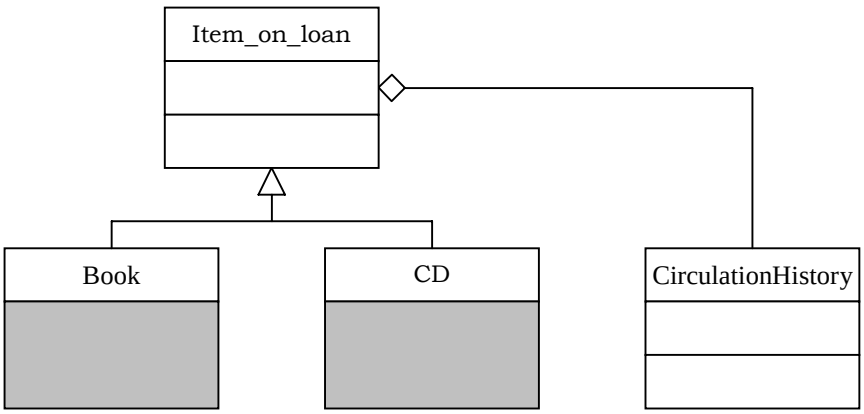


Fig. 4 Class Structure (1)

Sub-Question 2

In order to manage the circulation history, it was decided to make a "CirculationHistory" class containing issue_date, return_date, and user included in the lending form shown in Figure 1. A decision was also made to delegate all the circulation processes related to the "Item_on_loan" class from that class to the "CirculationHistory" class. In Figure 5, fill in the appropriate attribute names and operation names in the boxes of the "Item_on_loan" class and "CirculationHistory" class. Be sure to write only those attribute names and operation names which are required. In addition, write nothing in the boxes of the "Book" and "CD" classes.



Note: Do not write anything in the shaded areas.

Fig. 5 Class Structures (2)

Sub-Question 3

Enter the correct words or phrases in boxes [a] through [g] in the following description of object-oriented programming.

In object-oriented programming, data is handled by [b] using the concept of [a]. This means that, even if changes are made to [c] structures within the object, it is not necessary to change the program on the calling side. This in turn means that improved maintainability can be expected.

If the class structure shown in Figure 4 is used, it is also possible to redefine in subclasses operations defined in their superclass. This is called [d]. Redefining operations represents a different implementation using the same names. The fact that even if the implementation is different, the calling program does not need to be aware of these differences is called [e].

The objects instantiated by the program are secured in the [f] area. As it is difficult for the program to manage the objects which are no longer used, some language processors dispose of unnecessary objects inside this area by means of [g] for the reuse of this area.

Q3. Read the following description of encryption and authentication and answer Sub-Questions 1 through 5.

Company X is planning to exchange e-mails and data (hereinafter referred to as "messages") with its clients over the Internet. From a security standpoint, merely encrypting the text of a message is not enough to ensure safety when exchanging messages over the Internet. To ensure safety, it is necessary also to authenticate the party with whom the message is being exchanged. For the purpose of authenticating the other party, rather than a secret key cryptosystem that uses a common key for both encryption and decryption, it was decided to use the encryption algorithm of a public key cryptosystem. In this system, encryption and decryption are performed using a secret key and public key possessed by each individual. Figure 1 below shows the procedures when Receiver B uses a public key to authenticate Sender A.

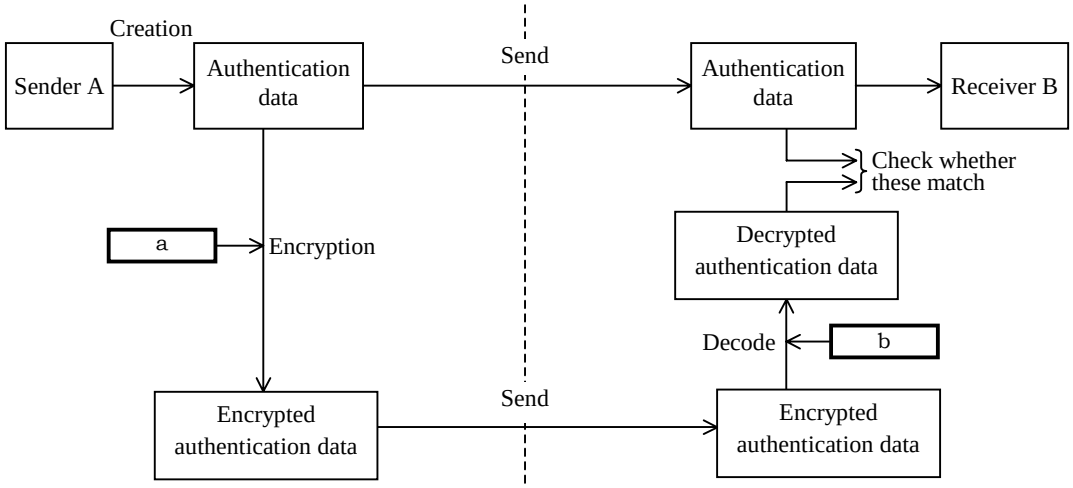


Fig. 1 Procedures for Authentication Using the Public Key System

Figure 2 shows the procedures used to send and receive messages which can be authenticated using the procedure shown in Figure 1. Note that the same words or phrases, respectively, go into [a] and [b] in Figure 2 as in [a] and [b] in Figure 1.

The procedures for sending and receiving messages shown in Figure 2 are described below.

(1) Sender A sends to Receiver B message text encrypted with a common key shared by A and B using a secret key system.

(2) In order to authenticate the other party, A generates a message digest using a hash function shared by A and B, encrypts the generated message digest using a public key system, and sends it to B. Here, assume that A and B can obtain each other's public key by a safe means.

(3) B decrypts the received message digest using the public key system. Then, message text is decrypted using the common key, and the message digest is created from the decrypted text using the hash function.

(4) B authenticates the other party by checking whether or not the decrypted message digest received from A matches the message digest that B generates from the message text.

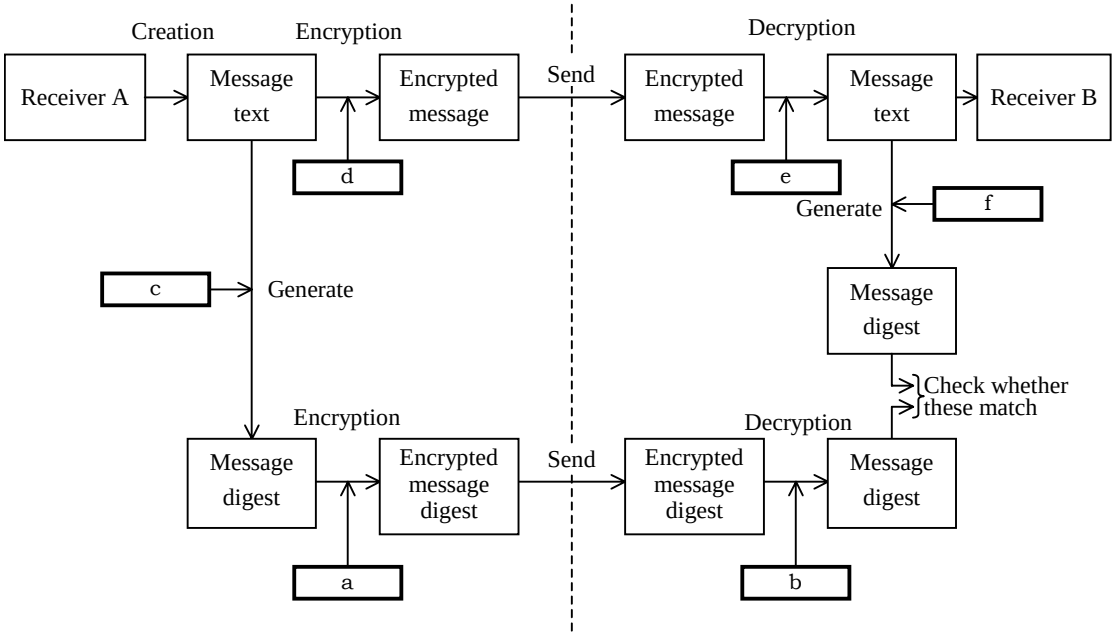


Fig. 2 Procedures for Sending and Receiving Messages Which Could be Authenticated

Sub-Question 1

In Figure 1, what are the correct words or phrases to place in boxes [a] and [b] ? Answer in a few words.

Sub-Question 2

In Figure 2, what are the correct words or phrases to place in boxes [c] through [f] ? Answer in a few words.

Sub-Question 3

What are the correct words or phrases to place in box [g] in the following description? Answer in less than several words.

When sending mail following the procedures given in Figure 2, a different encryption system is used for encrypting the message itself and for the message digest. This is because [g] is longer for a public key cryptosystem than a secret key cryptosystem.

Sub-Question 4

What is to be prevented by using a message digest for authentication data from a security standpoint? Answer in a few words.

Sub-Question 5

Select from the following answer group the characteristic that should be possessed by the hash function which generates the message digest.

Answer group:

a) It can create the same output data from different input data.
b) The time required to obtain output data from input data is short.
c) The time required to obtain output data from input data is long.
d) The length of data which is outputted is longer than that of data which is inputted.
e) It is not possible to restore input data from output data.

Q4. Read the description below about system performance and answer Sub-Questions 1 through 3.

Company Y in the securities business has decided to construct a contract and inquiry system. This system consists of one server and 100 client PCs connected to a LAN, and the server is connected to two securities exchanges by a private line. (See figure below.)

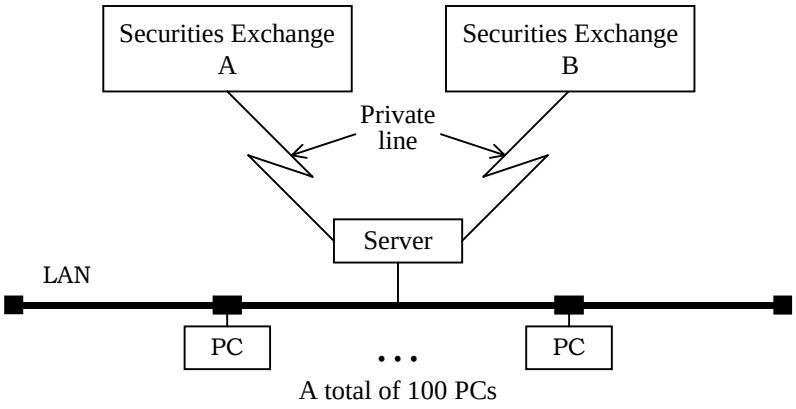


Fig. Contract and Inquiry System for Company Y

The server CPU is an expandable symmetrical multiprocessor, while the OS uses virtual storage. Regardless of the number of CPUs, the OS requires 60 MB of memory.

This system simultaneously performs four types of processes: receiving data from the securities exchanges, and performing the processes of Applications 1 through 3, described later. The server executes the processing for receiving data by itself, while the server and client PCs work together to execute the processes of Applications 1 through 3.

The PCs use a common client software, and one and only one of Applications 1 through 3 is selected and executed by a given PC. When execution is completed, another application is selected and executed. It is not possible to simultaneously execute more than one application on a single PC.

The following is a brief description of each type of process handled by this system.

(1) Receiving data from the securities exchanges

Securities Exchanges A and B both send to the server the latest transaction status every 10 seconds over the private lines. The server receives, edits and analyzes this data. The task to do this processing is the receiving task. One receiving task uses 5 MB of memory, and when processed by a single CPU, uses 200 ms of CPU processing time.

(2) Application 1

A PC which has selected Application 1 displays in real time the latest transaction status for Securities Exchange A. The task which performs Application 1 processing on the server (Application 1 Task) sends the latest transaction information for Securities Exchange A to the PC. This data has a fixed length of 64 bytes per record and consists of a total of 2,000 records. The server sends all records in less than 10 seconds.

Application 1 Task uses 6 MB of memory, and when processed by a single CPU, uses 300 ms of CPU processing time.

(3) Application 2

A PC which has selected Application 2 displays in real time the latest transaction status for Securities Exchange B. The task which performs Application 2 processing on the server (Application 2 Task) sends the latest transaction information for Securities Exchange B to the PC. Except for the fact that the total number of records is 1,000, the record length for data sent to the PC and the time required to send records are the same as those in Application 1.

Application 2 Task uses 6 MB of memory, and when processed by a single CPU, uses 200 ms of CPU processing time.

(4) Application 3

A PC which has selected Application 3 processes contract data. The task which performs Application 3 processing on the server (Application 3 Task) processes clients' contract data which is sent from a PC. The data sent between the PC and the server consists of 3,600 bytes per contract. The operating time per contract is 30 seconds.

Application 3 Task uses 10 MB of memory. Since the operating time of 30 seconds is long, CPU processing time can be ignored.

The server generates application tasks in accordance with the application selected by a PC. When processing on the PC is completed, the associated task also completes and terminates. In the case of the receiving task, one task is generated for each securities exchange when the server is started.

At the peak time, the number of PCs each executing an application is 50 in the case of Application 1, 30 in the case of Application 2, and 20 in the case of Application 3.

Since this system uses virtual storage, there is no need for all the memory used by the OS and tasks to be allocated to the main memory. However, the processing speed is assumed to drop if the system does not allocate 80% of the memory required by the OS, 50% of the memory required by application tasks, and 80% of the memory required by the receiving task to the main memory.

Sub-Question 1

What is the minimum amount of main memory that must be installed so that the processing speed of application processing may not drop at the peak time?

Sub-Question 2

At the peak time, one CPU alone will not complete processing within the specified length of time. We therefore want to add enough CPUs so that all processing may be completed within the specified length of time. What is the minimum number of CPUs that must be installed on the server? Note that processing speed is in proportion to the number of CPUs.

Sub-Question 3

Answer the following questions about the amount of network data and communication protocols.

(1) If TCP/IP is used as the communication protocol between the server and the PCs, what is the minimum amount of data flowing on the LAN at the peak time in terms of Mbps? Round the result to one decimal place.
It may be assumed here that only data flows on the LAN; control codes, etc., may be ignored.

(2) In case of Applications 1 and 2, nearly identical data is sent on the network to the PCs which have selected these. Even though TCP is currently anticipated as the communication protocol, it has been decided to study UDP also. From the following answer group, select the items which most appropriately describe the features of UDP as compared to TCP.

Answer group:

a) Data is fixed length.
b) It is possible to use broadcast packets to send the same information to more than one location simultanously.
c) An acknowledge packet is always returned when data is received.
d) Reliability is high.

Q5. Read the description below about a program used to check the pairing of parentheses, and answer Sub-Questions 1 through 3.

Consider an algorithm for a program used to check the pairing of parentheses in a text file. If a corresponding parenthesis is not found, the position of the unpaired parenthesis is displayed. The matching parenthesis for a given parenthesis does not have to occur on the same line.

There are two cases where a corresponding parenthesis cannot be found.

Case 1: A right parenthesis was found, but the corresponding left parenthesis does not exist. A message indicating that a left parenthesis could not be found, and the location of the right parenthesis in question, are both displayed. If unprocessed text remains, the check for pairing of parentheses for the rest of the file continues.

Case 2: After all parentheses have been checked, there remains one or more left parentheses for which no corresponding right parenthesis exists. Since there may be more than one unpaired left parenthesis, a message indicating that one or more right parentheses could not be found is displayed first, and then the locations of all the left parentheses in question are displayed below that message.

For a file with contents such as shown in Figure 1 below, error messages are displayed as shown in Figure 2.

```
(1+2)
abc)
((def)gx))
(((h)
ij)(k
(lml)
```

Fig. 1 Contents of Text File

```
Left parenthesis is missing.
Line 2, character 4

Left parenthesis is missing.
Line 3, character 10

Right parentheses is missing.
Line 5, character 4
Line 4, character 1
```

Fig. 2 Display of Error Messages

In this program, an integer stack is used to store the location of parentheses. Stack operations include `push` and `pop` as well as `empty`, which checks if the stack is empty or not, and `peek`, which returns the value on the top of the stack.

In addition to stack operations, it is also possible to use the functions `nextch` and `kind`. `nextch` is a function used to read the next character from the text file. At the same time that the input character is read and its value is returned, this function sets the variables `line` and `pos` to the line number and character position, respectively, from which the character in question was read. However, at the end of the file, the variable `EOF` is set to

"true" and a space is returned. The initial value of the variable `EOF` is "false." In addition, `kind` is a function which takes a character as its only argument and determines whether or not that character is a parenthesis. The values of this function are as follows:

`kind (c)` = 0 `c` is not a parenthesis.
1 `c` is a left parenthesis.
2 `c` is a right parenthesis.

The algorithm written based on this observation is given below:

[Algorithm 1]

```
Initialize the stack to empty
ch ← nextch()
while(not EOF)
      k ← kind(ch)
      if( a )
            b
            c
      elseif( d )
            if( e )
                  dummy ← pop()
                  dummy ← pop()
            else
                  Displays a message (A)
                  Displays the line number(line) and
                  character position (pos)
            endif
      endif
      ch ← nextch()
endwhile
if(not empty())
      Displays a message (B)
      while(not empty())
            L ← pop()
            P ← pop()
            Displays the line number(L) and character
            position (P)
      endwhile
endif
```

When there is more than one kind of parenthesis being used, it is necessary to check the pairing for each of the different kinds. Complete nesting structures like (···[···](···)) are required when multiple pairs of parentheses are used, and structures such as (···[···)] are not allowed.

At this point, Algorithm 1 has been revised to handle three kinds of parentheses: ( ), { }, and [ ]. In this case, it is necessary to include three types of data in the stack: the type of parenthesis, the line number, and the character position. First, codes are used as given in the table below to identify the type of parenthesis. `kind` has been revised to return to the corresponding parenthesis code when a parenthesis is encountered and to "0" for all other characters.

Table Parenthesis Codes

(	)	{	}	[	]
1	2	3	4	5	6

If a right parenthesis appears when a different kind of left parenthesis is on the top of the stack, a message indicating that the corresponding left parenthesis could not be found, and the location of the right parenthesis in question, is displayed. The left parenthesis on top of the stack is left as is and processing continues.

The algorithm changes as follows according to the description just given.

[Algorithm 2]

```
Initialize the stack to empty
ch ← nextch()
while(not EOF)
     k ← kind(ch)
     if(k > 0)
          if( [ f ] )
               Push data onto the stack
          elseif( [ g ] and [ h ] )
               dummy ← pop()
               dummy ← pop()
               dummy ← pop()
          else
                Displays a message (A)
                Displays the line number(line) and character position (pos)
          endif
     endif
     ch ← nextch()
endwhile
if(not empty())
     Displays a message (B)
     while(not empty())
          dummy ← pop()
          L ← pop()
          P ← pop()
          Displays the line number(L) and character position (P)
     endwhile
endif
```

Sub-Question 1

Put the character strings corresponding to the correct answer into boxes [a] through [e] in Algorithm 1.

Sub-Question 2

What messages are displayed in locations (A) and (B) in Algorithms 1 and 2?

Sub-Question 3

Select from the following answer group the processing which should be performed in boxes [f] through [h] in Algorithm 2. In addition, give the specific expressions required for this processing. If necessary, be sure to use "%" for operations in case a remainder needs to be found. Since the evaluation of conditional expressions proceeds left to right, assume that remaining evaluations are not performed if the value meets a condition before reaching the end of the expression.

Answer group:

(a) Is k other than a parenthesis code?
(b) Is k a left parenthesis code?
(c) Is k a right parenthesis code?
(d) Is the data at the top of the stack a left parenthesis corresponding to k?
(e) Is the data at the top of the stack a right parenthesis corresponding to k?
(f) Is the stack empty?
(g) Is the stack not empty?

Q6. Read the following description of a system for managing the progress status, and answer Sub-Questions 1 through 4.

Company A is an SI vendor. In order to manage the progress status of an application development project, an actual work summary was created based on the daily work reports submitted by the developer. This time, the progress status management system was developed for the purpose of reducing the project manager's workload and providing the actual data for other projects to follow.

At company A, a daily work report input screen based on the format of the existing forms filled out by hand (Fig.1) and the actual work summary table (Fig.2) were designed. The items entered by the worker and the items outputted to the actual work summary table were determined as described below.

[Items entered by the worker]

- Before going home, the worker enters works having been actually done that day into the daily work report input screen (Fig.1). At this time, the number "1" is entered into the "completion flag" when the work in question has been completed.
- Items inputted by the worker are classified into direct works such as writing application design specifications, coding source programs, and testing and indirect works such as meeting and general affairs. These are inputted with work codes.
- Work hours are inputted in units of 0.1 hours (= 6 minutes) for each work code.

[Items outputted to the actual work summary table]

- Of the items entered by the workers using this system, only data related to direct works is gathered, and outputted to an actual work summary table (Fig. 2) used for reference by the project manager at weekly progress meetings.
- The actual work summary table contains data arranged by work code and by worker code.
- The "actual hours" included in the actual work summary table are obtained by adding by work code and by worker code all the hours worked as entered from the daily work report input screen.
- The "estimated hours" in the actual work summary table are obtained by work code and by worker code at the time when a project is planned.

Next, in order to design the database used by this system, an E-R diagram (Fig. 3) was created for managing the work status.

Work report input screen

Worker code: 123456 Worker's name: Jim Dinkey Work date: 2001/04/15

Seq. No.	Work code	Work name	Work hours	Completion flag	Remarks
1	3359	PGM0020 internal design specifications	2.6	1	
2	3360	PGM 0020 source coding	4.5		
3	3361	PGM0020 test specification	2.5		
4	3461	PGM0030 test specifications	1.0		
5	9010	Progress meeting	1.0		
6					
Daily total			11.6		

Overtime hours (3.6)

Fig. 1 Work Report Input Screen

Actual work summary table

Seq. No.	Work code	Work name	Estimated hours	Worker code	Worker's name	Actual hours
1	3359	PGM0020 internal design specifications	16.0	123456	Jim Dinkey	18.3
2	3360	PGM 0020 source coding	16.0	123456	Jim Dinkey	11.5
3	3361	PGM0020 test specifications	8.0	123456	Jim Dinkey	3.5
4	3459	PGM0030 internal design specifications	18.0	789012	Jack Smith	11.2
5	3461	PGM0030 test specifications	4.0	123456	Jim Dinkey	1.0
6	3461	PGM0030 test specifications	4.0	789012	Jack Smith	1.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Fig. 2 Actual Work Summary Table

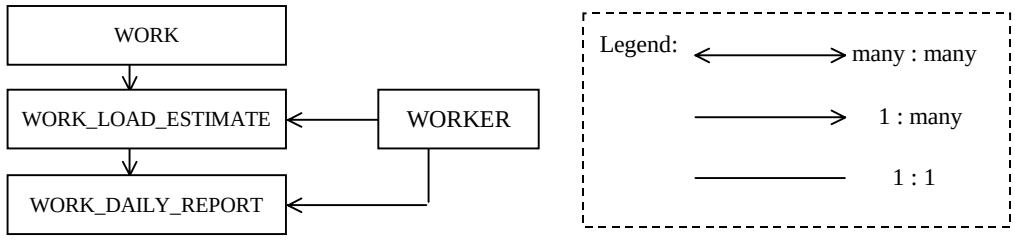


Fig. 3 E-R Diagram Related to Work Status Management

Sub-Question 1

Because data related to direct works and data related to indirect works cannot be shown separately in the case of the E-R diagram shown in Figure 3, a “work category table” was added as shown in Figure 4 below.

The work category table includes the following items and values as attributes:

When work category code = 1, the work category name = direct work
When work category code = 0, the work category name = indirect work

(1) Observing the rules shown in the legend of Fig. 3, complete the E-R diagram by adding the work categories to Fig. 3.

(2) Place the correct character strings in boxes [a] through [c] in Fig. 4. Underline the ID key (primary key) item(s).

WORK_TBL	(WORK_CODE, [a], WORK_NAME)
EFFORT_ESTIMATE_TBL	(WORKER_CODE, WORK_CODE, ESTIMATED_TIME)
WORK_DAILY_TBL	(WORKER_CODE, WORK_CODE, WORK_DATE, WORK_HOURS, COMPLETE_FLAG, REMARKS)
WORKER_TBL	(WORKER_CODE, WORKER_NAME)
WORK_CATEGORY_TBL	([b] , [c])

Fig. 4 Tables and Attributes Related to Work Management

Sub-Question 2

Create a user view for outputting the actual work summary table by gathering the work hours by work code and by worker code. Place the correct character strings in the boxes [d] through [h] in the following SQL code.

```
[ d ] WORK_ACTUAL_TBL
  (WORK_CODE, WORK_NAME, ESTIMATED_TIME, WORKER_CODE, WORKER_NAME,
  ACTUAL_TIME)
[ e ]   D.WORK_CODE, D.WORK_NAME, C.ESTIMATED_TIME,
        A.WORKER_CODE, A.WORKER_NAME, [ f ]
FROM WORKER_TBL A, WORK_DAILY_TBL B, EFFORT_ESTIMATE_TBL C, WORK_TBL D
[ g ]
GROUP BY [ h ]
```

Sub-Question 3

Using the user view defined in Sub-Question 2, re-arrange the data organized by work code and by worker code in the actual work summary table so that it can be outputted in the ascending order of work codes and in the descending order of worker codes. Place the correct character strings in the boxes [i] through [l] in the SQL code below.

Assume that the default interpretation for omitted values in the SQL syntax is not used.

```
[ i ] *
[ j ]
[ k ] WORK_CODE [ l ], WORKER_CODE [ l ]
```

Sub-Question 4

Using the user view defined in Sub-Question 2, an SQL statement was created to make a list of works for each of which the actual time required exceeded the estimated time. At this time, an error resulted when a `WHERE` statement was coded as shown below. Place the correct character strings in boxes [m] through [o] in the following description of the reason for the error. Place the same character string in box [f] as used for box [f] in Sub-Question 2.

```
WHERE  ACTUAL_TIME  > ESTIMATED_TIME
```

Reason for the error:

A query process on [m] is converted to a query process on [n] using the definition [m].

Therefore,

```
WHERE  ACTUAL_TIME  > ESTIMATED_TIME
```

is replaced with

```
WHERE [ f ] > ESTIMATED_TIME
```

This is an incorrect `WHERE` statement because it is not possible to specify [o] as a condition inside the `WHERE` statement.

2001 SPRING

Software Design and Development Engineer Examination (Afternoon, Part 2)

Questions must be answered in accordance with the following:

Question Nos.	Q1
Question Selection	All Sub-Questions are compulsory
Examination Time	15:30-16:30 (60 minutes)

Instructions:

1. Use an HB pencil. If you need to change an answer, erase your previous answer completely and neatly. Wipe away any eraser debris.
2. Mark your examinee information and test answers in accordance with the instructions below. Your test will not be graded if you do not mark properly. Do not mark on white on the answer sheet outside of the prescribed places.
   (1)Examinee Number
   Write your examinee number in the space provided, and mark the appropriate space below each digit.
   (2)Date of Birth
   Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.
   (3)Write each answer in the space specified for that question.
   (4)Write your answers clearly and neatly. Answers that are difficult to read will receive a lower score.
3. After the test, you may take this question booklet home with you.

Do not open the exam booklet until instructed to do so.
Inquiries about the exam questions will not be answered.

Q1. Read the following description of a hotel reservation system and answer Sub-Questions 1 through 5.

Travel Company A has decided to build a hotel reservation system. As shown in Figure 1, this system consists of a hotel reservation server and client PCs installed at customers' offices and hotels. They are connected over a network. The main task of this system is to accept reservations from the clients installed at the sites of customers who are registered as members of the system. Reservations are made to the hotel room information service, (hereinafter referred to as room inventory), which has been registered by client PCs in the hotels to the hotel reservation server.

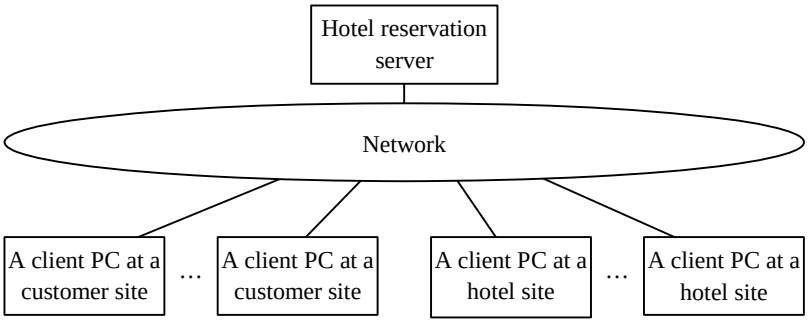


Fig. 1 Configuration of the Hotel Reservation System

Tasks handled by this system are shown in the table below.

Table　Tasks Handled by the Hotel Reservation System

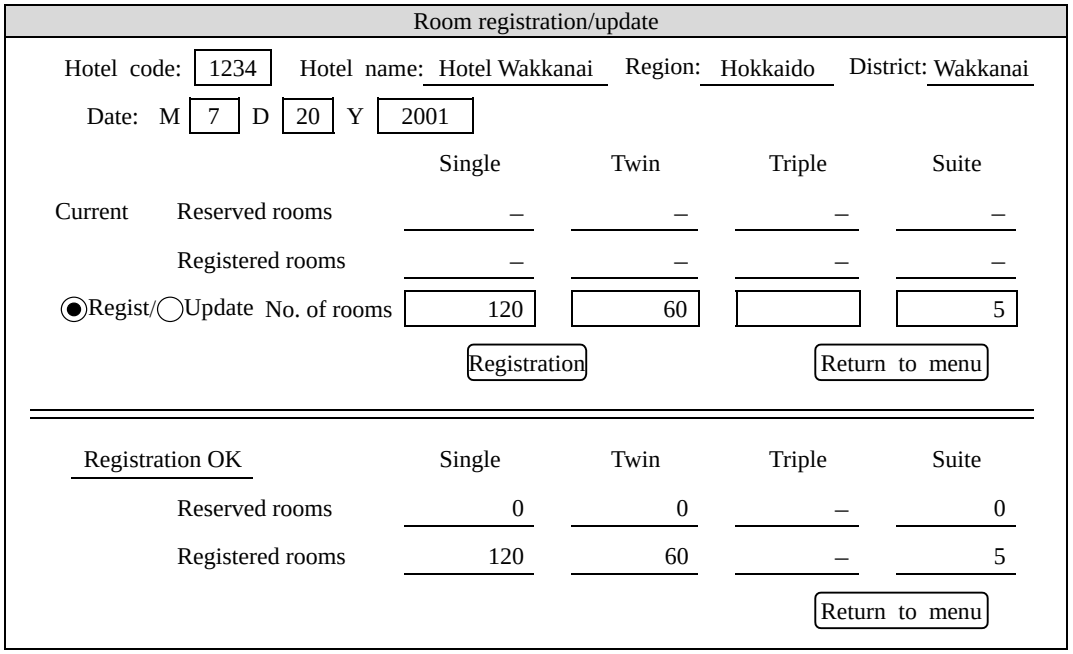
Clients	Tasks	Description of function
Hotel site	Register/update room inventory	Room inventory for each room type is registered by specifying the hotel and available dates.　It is also possible to update the room inventory.
	Output reservation list	The listing of the rooms reserved, customers, and room rates, is displayed for the specified hotel and check-in date.
	Register room rates	For the specified hotel, the room rates for each season and room type are set as specified.
Customer site	Search for/reserve hotel rooms	The region, district, check-in date, room type and number of rooms are specified and a search is made for hotels which satisfy specified conditions.　It is also possible to select any one of the hotels found and reserve a room or rooms.
	Query/cancel reservation details	Displays the details of a reservation made by a customer. Reservations may also be canceled.
	Register/update/delete member information	Customers may register as members of this service by entering the required customer information.　It is also possible to update and delete this customer information.

The prerequisites assumed for this system are given below.

- The system manages the number of rooms in the room inventory.　In other words, hotel reservations are only allowed when the number of available rooms is greater than or equal to the number of rooms being requested.　The number of available rooms is reduced or increased when a reservation or cancellation is made.
- There are four room types: single, twin, triple, and suite.
- There are three types of season: off-season, mid-season, and in-season. This is determined based on the check-in date.
- Room rates are set by hotel, by room type and by season.　The room rate quoted is for a single night in a room.
- The total price is calculated by multiplying the number of rooms by the room rate. Discounts are not considered.
- Only the customers who are registered as members may make hotel reservations.　It is impossible to delete the membership of a customer who has a reservation.

Note that issues such as authentication of client operators, the duration of each task, changes in room rates, issues of coupons and settlements are outside the scope of this problem.

Examples of the display screens and output reports used in this business process are shown in Figures 2 through 4. The legend used in Fig. 2 also applies to Fig. 3.



Legend:

: Input area

: Pull-down menu (selection from choices)

: Display area for output depending on input

◉, ◯ : Radio button (selection of one item)

: Execution button

Clicking on the button for executing a registration, search or reservation on the part of the screen above the double line ═══, causes the screen below the double line ═══ to be displayed.

Fig. 2 Room Registration / Update Screen (sample)

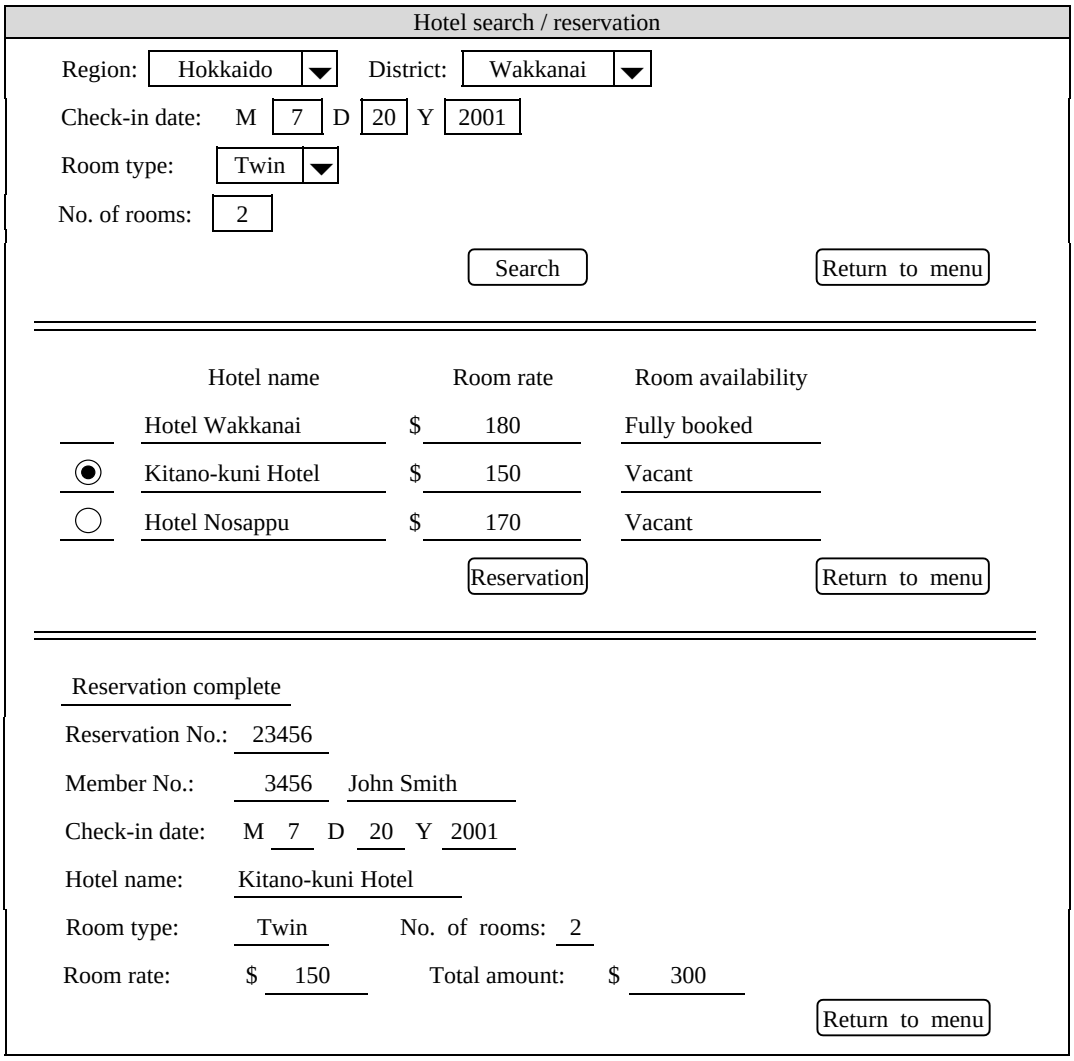


Fig. 3 Hotel Search / Reservation Screen (sample)

Reservation list — As of 10 July, 2001

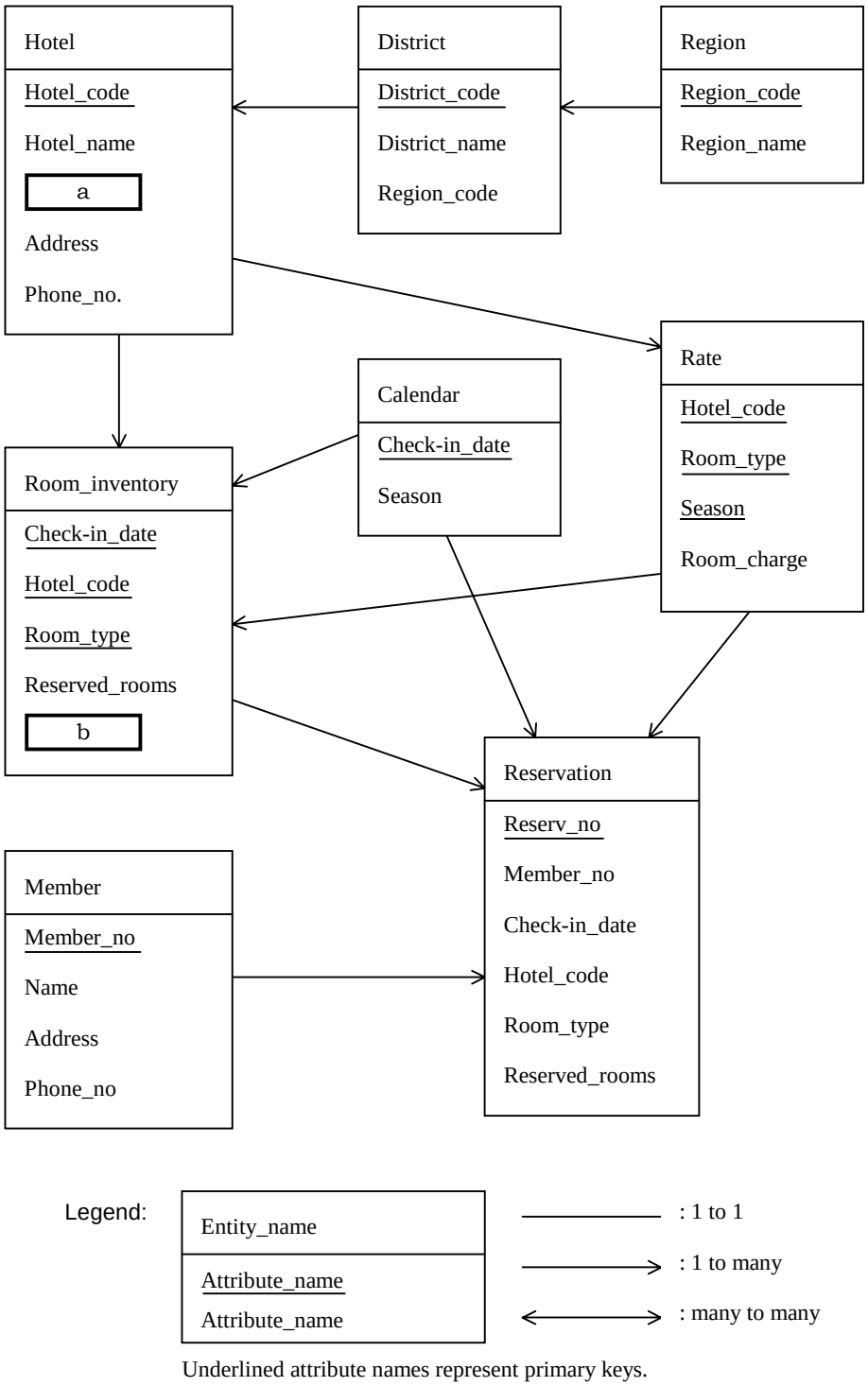
Hotel code: 1234 Hotel name: Hotel Wakkanai Region: Hokkaido District: Wakkanai

Check-in date: 15 July, 2001

Reservation No.	Room type	No. of rooms	Member No.	Name	Total amount
45678	Single	1	7890	Joe Smith	$120
56789	Single	2	8901	Mary Diamond	$240
67890	Twin	1	9012	John Lee	$180
⋮	⋮	⋮	⋮	⋮	⋮

Fig. 4 Output Reservation List (sample)

Figure 5 shows the E-R diagram determined for this system.



Underlined attribute names represent primary keys.

Fig. 5 E-R Diagram

Here, the hotel reservation server stores data into the relational databases which are defined by the tables reflecting the E-R diagram. In order to match the E-R diagram, entity names are used for the table names and attribute names for the column names, and appropriate data types must be used for the tables. Data is processed using embedded SQL within the host programming language used to code the main program to handle various task requests from clients.

Sub-Question 1

Place the correct attribute names in boxes [a] and [b] in the E-R diagram shown in Fig. 5.

Sub-Question 2

In the table definitions using SQL, the deletion of memberships for customers who already have reservations is prevented by setting the following integrity constraint. Place the correct words in the following boxes [c] through [f].

When it comes to [d] column in [c] table, [f] is set for [e] table.

Sub-Question 3

The following SQL code is the code for creating the reservation listing shown in Figure 4 for the hotel code and check-in date that have been specified. Place the correct word or expression in boxes [g] through [i] in the description below.

Note that ":XXX" is the host variable that stores the data on "XXX" specified by the client.

```
SELECT  RESERVATION.RESERV_NO, RESERVATION.ROOM_TYPE,
        RESERVATION.RESERVED_ROOMS, RESERVATION.MEMBER_NO,
       MEMBER.NAME, [ g ]
  FROM RESERVATION, MEMBER, RATE, CALENDAR
  WHERE RESERVATION.HOTEL_CODE = :HOTELCODE
```

```
AND RESERVATION.CHECKIN_DATE = :CHECKINDATE
AND [          h          ]
AND RATE.HOTEL_CODE = :HOTELCODE
AND RATE.ROOM_TYPE = RESERVATION.ROOM_TYPE
AND CALENDAR.CHECKIN_DATE = :CHECKINDATE
AND [          i          ]
```

Sub-Question 4

In order to handle multiple nights, the travel company decided to add an input area for “number of nights” to the hotel search/reservation screen so that all reservation operations with a common combination of the hotel, room type, and number of rooms which are related to a multiple-night stay could all be handled with a single reservation number at once. To do this, the “Reservation” entity in the E-R diagram shown in Figure 5 was split into two parts: “Reservation,” for single reservations, and “Reservation_details” for each night of stay, and the attributes were allocated to the two. Enter the correct attribute names associated with each entity according to the legend and complete the part for the E-R diagram.

Note that the room rates for the season in question are applied to each night of stay and that there should be no unnecessary, redundant data (the second normal form).

Sub-Question 5

Figure 6 shows the flow of processing for updating data when the Reservation button is pressed within the hotel reservation program after the specification changes in Sub-Question 4 have been made. With this program, the cursor coming from the room inventory table is declared as shown in Figure 7 and processing for multiple nights is handled using cursor operations.

Although multiple instances of data are updated when using this program, a transaction management function is used for the relational database in order to preserve integrity when updates are requested simultaneously by multiple clients, or when a system error occurs in the middle of processing. Using this function, transactions are started automatically when the program starts, and

are separated when an SQL COMMIT or ROLLBACK statement is executed. Data processed during that time is either confirmed or canceled.

Because data is not locked between the point that a hotel search is made and a hotel reservation is made, it is necessary to confirm that a room is still available at the time of reservation even if the hotel had empty rooms when it was found. CLOSE processing for the cursor is automatically executed when a COMMIT or ROLLBACK statement is executed.

In the case of multiple-night reservations, reservations are made only if rooms are available for all the nights being requested. Data related to the room inventory is not affected in any other cases.

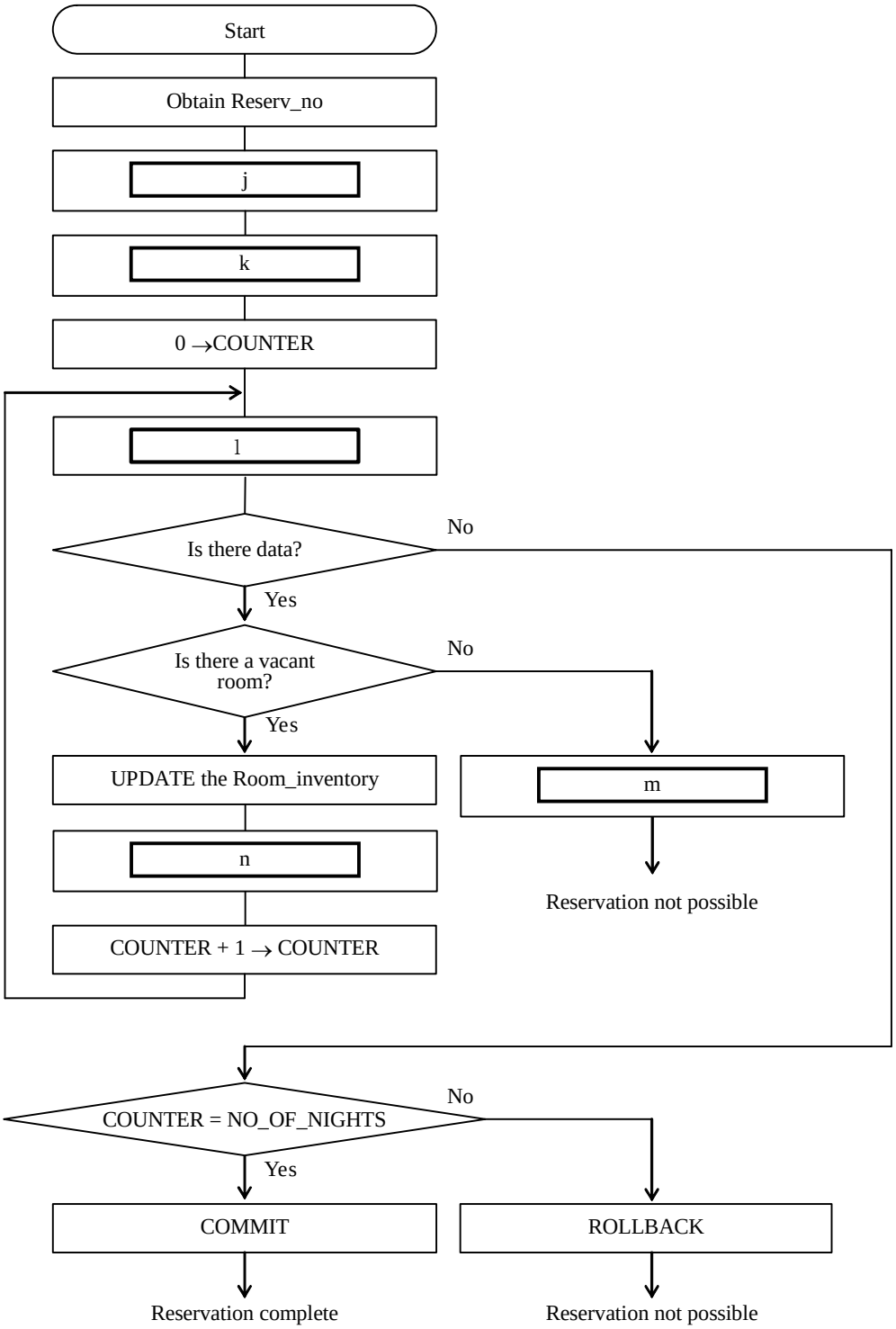


Fig. 6 Flow of Data Update Process When Reservation Button is

```
DECLARE ROOM_STOCK_CSR CURSOR
 FOR SELECT *
   [ o ]
  WHERE HOTEL_CODE = :HOTELCODE
    AND ROOM_TYPE = :ROOMTYPE
    AND [ p ]
  FOR UPDATE;
```

Fig. 7 Cursor Declaration

(1) Place the symbol for the correct word (appearing in the following answer group) in boxes [j] through [n] within the flowchart shown in Fig. 6 related to updating data when the Reservation button is pressed.

Answer group:

a) COMMIT
b) ROLLBACK
c) FETCH with the cursor
d) OPEN the cursor
e) INSERT into the Room_inventory
f) INSERT into the Reservation
g) UPDATE the Reservation
h) INSERT into the Reservation_details
i) UPDATE the Reservation_details

(2) Place the correct word or expression in boxes [o] through [p] in the cursor declaration shown in Fig. 7. Note, however, that the host variables ":CHECKINDATE_F" and ": CHECKINDATE _T" can be used. They are of the same DATE type as "CHECK_IN_DATE" in the database. They represent, respectively, the first night and last night of the continued stay (the check-in date + the number of nights – one day) as specified by the customer.