ITEE

The Project for Skill's Development of ICT Engineers Targeting Japanese Market

ITEE
Information Technology Engineers Examination

- Questions and Answers from 8 Latest Examinations (Afternoon) of ITEE -

October 2014 – March 2018

Source: http://itpec.org/

TABLE OF CONTENTS

March 2018

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the Ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.
   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is this Fundamental IT Engineer Examination conducted?

   Answer group
   a) February b) March c) April d) May

   Since the correct answer is " b) March ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc., of procedures, variables, etc.
/* text */		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 ――― process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then process 1 is executed. If it is `false`, then process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the iteration process, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	+, −, not	High
Multiplication, division	×, ÷, %	↑
Addition, subtraction	+, −	
Relational operation	>, <, ≥, ≤, =, ≠	
Logical product	and	↓
Logical sum	or	Low

Note: With division of integers, an integer quotient is returned as a result.
The “%” operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description of a technique used for storing passwords in a password file, and then answer Subquestions 1 through 3.

The front line of defense against intruders is the password system. Many multi-user systems use a userID and a password to authenticate the userID of an individual logging on to the systems. A userID provides security in the following ways:

(1) A userID determines whether the user is [A] to gain access to a system.

(2) A userID determines the attributes or access rights given to an individual. A few users may have "[B]" status that enables them to read files and perform functions that are especially protected by the operating system.

The modified Data Encryption Standard (DES) is one of the password encryption techniques, where each user can select a password of up to 8 printable characters in length. If the password is less than 8 characters, then the technique extends it to 8 characters by padding null characters at the end. The password is then converted into a 56-bit value (using 7-bit ASCII characters) that serves as the key input to the encryption routine. The DES algorithm is modified using a 12-bit "salt" value. Typically, this value is related to the time at which the password is assigned to the user.

The salt serves three purposes:

(1) It prevents duplicate passwords from being visible in the password file. Even if two users choose the same password, these passwords are assigned at different times. Therefore, the hashed passwords of the two users will differ.

(2) It effectively increases the length of the password without requiring the user to remember two additional characters. Hence, the number of possible passwords is increased by a factor of [C], increasing the difficulty of guessing a password.

(3) It prevents the use of a hardware implementation of DES.

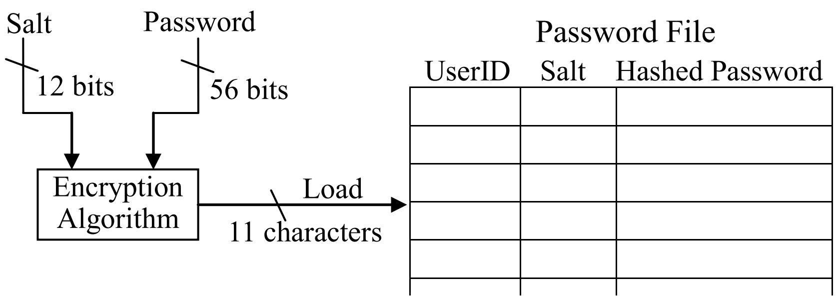


Figure 1 Procedure for loading a new password

Figure 1 shows the procedure for loading a new password. The modified DES algorithm first encrypts the input data consisting of a 64-bit block of zeros using the salt and password as the key. The output of the algorithm then serves as the input for the second encryption. This process is repeated for a total of 25 encryptions. The resulting 64-bit output is then translated into an 11-character sequence. The hashed password is then stored, together with a plaintext copy of the salt, in a password file for a corresponding userID. This method has been shown to be secured against a variety of crypt-analytic attacks.

When a user attempts to log on, the user provides a userID and a password. The system uses the userID to index into the password file, and retrieves the plaintext salt and the encrypted password. The salt and user-supplied password are used as input to the encryption algorithm. If the result matches the stored value, the password is accepted.

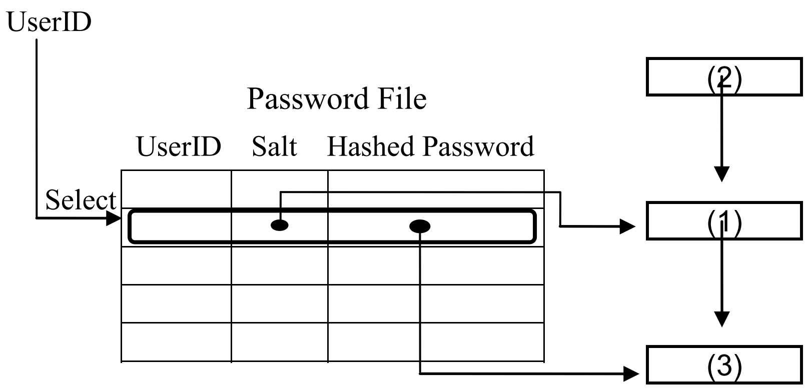


Figure 2 Verifying a password

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\text{A}}$ through $\boxed{\text{C}}$ in the above description.

Answer group for A

a) authorized
b) educated
c) responsible
d) skillful

Answer group for B

a) general user
b) privileged user
c) programmer
d) systems engineer

Answer group for C

a) 4
b) 12
c) 24
d) 12^2
e) 2^{12}

Subquestion 2

From the answer group below, select the correct combination of the terms that correspond to (1), (2) and (3) in Figure 2.

Answer group

	(1)	(2)	(3)
a)	Concatenation	Encryption Algorithm	Decryption Algorithm
b)	Decryption Algorithm	Hashed Password	Comparison
c)	Encryption Algorithm	Decrypted Password	Concatenation
d)	Encryption Algorithm	Plaintext Password	Comparison
e)	Hashed Password	Encryption Algorithm	Concatenation
f)	Plaintext Password	Comparison	Decryption Algorithm

Subquestion 3

A security administrator is planning to create a password file as shown in Figures 1 and 2 under the following conditions:

(1) A userID is at least 4 characters in length, but must be less than 32 characters.
(2) A password file contains password information for 400 users.
(3) A password file does not contain any other additional or formatting information.
(4) Each character in the password file is represented by 1 byte.

From the answer group below, select <u>two</u> correct statements.

Answer group

a) A user can retrieve the old password from the password file.
b) The hashed password may be used as the primary key of the password file.
c) The maximum size of the password file is 17,800 bytes.
d) The minimum size of the password file is 6,800 bytes.
e) The security administrator shall be able to recover any forgotten password.
f) The userID is not a candidate key for the password file.

Q2. Read the following description of the LRU mechanism for cache management, and then answer Subquestions 1 through 3.

Most computers use a cache memory. Cache memory is comparatively expensive and smaller compared to main memories, but provides a faster access time to the CPU. When a CPU refers to a page, the reference page is, at first, searched in the cache memory and the search result is called a "hit" if it is found. If it is not found, the search result is called a "miss" and in such cases, the page is transferred $\boxed{\quad A \quad}$.

The average page reference time T by the CPU is thus given as:

$$T = m \times Tm + Th + E$$

where m is the miss ratio, Tm is the time to perform a main memory access when there is a "miss", Th is the latency time to reference the cache memory when there is a "hit", and E is the time required for various secondary effects such as the queueing delay.

If Tm = 50ns, Th = 2ns, and E = 5ns, then T is $\boxed{\quad B \quad}$ when there is a "hit", and T is $\boxed{\quad C \quad}$ when a "miss" is observed. In this case, if m = $\boxed{\quad D \quad}$, T will be 32ns.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\qquad}$ in the above description.

Answer group for A

a) from cache memory to the CPU without accessing main memory
b) from cache memory to main memory, and then to the CPU
c) from main memory to the CPU without accessing cache memory
d) from main memory to cache memory, and then to the CPU

Answer group for B and C

a) 2 ns	b) 5 ns	c) 7 ns
d) 52 ns	e) 55 ns	f) 57 ns

Answer group for D

a) 0.14	b) 0.40	c) 0.50
d) 0.56	e) 0.64	

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank $\boxed{\quad\text{E}\quad}$ in Figure 1.

As the cache memory is unable to store all of the pages of the main memory, the least recently used (LRU) paging strategy is implemented to discard the least recently used pages from the cache memory to make room for newly requested pages. To implement LRU, it is necessary to maintain a linked list of all pages in the cache memory, with the most recently used page at the front and the least recently used page at the rear. However, the linked list technique is time consuming as the list must be updated at every page reference. The LRU is therefore implemented using a matrix of $n \times n$ bits for preserving n pages in the cache memory.

The steps of the method are as follows:

(1) The method initially sets all the bits of the $n \times n$ matrix to 0.

(2) Whenever the page frame k is referenced, the method first sets all the bits of the row k to 1, and then sets all the bits of column k to 0.

(3) At any instant, the row containing the least number of 1s is the least recently used.

For example, consider a case where a cache memory can hold 4 page frames, and a CPU references the page frames 0, 3, 2, 1, and 2 in this sequence. Initially, all the bits of the $n \times n$ matrix are set to 0s as shown in Figure 1 (1).

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

(1)

0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

(2)

0	1	1	0
0	0	0	0
0	0	0	0
1	1	1	0

(3)

(intentionally left blank)

(4)

$\boxed{\quad\text{E}\quad}$

(5)

0	0	0	0
1	0	0	1
1	1	0	1
1	0	0	0

(6)

Figure 1 Implementation of LRU using an $n \times n$ matrix

When page frame 0 is referenced, all the bits of row 0 become 1s and then all the bits of column 0 become 0s as shown in Figure 1 (2). Subsequently, after the referencing of pages 3, 2, 1 and 2, the resulting matrices are shown in (3), (4), (5) and (6) of Figure 1 respectively. After all of these page frame references, page frame 0 is the LRU page frame as it contains minimum the number of 1s.

Answer group for E

a)

0	1	0	0
0	0	0	0
1	1	0	1
1	1	0	0

b)

0	0	0	0
1	0	1	1
1	0	0	1
1	0	0	0

c)

0	0	0	0
1	0	1	0
1	0	0	0
1	1	1	0

d)

0	0	0	0
1	0	0	1
1	1	0	1
1	0	0	0

Subquestion 3

From the answer group below, select the correct answer to be inserted in the blank [ ] in the following description.

Consider another case where a cache memory can hold 4 page frames, and a CPU references the page frames 2, 1, 3, 0, 2 and 1 in this sequence. Initially, the cache memory is empty, i.e. no page frames are loaded. In this case, the required time to access all the pages by the CPU is [F]. Here, Tm = 40ns, Th = 4ns, and E = 10ns.

Answer group for F

a) 84 ns
b) 164 ns
c) 244 ns
d) 324 ns
e) 404 ns

Q3. Read the following description of the database system of a floral company, and then answer Subquestions 1 through 3.

A floral company provides services delivering floral gifts to customers. To improve sales, the company implements marketing activities such as launching a Web site, releasing press announcements, and advertising programs in the media. At the beginning of a year, the company schedules the marketing plans for each marketing activity to be performed in the coming year and assigns duties to departments. The scheduled marketing activities are recorded in the table TblMktPlan. At the end of each year, the company collects the budgets used for the scheduled marketing plans. Every budget used to implement individual marketing activities is recorded in the table TblBgtUsed. Some activities are scheduled more than once within a year. Therefore, there is a one-to-many relation between TblMktPlan and TblBgtUsed. The table structures of these tables are as follows:

TblMktPlan (<u>PlanID</u>, MktActivity, Department)

TblBgtUsed (<u>BgtID</u>, PlanID, StartDate, EndDate, BgtUsed, Manager)

Note: Underline indicates the primary key

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank [   ] in the following SQL statement.

The company checks both the implemented and unimplemented marketing activities based on a schedule. The following SQL statement "SQL1" outputs a detailed list of implemented and unimplemented marketing activities with the start date, end date, budget used, manager, and department.

```
-- SQL1 --
SELECT  Mkt.MktActivity, Bgt.StartDate, Bgt.EndDate,
        Bgt.BgtUsed, Bgt.Manager, Mkt.Department
FROM TblMktPlan Mkt [    A    ] TblBgtUsed Bgt ON [    B    ]
```

The sample output of SQL1 is as follows:

MktActivity	StartDate	EndDate	BgtUsed ($)	Manager	Department
Define Marketing Programs	01/01/2017	01/15/2017	3,500	Mr. A	Planning
Locate & Secure Retail Space	05/01/2017	05/31/2017	500	Ms. J	Sales
Locate & Secure Retail Space	11/01/2017	11/30/2017	500	Mr. K	Sales
Launch Website			(NULL)		Web Sales
Purchase Inventory & Supplies	03/01/2017	03/15/2017	1,500	Ms. L	Procurement
Purchase Inventory & Supplies	09/01/2017	09/15/2017	1,500	Ms. L	Procurement
…	…	…	…	…	…

Note:

"INNER JOIN" returns all matching rows from both tables that satisfies the condition
"LEFT JOIN" returns all rows from the left table with matching rows in the right table
"RIGHT JOIN" returns all rows from the right table with matching rows in the left table

Answer group for A and B

a) `INNER JOIN`
b) `LEFT JOIN`
c) `RIGHT JOIN`
d) `Mkt.PlanID = Bgt.PlanID`
e) `Mkt.PlanID = Bgt.PlanID GROUP BY Mkt.Department`
f) `Mkt.PlanID = Bgt.PlanID GROUP BY Mkt.PlanID`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following SQL statement.

The floral company defines four types of sales: retail sale, commercial account sale, frequent flower gift program sale, and holidays/events sale. The retail sale is a daily sale. The commercial account sale is for businesses that require weekly floral arrangements. The holidays/events sale is for holidays/events such as birthdays and Christmas day. The frequent flower gift program sale is for people who work in highly-paid professions. These four types of sales and their descriptions are recorded in the table TblSaleType. For the holidays/events sale, more detailed types of days are defined in the table TblDayType.

The table structures of these tables and their sample data are as follows:

TblSaleType (STypeID, SaleType)

STypeID	SaleType
S01	Retail
S02	Commercial Account
S03	Holidays/Events
S04	Frequent Flower Gift Program

TblDayType (DTypeID, DayType)

DTypeID	DayType
S03D01	Birthday
S03D02	Christmas day
S03D03	Mother's day
S03D04	New year
…	…

The detailed floral sale information is recorded in the table TblSales. The table structure of this table is as follows:

TblSales (SaleID, SaleDate, CustName, STypeID, DTypeID, UnitPrice, Qty)

In December, the company sends promotional information for "Christmas day" and "New year" sale to the customers. The following SQL statement "SQL2" displays the names of customers who bought flowers for "Christmas day" or "New year". A customer name is displayed once even if there are multiple concerned records in the table TblSales.

```
-- SQL2 --
SELECT S.CustName
  FROM TblSales S
  WHERE [   C   ]
```

Answer group for C

a)
```
S.DTypeID = (
  SELECT DT.DTypeID FROM TblDayType DT
    WHERE DT.DayType IN ('Christmas day', 'New year'))
```

b)
```
S.DTypeID IN (
  SELECT DT.DTypeID FROM TblDayType DT
    WHERE DT.DayType IN ('Christmas day', 'New year'))
```

c)
```
S.DTypeID IN (
  SELECT DT.DTypeID FROM TblDayType DT
    WHERE DT.DayType IN ('Christmas day', 'New year'))
GROUP BY S.CustName
```

d)
```
S.DTypeID IN (
  SELECT DT.DTypeID FROM TblDayType DT
    WHERE DT.DayType IN ('Christmas day', 'New year'))
GROUP BY S.CustName
```

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following SQL statement.

In order to forecast the coming year's sales, the company also analyzes the buying patterns of customer behavior by sale type. The following SQL statement "SQL3" outputs the sales amount per year by sale type with the average amount and frequency. The frequency indicates the number of times customers make purchases. Here, the table TblSales contains this year's sales data.

```
-- SQL3 --
SELECT ST.SaleType, [    D    ] AS Average,
       COUNT(S.SaleID) AS Frequency, [    E    ] AS PerYear
FROM   TblSales S, TblSaleType ST
WHERE  S.STypeID = ST.STypeID
GROUP BY ST.SaleType
```

The sample output of SQL3 is as follows:

SaleType	Average ($)	Frequency	PerYear ($)
Retail	30	3,000	90,000
Commercial Account	750	150	112,500
Holidays/Events	50	400	20,000
Frequent Flower Gift Program	100	100	10,000

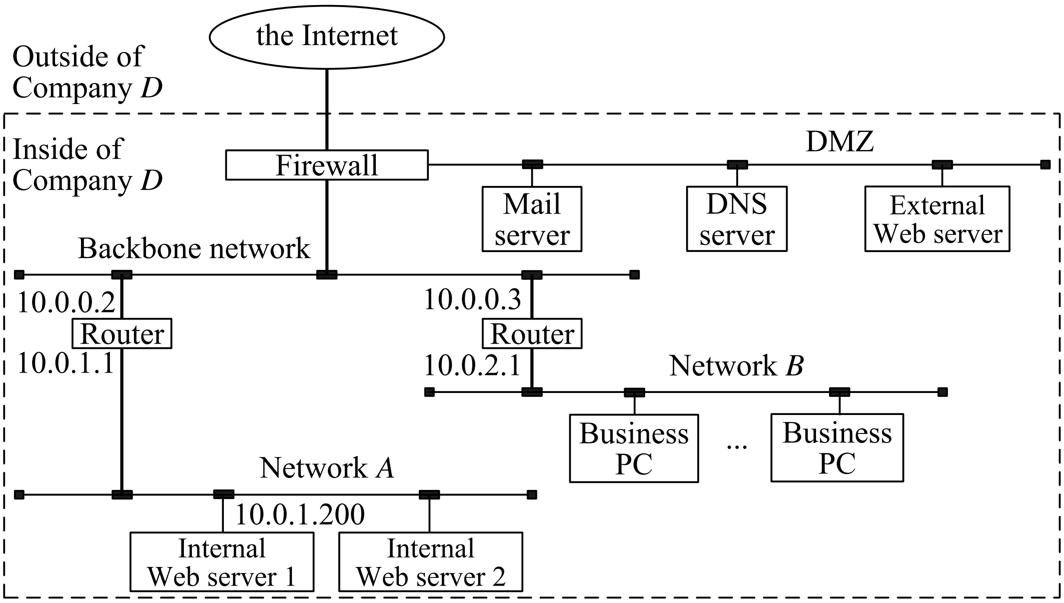
Answer group for D and E

a) `AVG(S.Qty)`
b) `AVG(S.UnitPrice * S.Qty)`
c) `S.UnitPrice * S.Qty`
d) `SUM(S.Qty)`
e) `SUM(S.UnitPrice * S.Qty)`

Q4. Read the following description of construction of a network, and then answer Subquestions 1 and 2.

Figure 1 shows the current network configuration of Company *D*. A mail server, a DNS server, and an external Web server for external access are connected to DMZ. Internal Web servers for internal business use are connected to network *A* and business PCs for employees' normal office work are connected to network *B*.

A firewall blocks all communication from the Internet to the backbone network, as well as all communication from the backbone network to the Internet. Therefore, the business PCs can access the external Web server and the internal Web servers located inside of Company *D*, but cannot access Web servers outside of Company *D*.



Note: The numbers are the IP addresses of routers and the internal Web server 1 in the relevant networks.

Figure 1 Current network configuration of Company *D*

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

The IP addresses of devices that are connected to the relevant networks of Company *D* indicate that the subnet mask of network *A* is $\boxed{\text{ A }}$. Based on the network addresses and the subnet mask of network *A*, among the IP addresses shown below, the number of IP addresses that are available for the internal Web server 2 is $\boxed{\text{ B }}$.

[IP address]

10.0.0.2	10.0.0.3	10.0.0.4	10.0.1.1	10.0.1.2
10.0.1.3	10.0.2.1	10.0.2.2	10.0.2.3	10.0.2.4

Answer group for A

a) 255.0.0.0
b) 255.255.0.0
c) 255.255.255.0
d) 255.255.255.128

Answer group for B

a) 1
b) 2
c) 3
d) 4
e) 5
f) 6
g) 7
h) 8

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

Company *D* decides to use DHCP in order to set network information such as IP address for the business PCs. A PC that uses DHCP broadcasts a message to find the DHCP server. Since Company *D* does not install any devices that relay DHCP messages, the DHCP server needs to be installed in $\boxed{\text{C}}$ in order to receive messages from the business PCs.

Company *D* also decides to install a proxy server in order for the business PCs to access Web servers outside of Company *D*. Based on the request from a client, the proxy server accesses a Web server for the client and forwards the response from the Web server to the client. Since Company *D* wants any direct communication between the Internet and the backbone network to remain blocked, the proxy server is installed in $\boxed{\text{D}}$.

The proxy server to be installed has the functionality of a cache server. When the Web page or the image requested from a client is already cached (if there is a cache hit), the cache server sends the cached contents to the client without accessing the Web server again. This reduces the response time. However, when the request does not result in a cache hit, the cache server accesses the Web server, forwards the response from the Web server to the client, and also caches the contents. This results in additional overhead.

If the average response time without using a cache server is 100, the average response time when a cache server is used is 30 in the case of a cache hit and 110 in the case of no cache hit. Under these conditions, if the cache hit rate is $\boxed{\text{E}}$% or more, the average response time when a cache server is used is less than half of the average response time when a cache server is not used.

Answer group for C and D

a) DMZ	b) network *A*
c) network *B*	d) the backbone network

Answer group for E

a) 50	b) 55	c) 60	d) 65
e) 70	f) 75	g) 80	h) 85

Q5. Read the following description of a hotel reservation system, and then answer Subquestions 1 and 2.

Hotel K is planning to develop a hotel reservation system (hereinafter, the system). The system provides information on Hotel K for any person on the Internet (hereinafter, a user). A user can view hotel information but can not make a hotel reservation. When a user would like to make a hotel reservation, the user first registers in the system to become a registered member customer (hereinafter, a member). During the registration process, the user needs to provide a user name, password, email address, a unique social ID, address, phone number and credit card number. Existing members can update the registered profile by using this registration process.

For an online reservation, a member first logs in to the system and checks if desirable rooms are available or not. While performing the reservation process, the member needs to provide the lodging date, number of days, number of guests, and credit card number. At that time, the system checks the validity of the credit card, balance of the account, and payment status. Subsequently, the system books the reservation, generates a unique reservation number, stores the reservation information in the reservation history, and sends a confirmation e-mail to the member.

For members, the system takes 15% discount off the total payment for each reservation. A member needs to confirm the reservation 3 days before the lodging date. If a confirmation is not provided, the reservation is automatically cancelled. In the reservation confirmation and cancellation processes, the member needs to view the reservation history that contains a list of detailed reservation histories for that member.
When the member comes to Hotel K to lodge, the member needs to show the reservation number (that was sent by the confirmation e-mail) to the front desk.

Figure 1 shows the use case diagram of the system.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in Figure 1.

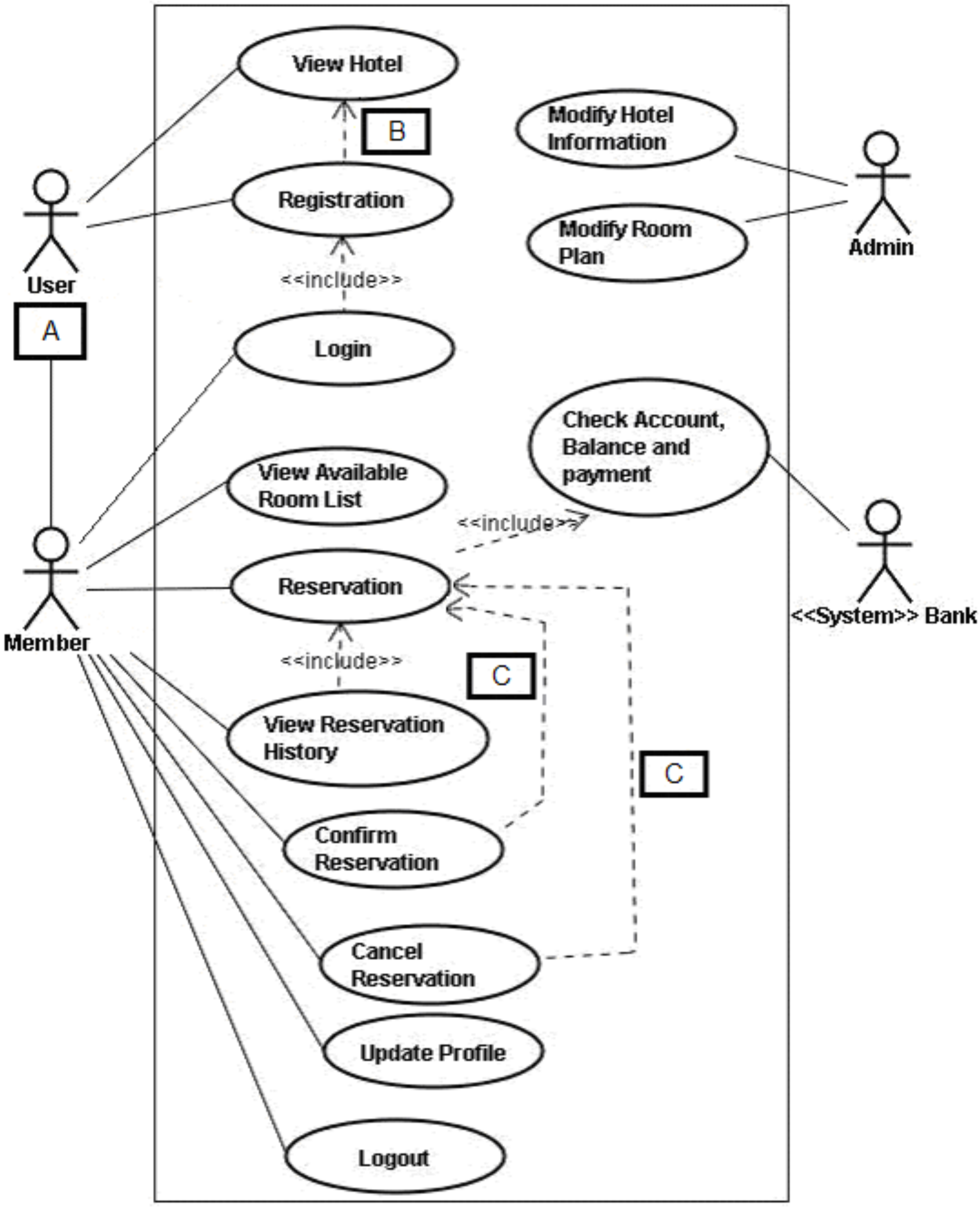


Figure 1 Use case diagram of the system

Answer group for A

a) ◇ (aggregation) b) ↑ (association) c) ♦ (composition) d) △ (generalization)

Answer group for B and C

a) << extend >> b) << include >>

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in Figure 2.

Figure 2 shows the state transition diagram of the system.

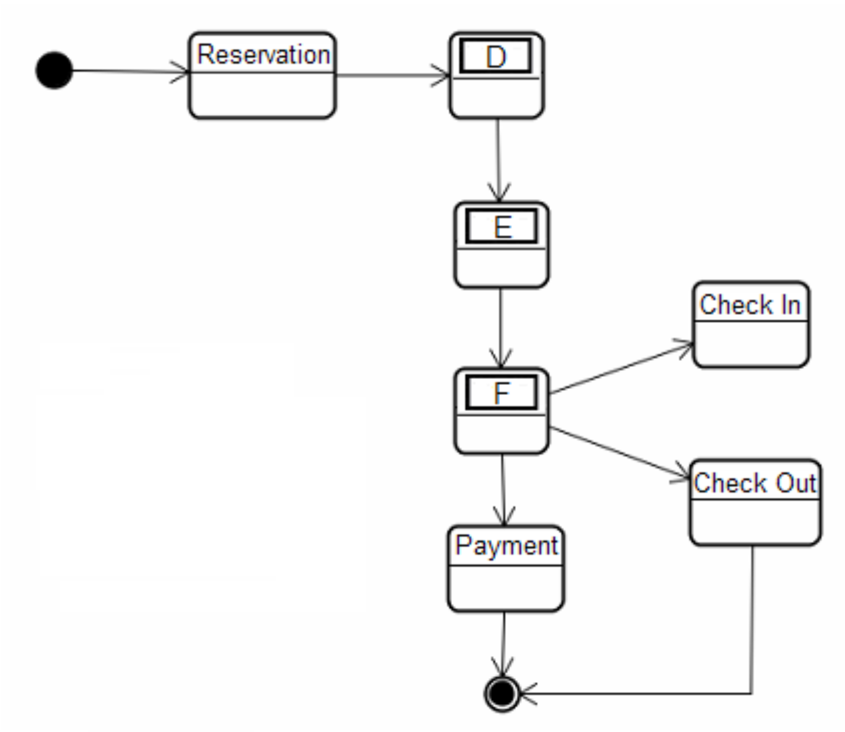


Figure 2 State transition diagram of the system

Answer group

a) Cancel reservation
b) Check available rooms
c) Confirm reservation
d) Login
e) Make reservation
f) Update profile
g) View hotel informatiion

Q6. Read the following description of programs and the programs themselves, and then answer Subquestions 1 through 3.

[Program Description 1]

The cave system in a nature museum is very famous and attracts many visitors to visit and to explore the mysterious beautiful underground.

In the program, people close the way out of some caves to ensure that there is no more than one path between any two caves. In addition, every cave has a guiding machine.

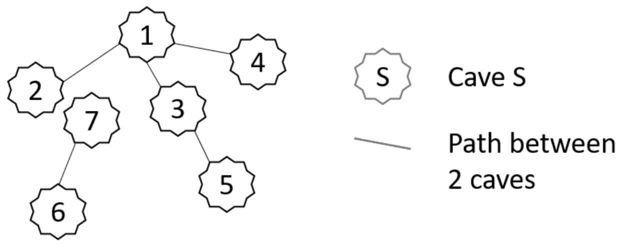


Figure 1 Example of cave map

At cave s, just enter the integer d – the cave that a visitor wants to go to, the machine will display the integer t – the cave adjacent to s that the visitor needs to move into in order to reach cave d; t receives the value -1 if there is no path from s to d. For example, with the cave map shown in Figure 1, at cave number 5 if a visitor wants to go to cave number 4, the machine will show that the visitor needs to go into cave number 3. Continuously using the machine to look up each cave, the visitor will be able to reach the destination (cave number 4).

The subprograms BuildTree and MakeEdge are used to build the trees to store the cave system. The subprogram GuideNext is used to determine the next cave that the visitor should move into in order to reach the desired destination.

(1) Given that N is the number of caves, and K is the number of pairs of A[i] and B[i] which shows the direct link from cave A[i] to cave B[i], where $1 \leq$ A[i], B[i] $\leq$ N, A[i] $\neq$ B[i], and i ranges from 1 to N. Each query at the guiding machine receives the tuples s and d where s is the current cave to put the machine in (in addition to the current cave that the visitor is standing in) and d is the destination cave where the visitor wants to go to. Table 1 illustrates the concrete values in arrays A and B in the case of the tree shown in Figure 1.

Table 1 Values in arrays A[] and B[]

i	1	2	3	4	5
A[i]	1	1	1	3	7
B[i]	2	4	3	5	6

(2) There is no more than one path between any two caves, so a tree data structure can be used to build the cave system. Each cave has one node on the tree, and the link between two caves is the tree edge. In Figure 1, the cave system is composed of two trees. If s and d are in different trees, there is no path from s to d; otherwise, the machine will show the next cave for the next move in order to reach the destination.

(3) Given that each node on the tree has only one parent except the root, an array can be used to store the trees. An array element is mapped to a node on the tree. An index of the array element indicates the tree node's data – the cave number. The data in the array element indicates the node's parent. A negative number indicates that the node has no parent (e.g. root node). For each tuple a and b, set the array such that node b is the parent of node a. If a already had a parent, for example c, reverse the edge between a and c so that node a is parent of node c, and then apply the same mechanism for c recursively.

(4) The argument specification for the subprograms BuildTree and MakeEdge are given in Table 2 and Table 3 respectively.

Table 2 Argument specification for the subprogram BuildTree

Variable	Input/Output	Description
N	Input	The total number of caves
K	Input	The number of direct links between two caves
A[K] B[K]	Input	Two arrays of K elements where pairs of A[i] and B[i] indicate the direct link between two caves A[i] and B[i]
T[N]	Output	The array of N elements that holds the trees for the cave system

Table 3 Argument specification for the subprogram MakeEdge

Variable	Input/Output	Description
T[N]	Input/Output	The array of N elements that holds the trees for the cave system
a b	Input	The direct link between two caves a and b

(5) The indexes of the arrays start at 1.

[Program 1]

```
○ SubProgram: BuildTree(Integer type: N, Integer type: K,
                        Integer type: A[K], Integer type: B[K],
                        Integer type: T[N])
  ○ Integer type: I

  ■ I: 1, N, 1
  │ • T[I] ← -1
  ■
  ■ I: 1, K, 1
  │ • MakeEdge(T, A[I], B[I])
  ■

○ SubProgram: MakeEdge(Integer type: N, Integer type: a,
                       Integer type: b, Integer type: T[N])
  ▲ T[a] ≠ -1
  │ • MakeEdge(T, T[a], a)
  ▼
  • T[a] ← b
```

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank [ ] in the following description.

Assume that arrays A[] and B[] contain the values shown in Table 4, and T[] is array of integers with 12 elements.
After the execution of BuildTree(12, 10, A[], B[], T[]), T[] should be set as follows:

T[] = { 4, 1, [A], 8, 2, [B], 3, -1, 10, 12, 10, -1 }

Table 4 Example of arrays A[i] and B[i]

i	1	2	3	4	5	6	7	8	9	10
A[i]	1	1	1	2	3	3	4	9	10	10
B[i]	2	3	4	5	6	7	8	10	11	12

Answer group for A and B

a) 0 b) 1 c) 2
d) 3 e) 4 f) 5

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in Program 2.

[Program Description 2]

(1) Based on the trees for the cave system, subprogram `GuideNext` is used to find the next cave for a visitor at cave `s` who wants to go to cave `d`. The algorithm finds the path between `s` and `d` via their ancestors. If `s` and `d` have the same ancestor then the next cave is `s`'s ancestor. If `d` is an ancestor of `s` then the next cave is `s`'s ancestor. If `s` is an ancestor of `d` then the next cave is one of `s`'s descendants. If `s` and `d` are in different trees then there is no path from `s` to `d` and `-1` is stored to `t`.

(2) The argument specification for the subprogram `GuideNext` is given in Table 5.

Table 5 Argument specification for the program `GuideNext`

Variable	Input/Output	Description
`N`	Input	The total number of caves
`T[N]`	Input	The array of `N` elements that holds the trees for the cave system
`s`	Input	The starting cave
`d`	Input	The destination cave
`t`	Output	The next cave that a visitor needs to move into to reach `d`

(3) The indexes of the arrays start at 1.

[Program 2]

```
○ Program: GuideNext(Integer type: N, Integer type: T[N],
                     Integer type: s, Integer type: d,
                     Integer type: t)
  ○ Integer type: sa, da, dd, num, C[N]

  • sa ← s
  • da ← d
  ■ I: 1, N, 1
  │ • C[I] ← -1
  ■
  • t ← [    C    ]
  • C[s] ← 1
  • C[d] ← 1
  ■
  │ • num ← 0                        ← α
  │ • dd ← da
  │ ▲ sa ≠ -1
  │ │ • sa ← T[sa]
  │ ▼
  │ ▲ da ≠ -1
  │ │ • da ← T[da]                   ← β
  │ ▼
  │ ▲ sa ≠ -1
  │ │ ▲ C[sa] ≠ -1
  │ │ │ • t ← [    D    ]
  │ │ │ • return
  │ │ ▼
  │ │ • C[sa] ← 1
  │ │ • num ← num + 1
  │ ▼
  │ ▲ da ≠ -1
  │ │ ▲ C[da] ≠ -1
  │ │ │ ▲ da = s
  │ │ │ │ • t ← [    E    ]
  │ │ │ ├──────
  │ │ │ │ • t ← [    D    ]
  │ │ │ ▼
  │ │ │ • return
  │ │ ▼
  │ │ • C[da] ← 1
  │ │ • num ← num + 1
  │ ▼
  ■ num > 0
```

Answer group for C through E

a) `-1` b) `0` c) `T[d]` d) `T[s]`

e) `T[dd]` f) `da` g) `dd` h) `sa`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank [ ] in the following description.

Set `T[]` by calling `BuildTree` with arrays `A[i]` and `B[i]` shown in Table 4. Then, by using `T[]` set by `BuildTree`, call `GuideNext(12, T[], 5, 8, t)`. After the execution of `GuideNext`, [F] is set to variable `t`. Until the program `GuideNext` is terminated, the statement surrounded by the dashed rectangle that is pointed by "← α" will be executed [G] time(s) and the statement pointed by "← β" will be executed [H] time(s).

Answer group for F through H

a) 0 b) 1 c) 2 d) 3

e) 4 f) 5 g) 6 h) 7

i) 8 j) 9

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestion.

An integer number is called a "non-decreasing number" if all the digits from left to right are in a non-decreasing form. See the following examples:

(1) Given an integer 124, it is composed of 3 digits that are 1, 2, and 4. Comparing each pair of the digits from left to right, 1 and 2, the value increases from 1 to 2. When comparing 2 and 4, the value is also increasing. Therefore, the integer 124 is a non-decreasing number.
(2) Given an integer 221, when comparing 2 and 2, they are the same and are non-decreasing. However, when comparing 2 and 1, the value is decreasing. Therefore, the integer 221 is not a non-decreasing number.
(3) 1-digit numbers are regarded as non-decreasing numbers.

[Program Description]

(1) The program reads two integer numbers, `num1` and `num2`, from the standard input. Here, `num1` and `num2` satisfy the following three conditions: (i) 0 < `num1`, (ii) `num1` ≤ `num2`, and (iii) `num2` < 32768.
(2) The program tests all the integer numbers that lie between `num1` and `num2`.
(3) The program outputs the total count of non-decreasing numbers that lie between `num1` and `num2`. This count is stored in the variable `ndCount`.
(4) An example of the output of this program is as follows:
```
Enter two integers: 18 23
Count of non-decreasing numbers between 18 and 23: 4
```
(5) The following two user-defined functions are used in the program:
   (i) `int isNonDecreasingNumber(char *s)`
   The function returns 1 if the input number in the string `*s` is a non-decreasing number, otherwise, returns 0.
   (ii) `void convert(int num, char *s)`
   The function converts the integer number `num` into the character string `*s`. For example, an integer 135 is converted into "135".
(6) The following library function is used in the program:
   `size_t strlen(const char *s)`
   The function returns the length of the string `*s` (not including the terminating '\0').

[Program]

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 5 + 1

int isNonDecreasingNumber(char *s);
void convert(int num, char *s);

int isNonDecreasingNumber(char *s) {
   int i;

   for (i = 0; i < strlen(s) - 1; i++) {
      if (    A     >     B    ) {
         return 0;
      }
   }
   return 1;
}

void convert(int num, char *s) {
   int i, digits = 0;
   char buffer[MAX_LEN];

   while (num != 0) {
      buffer[digits] =     C     + '0';
      num =     D    ;
      digits++;
   }
   for (i = 0; i < digits; i++) {
      s[i] =     E    ;
   }
       F    ;
}

int main() {
   int num1, num2, x;
   int ndCount = 0;
   char numString[MAX_LEN];

   printf("Enter two integers: ");
   scanf("%d %d", &num1, &num2);
```

```
    for (x = num1; x <= num2; x++) {
        convert(x, numString);
        if (    G    ) {
            ndCount++;
        }
    }
    printf("Count of non-decreasing numbers between %d and %d: %d\n",
           num1, num2, ndCount);
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A and B

a) `&s[i-1]`
b) `&s[i]`
c) `&s[i+1]`
d) `s[i-1]`
e) `s[i]`
f) `s[i+1]`

Answer group for C and D

a) `digits`
b) `digits++`
c) `num`
d) `num % 10`
e) `num * 10`
f) `num + 10`
g) `num - 10`
h) `num / 10`
i) `num ^ 2`

Answer group for E

a) `buffer[digits - 1]`
b) `buffer[digits - i - 1]`
c) `buffer[digits - i]`
d) `buffer[digits - i + 1]`
e) `buffer[i]`
f) `buffer[i + digits]`

Answer group for F

a) `buffer[digits] = '0'`
b) `buffer[digits] = '\0'`
c) `buffer[digits] = s[digits]`
d) `digits++`
e) `digits = 0`
f) `s[digits] = '0'`
g) `s[digits] = '\0'`
h) `s[digits] = buffer[digits]`

Answer group for G

a) `!isNonDecreasingNumber(*numString)`
b) `!isNonDecreasingNumber(numString)`
c) `isNonDecreasingNumber(*numString)`
d) `isNonDecreasingNumber(numString)`

Q8. Read the following description of a Java programs and the programs themselves, and then answer Subquestions 1 through 3.

[Program Description]

The programs consist of classes containing some utility methods for manipulating numbers and iterating through the digits of a number from left to right (from the most significant digit) or right to left (from the least significant digit) as described below. There is a class to test those methods.

Subquestion 1

Program 1 is the utility class `IntegerToText` containing the following methods.

(1) The `countCharacters` method returns the number of digits present in a specified number `n`. If the specified number is negative, it returns an extra character count for the minus sign. For example,

  (i) if `n` = 7, it returns 1,
  (ii) if `n` = -7, it returns 2,
  (iii) if `n` = 789, it returns 3.

(2) The `convert` method returns a `String` array representing the digits in words present in a specified number `n`. If the specified number is negative, the minus sign is represented as `"minus"`. For example,

  (i) if `n` = 7, it returns an array consisting of a word: `["seven"]`
  (ii) if `n` = -7, it returns an array consisting of the words: `["minus", "seven"]`
  (iii) if `n` = 79, it returns an array consisting of the words: `["seven", "nine"]`

From the answer groups below, select the correct answer to be inserted in each blank [ ] in Program 1 so that the methods in Program 1 work as described above.

[Program 1]

```
public final class IntegerToText {
   private static final String[] WORDS = {
      "zero", "one", "two", "three", "four",
      "five", "six", "seven", "eight", "nine"
   };

   public static int countCharacters(int n) {
      int characterCount =  (n < 0) ? 1 : 0;
      do {
         n [   A   ] 10;
         ++characterCount;
      } while (n != 0);
      return characterCount;
   }
```

```
    public static String[] convert(int n) {
        int characterCount = countCharacters(n);
        String[] inWords = new String[characterCount];
        long m = n;
        if (n < 0) {
            m = -m;
            inWords[0] = "minus";
        }

        int i = inWords.length;
        do {
            inWords[--i] = WORDS[(int) (m % 10)];
            m [    A    ] 10;
        } while (m [    B    ] 0);

        return inWords;
    }
}
```

Answer group for A

a) *= b) -= c) %= d) /=

Answer group for B

a) < b) <= c) > d) >=

Subquestion 2

Program 2 is the abstract class `NumberIterator` that implements the interface `Iterator<String>` to iterate through the *in-words representation* of a number. The in-words representation is a series of words representing all digits of a number, optionally with its negative sign. For example, -79 is represented in the in-words representation by three words: "minus", "seven", and "nine".

The `NumberIterator` class has the following public instance methods defined in `Iterator`.

(1) The `hasNext` method returns `true` if the iteration has more words, or `false` otherwise.

(2) The `next` method returns the next word, or throws a `NoSuchElementException` if there is no next word.

(3) The `remove` method always throws an `UnsupportedOperationException`.

There are two nested classes in `NumberIterator` implementing iterations through the in-words representation. One is the `LeftToRightIterator` that supports the left-to-right iteration, and the other is the `RightToLeftIterator` that supports the right-to-left iteration.

(1) In the case of the `LeftToRightIterator`, the `next` method returns the next word from the left side, corresponding to the most significant digit. For example, if the number is 79, its in-words representation is `"seven"`, `"nine"`. The first call to the method will return `"seven"`. The second call will return `"nine"`. The third call will throw a `NoSuchElementException`.

(2) In the case of the `RightToLeftIterator`, the `next` method returns the next word from the right side, corresponding to the least significant digit. For example, if the number is 79, its in-words representation is `"seven"`, `"nine"`. The first call to the method will return `"nine"`. The second call will return `"seven"`. The third call will throw a `NoSuchElementException`.

The `NumberIterator` class has two static factory methods, `getLeftToRightNumberIterator` and `getRightToLeftNumberIterator`, each of which returns a concrete instance of `NumberIterator` created from the specified number.

From the answer groups below, select the correct answer to be inserted in each blank ☐ in Program 2.

[Program 2]

```
import java.util.Iterator;
import java.util.NoSuchElementException;

public abstract class NumberIterator implements Iterator<String> {
   final String[] words;
   int currentPosition;

   private NumberIterator(String[] words, boolean isLeftToRight) {
      if (words == null) {
         throw new NullPointerException();
      }
      this.words = words;
      if (isLeftToRight) {
         this.currentPosition = 0;
      } else {
         this.currentPosition = words.length - 1;
      }
   }
```

```
public static NumberIterator getLeftToRightNumberIterator(int n) {
   String[] inwords = IntegerToText.convert(n);
   return new LeftToRightIterator(inwords);
}

public static NumberIterator getRightToLeftNumberIterator(int n) {
   String[] inwords = IntegerToText.convert(n);
   return new RightToLeftIterator(inwords);
}

public String next() {
   if (!hasNext()) {
      throw new NoSuchElementException();
   }
   return nextWord();
}

public void remove() {
   throw new UnsupportedOperationException();
}

abstract String nextWord();

private static class LeftToRightIterator extends NumberIterator {
   private LeftToRightIterator(String[] words) {
      super(words, true);
   }

   public boolean hasNext() {
      return [     C     ];
   }

   String nextWord() {
      return words[     D     ];
   }
}

private static class RightToLeftIterator extends NumberIterator {
   private RightToLeftIterator(String[] words) {
      super(words, false);
   }

   public boolean hasNext() {
      return [     E     ];
   }
```

```
        String nextWord() {
            return words[    F    ];
        }
    }
}
```

Answer group for C and E

a) `currentPosition < words.length`
b) `currentPosition < words.length - 1`
c) `currentPosition <= words.length`
d) `currentPosition > 0`
e) `currentPosition >= 0`

Answer group for D and F

a) `++currentPosition`
b) `--currentPosition`
c) `currentPosition`
d) `currentPosition + 1`
e) `currentPosition - 1`
f) `currentPosition++`
g) `currentPosition--`

Subquestion 3

From the answer group below, select the correct answer to be inserted in the blank [ ] in the following description.

Program 3 is the `Tester` class to test Program 1 and Program 2.

The lines shown in Figure 1 are outputted when the `main` method of the `Tester` class is executed.

```
1 character(s) of 0 in words from right side -> zero
3 character(s) of 100 in words from left side -> one zero zero
3 character(s) of 100 in words from right side -> zero zero one
4 character(s) of -857 in words from [    G    ]
```

Figure 1 Output of the `main` method of the `Tester` class

[Program 3]

```
public class Tester {
   public static void main(String[] args) {
      int[] testData = {0, 100, -857};
      for (int n : testData) {
         if (n > 0) {
            System.out.print(IntegerToText.countCharacters(n)
                  + " character(s) of " + n
                  + " in words from left side ->");
            NumberIterator l2r;
            l2r = NumberIterator.getLeftToRightNumberIterator(n);
            while (l2r.hasNext()) {
               System.out.print(" " + l2r.next());
            }
            System.out.println();
         }
         System.out.print(IntegerToText.countCharacters(n)
               + " character(s) of " + n
               + " in words from right side ->");
         NumberIterator r2l;
         r2l = NumberIterator.getRightToLeftNumberIterator(n);
         while (r2l.hasNext()) {
            System.out.print(" " + r2l.next());
         }
         System.out.println();
      }
   }
}
```

Answer group for G

a) `left side -> eight five seven minus`

b) `left side -> minus eight five seven`

c) `left side -> seven five eight minus`

d) `right side -> minus eight five seven`

e) `right side -> minus seven five eight`

f) `right side -> seven five eight minus`

25th ITPEC FE Afternoon Exam -- March 2018

Q	SQ	BQ	Group	Correct Answer
1	1	A		a
		B		b
		C		e
	2			d
	3		A	b
			A	d
2	1	A		d
		B		c
		C		f
		D		c
	2	E		b
	3	F		c
3	1	A		b
		B		d
	2	C		c/d
	3	D		b
		E		e
4	1	A		c
		B		b
	2	C		c
		D		a
		E		f
5	1	A		d
		B		a
		C		b
	2	D		b
		E		e
		F		c
6	1	A		b
		B		d
	2	C		a
		D		d
		E		g
	3	F		c
		G		e
		H		b
7	1	A		e
		B		f
		C		d
		D		h
		E		b
		F		g
		G		d
8	1	A		d
		B		c
	2	C		a
		D		f
		E		e
		F		g
	3	G		f

ITPEC Information Technology Professionals Examination Council

October 2017

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the Ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the autumn Fundamental IT Engineer Examination conducted?

   Answer group
   a) September b) October c) November d) December

   Since the correct answer is " b) October ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc., of procedures, variables, etc.
/* text */		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 ――― process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then process 1 is executed. If it is `false`, then process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the iteration process, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	`+`, `−`, `not`	High
Multiplication, division	`×`, `÷`, `%`	↑
Addition, subtraction	`+`, `−`	
Relational operation	`>`, `<`, `≥`, `≤`, `=`, `≠`	
Logical product	`and`	↓
Logical sum	`or`	Low

Note: With division of integers, an integer quotient is returned as a result.
The “%” operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description of authenticity and confidentiality of e-mail data, and then answer Subquestions 1 through 3.

Electronic mail, i.e. e-mail, is the most extensively used network-based distributed application. E-mails may be sent and received to and from various system architectures, host operating systems, vendor platforms and communication suites, rendering it an effective tool for communication. E-mails are likely to contain critical information that is to be secured. The security of an e-mail is maintained by ensuring its authenticity and confidentiality.

PGP (Pretty Good Privacy) is a remarkable technique to check the authenticity of e-mail data. This technique provides authentication services to an e-mail by incorporating a digital signature in it.

At the sender side (Figure 1), PGP generates the digital signature from the e-mail body using SHA-1 (Secure Hash Algorithm-1) according to the following steps:

(1) SHA-1 generates a 160-bit hash code of the message.
(2) The hash code is encrypted with the RSA algorithm using the sender's private key.
(3) The result is appended to the message.

The message is then compressed at a ratio of approximately 2.0 using the ZIP algorithm. The compressed message enables the effective use of transmission and storage resources.

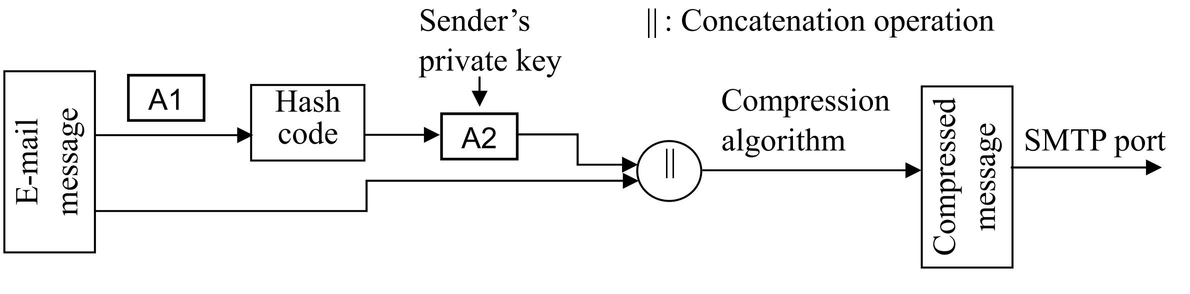


Figure 1 Authentication operations at the sender side

At the receiver side (Figure 2), the digital signature is separated from the incoming e-mail message after the message is decompressed. Further operations proceed according to the following steps:

(1) Decrypt the hash code from the digital signature by using the sender's public key.
(2) Generates a new hash code from the received e-mail message.
(3) If the two hash codes match, the message is verified to be authentic.

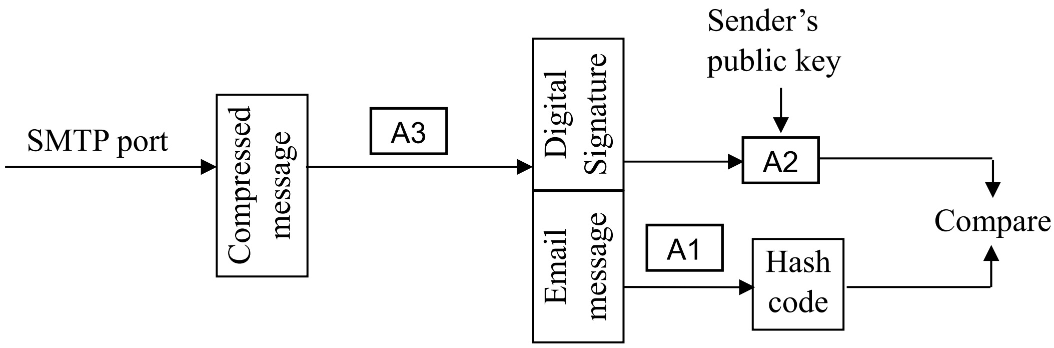


Figure 2 Authentication operations at the receiver side

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank in Figure 1 and Figure 2. Here, the answers to be inserted in A1, A2 and A3 should be selected as the correct combination from the answer group for A.

Answer group for A

	A1	A2	A3
a)	RSA algorithm	Compression algorithm	ZIP algorithm
b)	RSA algorithm	Decompression algorithm	ZIP algorithm
c)	SHA-1	RSA algorithm	Compression algorithm
d)	SHA-1	RSA algorithm	Decompression algorithm
e)	ZIP algorithm	SHA-1	RSA algorithm
f)	ZIP algorithm	SHA-1	RSA algorithm

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank in Figure 3.

In PGP, a symmetric 128-bit session key is generated and used each time a new e-mail is sent. The key is bound to the message and transmitted with it. The effectiveness of this method is limited to ensuring the possessor's authenticity; it does not ensure the confidentiality of the e-mail.

For ensuring the confidentiality of e-mail contents, the symmetric encryption algorithm may be used to encrypt an e-mail message in 64-bit cipher feedback mode. The session key is encrypted by using the receiver's public key. Figure 3 shows the operations for ensuring both authenticity and confidentiality of the e-mail body.

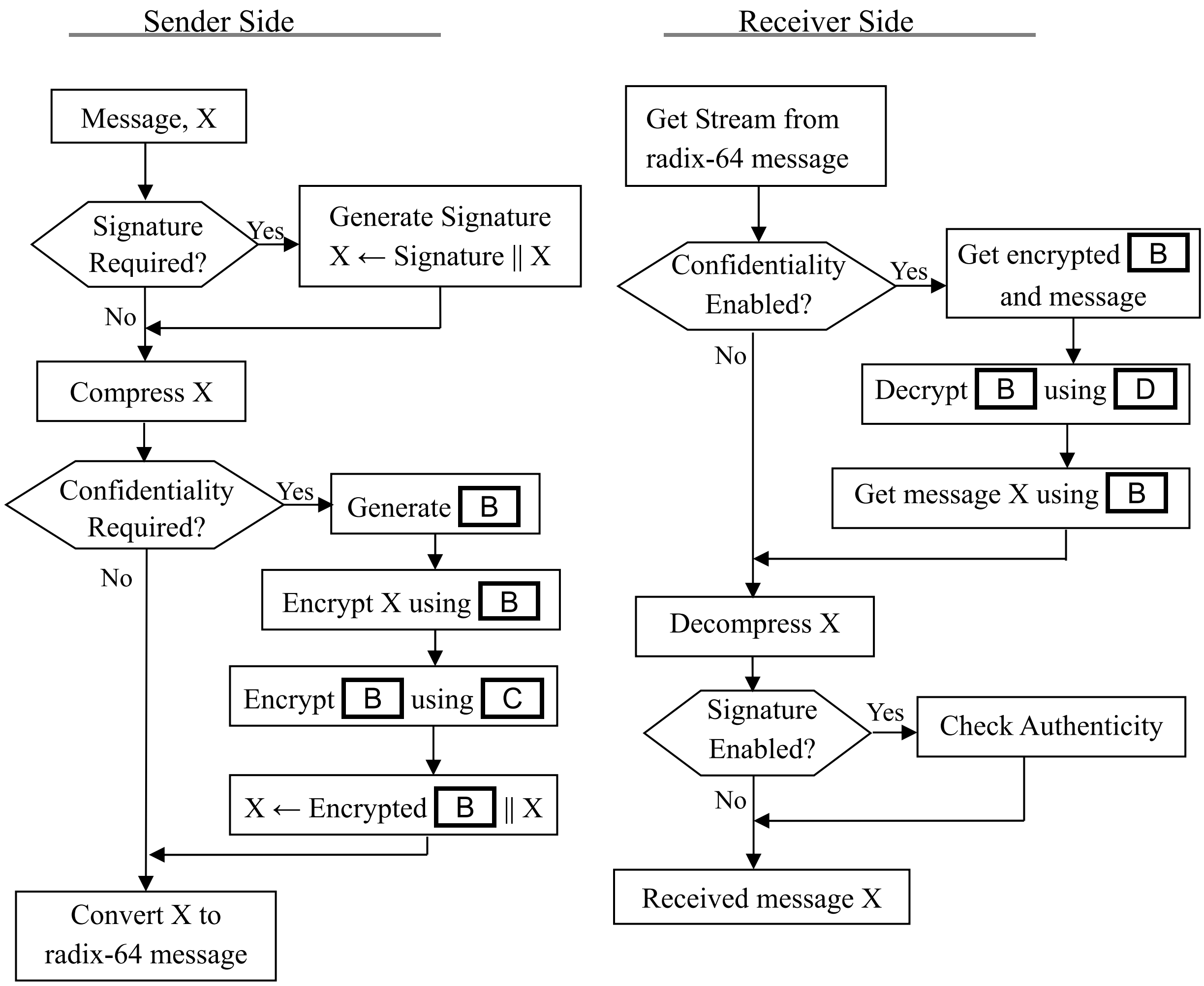


Figure 3 Operations for ensuring both authenticity and confidentiality of the e-mail body

Answer group for B through D

a) Compression key
b) Hash key
c) Receiver's private key
d) Receiver's public key
e) Sender's private key
f) Sender's public key
g) Session key

Subquestion 3

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following description.

The digital signature is generated from the plaintext rather than the encrypted or compressed message. This ensures that for the purpose of signature verification, a third-party need not be concerned with the symmetric key or compression algorithm. Message encryption is applied after compression in order to strengthen cryptography security because the compressed message exhibits less redundancy than the original plaintext. Less redundancy renders it more challenging for a hacker to perform cryptanalysis process.

The resulting encrypted message at the sender side is a stream of 8-bit binary codes. However, numerous e-mail systems permit the use of only ASCII characters. To accommodate this restriction, PGP provides the service of converting the 8-bit binary stream to a stream of ASCII characters using radix-64 conversion. Each group of three 8-bit binary codes is mapped into four ASCII characters, expanding the message by 33%.

For example, when a sender sends an e-mail consisting of 9k ASCII characters, it is transmitted over the Internet as an 8-bit binary stream of size $\boxed{\quad E \quad}$ kB. Assume that the ZIP algorithm exhibits a compression ratio of 2.0 and that the sizes of the digital signature and encrypted session key are negligible.

Answer group for E

a) 4.5	b) 6	c) 8
d) 9	e) 12	f) 16

Q2. Read the following description of shared resource access, and then answer Subquestion.

In a multi-programming operating system, there is a possibility of accessing the same resource by two or more processes simultaneously. The mutually exclusive control ensures that two or more concurrent processes do not enter their critical sections simultaneously. A critical section is a segment of a process that cannot be executed concurrently by two or more processes.

It is necessary for the operating system to ensure that:

(1) if one process is accessing a shared resource such as modifiable data, the other processes are excluded from using that resource,

(2) each process receives enough processor resource to function effectively, and

(3) the processor makes an effort to minimize its idle status.

[Mutually exclusive control - Case 1]

In a processor, two processes P0 and P1 are running concurrently. Both P0 and P1 attempt to update the shared data. The processor maintains data consistency by permitting only one process to update the shared data, and preventing the other process from accessing the shared data while the permitted process is updating that data.

A global variable `turn`, shared by P0 and P1, is used for mutually exclusive control. The initial value of `turn` is 0.

Figure 1 shows the mutually exclusive control for P0 and P1.

```
PROCESS P0 {                                  PROCESS P1 {
   ...                                           ...
   WHILE (turn is not 0) {                       WHILE (turn is not 1) {
      WAIT for a predefined time unit               WAIT for a predefined time unit
   }                                             }
   UPDATE shared data                            UPDATE shared data
   SET turn ← 1                                  SET turn ← 0
   ...                                           ...
}                                             }
```

Figure 1 Mutually exclusive control for P0 and P1

Table 1 shows, as an example, the functioning of this mutually exclusive control.

Table 1 Execution status of P0 and P1 in time sequence

Time	Execution status of P0 and P1
t	• P0 and P1 are loaded and started. The initial value of `turn` is 0. • Both P0 and P1 attempt to access the shared data. • Then [A] enters into its critical section and accesses the shared data.
$t+1$	• P1 continues attempt to access the shared data. • P1 continues to wait in the loop because the value of `turn` is [B].
$t+2$	• P0 completes updating the shared data.
$t+3$	• P1 continues attempt to access the shared data. • Subsequently, P1 enters into its critical section and accesses the shared data.
$t+4$	• P0 attempts to access the shared data again. • P0 continues to wait in the loop.
$t+5$	• P1 completes updating the shared data.
$t+6$	• P0 continues attempt to access the shared data. • Subsequently, P0 enters into its critical section and accesses the shared data.
$t+7$	• P0 completes updating the shared data. • At this point, the value of `turn` is [C].
$t+8$	• ...

This mutually exclusive control for P0 and P1 is likely to involve a problem. For example, at the time $t+8$ in Table 1, a problem occurs under the situation where [D].

[Mutually exclusive control - Case 2]

To solve the problem in Case 1, an alternative mutually exclusive control is proposed by introducing two global variables `flag0` and `flag1`, instead of using a single variable `turn`. The initial value of both `flag0` and `flag1` is set to 0.

Figure 2 shows the alternative mutually exclusive control for P0 and P1.

```
PROCESS P0 {
    ...
    SET flag0 ← 1
    WHILE (flag1 is 1) {
        WAIT for a predefined time unit
    }
    UPDATE shared data
    SET flag0 ← 0
    ...
}
```

```
PROCESS P1 {
    ...
    SET flag1 ← 1
    WHILE (flag0 is 1) {
        WAIT for a predefined time unit
    }
    UPDATE shared data
    SET flag1 ← 0
    ...
}
```

Figure 2 Alternative mutually exclusive control for P0 and P1

However, this mutually exclusive control for P0 and P1 is also likely to involve a problem. For example, a problem occurs under the situation where [E].

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the description.

Answer group for A through C

a) 0 b) 1 c) P0 d) P1

Answer group for D

a) Both P0 and P1 attempt to access the shared data.
b) Both P0 and P1 have no more need to access the shared data.
c) P0 attempts to access the shared data; however, P1 has no more need to access it.
d) P1 attempts to access the shared data; however, P0 has no more need to access it.

Answer group for E

a) Both P0 and P1 update the shared data exactly once.
b) P0 (or P1) updates the shared data continuously.
c) P0 sets `flag0` to 1, and P1 sets `flag1` to 1 simultaneously.
d) The number of times the shared data is updated by P0 differs from that by P1.

Q3. Read the following description of a relational database, and then answer Subquestions 1 through 3.

An IT training center in a city is developing a relational database to manage its training courses and students.

The relational database is composed of three tables: COURSE table, BATCH table, and STUDENT table. Each course has numerous batches. Moreover, each batch of a course has a number of students.

For each table, the table structure with its sample data are shown below. Here, a solid underline indicates a primary key, and a dotted underline indicates a foreign key.

(1) COURSE table

COURSE table contains the information about the course code, course name, and description.

<u>CourseCode</u>	CourseName	Description
10	Database	For database engineers
20	Programming	For C and Java programmers
30	Network	For network engineers

(2) BATCH table

BATCH table contains the information about the batch code, starting date, duration, course fee, net income, expected income, and course code. A course has numerous batches.

<u>BatchCode</u>	StartingDate	Duration	CourseFee	NetIncome	ExpectedIncome	CourseCode
101709	2017-09-04	2	1,000	8,000	10,000	10
101710	2017-10-02	2	1,000	16,000	10,000	10
201709	2017-09-11	4	2,000	16,000	20,000	20
301709	2017-09-25	3	1,500	18,000	15,000	30

(3) STUDENT table

STUDENT table contains the information about the student ID, student name, age, address, and email address. A student is allowed to attend any number of courses. When a student makes an application for a course, he/she specifies the batch he/she wants to attend.

<u>StudentID</u>	StudentName	Age	Address	EmailAddress
1	Ms. Alice	20	East 11, City S	alice@example.com
2	Mr. Bobby	28	North 22, City T	bobby@example.com
3	Mr. Charles	24	South 33, City U	charles@example.com

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\quad\quad}$ in the following SQL statement.

In BATCH table, the ExpectedIncome column shows the planned income at the break-even point of profit and loss, and the NetIncome column shows the actual income obtained from the batch. Therefore, if NetIncome > ExpectedIncome, the batch gains a profit, and if NetIncome < ExpectedIncome, the batch suffers a loss. For the batch that is not completed, the NetIncome column has the value 0.

The following SQL statement outputs the course code, course name, and net profit of each course in descending order of net profit.

```
SELECT T.CourseCode, C.CourseName, T.Profit AS NetProfit
FROM (SELECT B.CourseCode, [    A    ] AS Profit
      FROM BATCH B
      GROUP BY B.CourseCode) T, COURSE C
WHERE [    B    ]
ORDER BY NetProfit DESC
```

The SQL statement creates the following output from the sample data of COURSE table and BATCH table shown in the description. For example, for course code 10, the batch code 101709 suffers a loss of 2,000, and the batch code 101710 gains a profit of 6,000. Therefore, the course code 10 gains a total profit of 4,000.

CourseCode	CourseName	NetProfit
10	Database	4,000
30	Network	3,000
20	Programming	-4,000

Answer group for A

a) `SUM(B.ExpectedIncome) - SUM(B.NetIncome)`
b) `SUM(B.NetIncome) - SUM(B.ExpectedIncome)`
c) `SUM(T.ExpectedIncome) - SUM(T.NetIncome)`
d) `SUM(T.NetIncome) - SUM(T.ExpectedIncome)`

Answer group for B

a) `B.BatchCode = T.BatchCode`
b) `B.CourseCode = C.CourseCode`
c) `B.CourseCode = T.CourseCode`
d) `C.CourseCode = T.CourseCode`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following SQL statement.

Training administrators wish to determine the average net income for each course.
The following SQL statement outputs the course code and average net income for each course that have an average net income greater than 15,000.

```
SELECT CourseCode, AVG(NetIncome) AS AveNetIncome
FROM BATCH
[    C    ] CourseCode
[    D    ] AVG(NetIncome) > 15000
```

The SQL statement creates the following output from the sample data of BATCH table shown in the description.

CourseCode	AveNetIncome
20	16,000
30	18,000

Answer group for C and D

a) `GROUP BY`
b) `HAVING`
c) `ORDER BY`
d) `WHERE`

Subquestion 3

A lecturer of a training course wishes to have a list of the students of the forthcoming batch. A database engineer intends to provide the list as shown below.

```
Course code: 10          Batch code: 101710
Course name: Database    Starting date: 2017-10-02

Student ID  Student name  Age
       1    Ms. Alice      20
       3    Mr. Charles    24
      ...     ...         ...
```

In order to provide the list, the database engineer is required to modify the structure of the tables.

From the answer group below, select the most appropriate action to modify the structure of the tables considering data redundancy and data integrity.

Answer group

a) Add the BatchCode column to STUDENT table.
b) Add the StudentID column to BATCH table.
c) Add the StudentID column to COURSE table.
d) Create a new table that consists of BatchCode and StudentID columns.
e) Create a new table that consists of BatchCode, CourseCode and StudentID columns.

Q4. Read the following description of an installation of a small-size business network, and then answer Subquestions 1 through 3.

Company V intends to install a small-size business network (hereinafter, internal network). IP addresses 10.1.1.0/24 are used for all the hosts and devices in the internal network excluding the hosts in the development department. IP addresses 10.1.2.0/25 are used for all the hosts in the development department.

There are several network segments. Variable length subnet masks are used for various network segments. They are connected to each other through the Central Router (CR), Branch 1 Router (BR1), and Branch 2 Router (BR2). To access the Internet from the internal network, NAT function is configured at CR. A packet filtering policy is also configured at CR to filter the TCP/IP traffic between the internal network and the Internet.

Figure 1 shows the configuration of the internal network, and Table 1 shows the routing table of BR1, including administrative distance (AD). An AD value is used by routers to determine the optimum route that is to be used when multiple paths to a destination exist. An AD is an integer from 0 to 255, where 0 is the most trusted, and 255 implies no traffic is to be passed via this route. A routing protocol with a lower AD value is considered as a "more preferred" path.

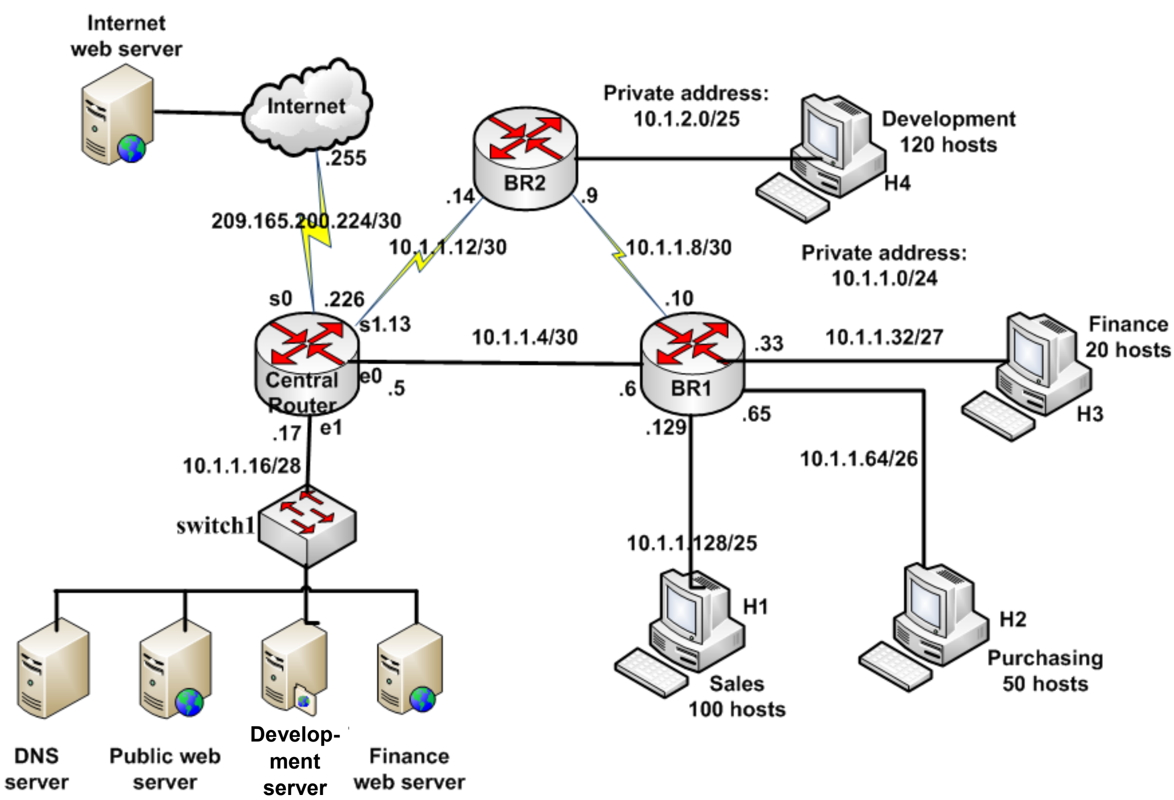


Figure 1 Configuration of the internal network

In Figure 1, the following symbols are used:

s*n*: Serial interface *n*, where *n* is an interface identification number

e*n*: Ethernet interface *n*, where *n* is an interface identification number

.*n*: Rightmost part of the IP address of the interface

For example, **s1 .13** implies that the IP address of serial interface **1** of CR is 10.1.1**.13** on the segment 10.1.1.12/30, and **e0 .5** implies that the IP address of Ethernet interface **0** of CR is 10.1.1**.5** on the segment 10.1.1.4/30.

Table 1 Routing table of BR1

Network	AD	Interface	Next-hop
10.1.1.4/30	0	e0	directly connected
10.1.1.32/27	0	e1	directly connected
10.1.1.64/26	0	e2	directly connected
10.1.1.128/25	0	e3	directly connected
10.1.1.8/30	0	s0	directly connected
10.1.1.16/28	1	e0	10.1.1.5
10.1.1.12/30	1	s0	10.1.1.9
10.1.2.0/25	1	s0	10.1.1.9
any/any	1	e0	10.1.1.5
any/any	50	s0	10.1.1.9

Subquestion 1

From the answer group below, select the correct answer to be inserted in the blank [ ] in the following description.

Concerning the hosts H2 in the purchasing department, a host creates a request to the Internet Web server to access the Internet Web services. BR1 receives this request packet via interface e2. According to the routing table, BR1 sends the packet via interface [A] for the Internet Web services.

Answer group for A

a) e0 b) e1 c) e2

d) e3 e) s0

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the following description.

A network administrator intends to install new servers in the server farm on the subnet 10.1.1.16/28. The maximum number of IP addresses (excluding the address of the router) that can be assigned to the servers in the server farm is $\boxed{\text{B}}$.

During the network test, a problem occurred. None of the hosts in the internal network could access the DNS server. The network administrator followed the trouble-shooting steps by using tools and issuing the ping commands, and determined that the hosts could communicate with the local network of BR1 (10.1.1.4/30, 10.1.1.128/25, 10.1.1.8/30) although not with the remote network address of the DNS server. By referring to the server information in Table 2 and the test results, the network administrator concluded that the cause of the problem was $\boxed{\text{C}}$.

Table 2 Server information

Server name	IP address/netmask	Default gateway
DNS server	10.1.1.31/28	10.1.1.17
Public web server	10.1.1.30/28	10.1.1.17
Development server	10.1.1.29/28	10.1.1.17
Finance web server	10.1.1.28/28	10.1.1.17

Answer group for B

a) 4
b) 5
c) 13
d) 14
e) 15

Answer group for C

a) TCP/IP has not been correctly installed on the hosts.
b) The default gateway on BR1 is incorrect.
c) The DNS server has been configured with the broadcast address of the subnet.
d) The network interface card installed in the hosts is not functioning.

Subquestion 3

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

On the development server in the server farm, Web services are running. These Web services are accessed only from Web browsers running on the hosts H4 in the development department. The network administrator is considering security enhancement to limit the accesses to these Web services by adding the packet filtering policy on CR. The packet filtering policy is described as a set of access lists.

The format of the access list is as follows:

`access-list` `permit|deny` protocol {source-address source-mask|`any`}
{destination-address destination-mask|`any`} {`eq` port-number|`any`}

Example 1:

```
access-list permit tcp 10.1.1.64 0.0.0.255 10.1.2.128 0.0.0.255 any
```

This access-list permits TCP traffic from IP addresses 10.1.1.*x* to IP addresses 10.1.2.*y* for any port-number. Here, *x* is any value from 0 to 255 because the source-mask exhibits the value 0.0.0.255, indicating that "don't care" the right-most 8 bits of the source-address. Similarly, *y* is also any value from 0 to 255.

Example 2:

```
access-list deny tcp any any eq 23
```

This access-list denies TCP traffic from any IP address to any IP address for port-number 23.

Examples of port numbers are: 23 (telnet), 25 (smtp), 80 (http), 110 (pop3).

The network administrator creates the following two access lists:

```
access-list permit tcp [ D ] 10.1.1.29 0.0.0.0 eq [ E ]
access-list deny tcp any 10.1.1.29 0.0.0.0 [ E ]
```

When a packet arrives at CR, the access lists are tested from the top. If the source, destination, and port-number of the packet match those in the first access list, the action (permit or deny) on the first access list is applied. Otherwise, the second access list is tested.

The first access list permits Web accesses from the hosts H4 in the development department to the Web services on the development server. The second access list denies Web accesses from any hosts to the Web services on the development server.

These access lists are to be applied at [F] interface (outbound direction).

Answer group for D

a) `10.1.1.0 0.0.0.0` b) `10.1.1.0 0.0.0.255` c) `10.1.2.0 0.0.0.127`
d) `10.1.2.0 0.0.0.255` e) `any`

Answer group for E

a) 23 b) 25 c) 80 d) 110

Answer group for F

a) e0 b) e1 c) s0 d) s1

Q5. Read the following description of a bus ticket reservation system, and then answer Subquestions 1 and 2.

Company W is a bus company that operates long-distance bus services. It develops a bus ticket reservation system. This system uses the Bus file, Booking file, and Seat file.
When a user wishes to book bus seats, he/she first selects the bus name, route name, departure date, and departure time on the screen. Then, the system displays detailed information of the selected bus with a seat map that shows the booking status of individual seats.

There are two types of booking processes: “buy” and “hold”.
If the user decides to “buy” the bus tickets, the system receives 100% of the fares, issues the bus tickets, and sets the relevant seat status to “sold”.
If the user wishes to “hold” the seats at the moment, the system receives 10% of the fares, and sets the relevant seat status to “held”. The user can specify “hold” by 3 days before the departure date. When the user decides to purchase the bus tickets for the held seats until 3 days before the departure date, the system receives the remaining 90% of the fares, issues the bus tickets, and alters the relevant seat status to “sold”. When the user wishes to cancel the held seats by 3 days before the departure date, the system returns the received 10% of the fares. When the user takes no action for the held seats by 3 days before the departure date, the system does not return the received 10% of the fares, and alters the relevant seat status of Booking file to “canceled” and the relevant seat status of Seat file to “available”.

Figures 1, 2 and 3 show the record format of Bus file, Booking file and Seat file respectively, with sample data. The underlined fields are primary keys.

<u>BusNo</u>	BusName	RouteName	DepartureDate	DepartureTime	SeatingCapacity	Fare
1021118	YM18	City Y → City M	2017-10-21	18:00	50	5,000
1021119	YM19	City Y → City M	2017-10-21	19:00	50	5,000
1021121	YM21	City Y → City M	2017-10-21	21:00	50	6,000
1021318	YB18	City Y → City B	2017-10-21	18:00	30	3,000

Figure 1 Record format of Bus file

<u>BookingID</u>	BusNo	UserName	PhoneNo	NumberOfSeats	TotalFare	Paid	SeatStatus
101402	1021118	Mr. A	09-1234567	2	10,000	10,000	sold
101403	1021118	Ms. B	09-2345678	3	15,000	1,500	held

Note: SeatStatus holds one of the three values: “canceled”, “held”, or “sold”.

Figure 2 Record format of Booking file

BusNo	SeatID	BookingID	SeatStatus
1021118	1A	101403	held
1021118	1B	101403	held
1021118	1C	101403	held
1021118	1D	(null)	available
1021118	2A	101402	sold
1021118	2B	101402	sold
…	…	…	…

Note: SeatStatus holds one of the three values: "available", "held", or "sold".

Figure 3 Record format of Seat file

In Program 1 in Subquestion 1 and Program 2 in Subquestion 2, a `READ` statement

- `READ` *filename* `file` `WHERE` *condition*

reads a record that satisfies the *condition* from the *filename* file.

If the record is read successfully, the host variable `&ReadStatus` is set to "success", and each field of the record can be referred to by specifying *filename*`.`*fieldname*, such as `Seat.BusNo`. If there are multiple records that satisfy the *condition*, the records can be read in one by one by executing the `READ` statement repeatedly.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in Program 1.

Program 1 is a small-size module that performs a part of the cancellation process. It is called when a user requests cancellation of held seats.

[Program 1]

```
○ String type: WebInput
• DISPLAY "Enter [   A   ]:"
• GET the value entered from keyboard into WebInput
• READ Booking file WHERE Booking.[   A   ] = WebInput
▲ (&ReadStatus = "success") AND (Booking.SeatStatus = "held")
│ • READ Seat file WHERE Seat.BookingID = Booking.BookingID
│ ■ &ReadStatus = "success"     /* the next record exists */
│ │ • Seat.BookingID ← "(null)"
│ │ • [   B   ]
│ │ • WRITE the updated record into Seat file
│ │ • READ Seat file WHERE Seat.BookingID = Booking.BookingID
│ ■
│ • [   C   ]
│ • WRITE the updated record into Booking file
▼
```

Answer group for A

a) `BookingID`
b) `BusNo`
c) `SeatStatus`
d) `UserName`

Answer group for B and C

a) `Booking.SeatStatus ← "canceled"`
b) `Booking.SeatStatus ← Seat.SeatStatus`
c) `Booking.UserName ← "(null)"`
d) `Seat.SeatID ← "(null)"`
e) `Seat.SeatStatus ← "available"`
f) `Seat.SeatStatus ← Booking.SeatStatus`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in Program 2.

Program 2 is a small-size module that also performs a part of the cancellation process. It starts at 1:00 daily as a batch job subsequent to the file backup operation.
Here, the date of the day after tomorrow is provided by the host variable `&TodayPlus2`.

[Program 2]

```
• READ Bus file WHERE [   D   ]
■ &ReadStatus = "success"
│ • READ Booking file WHERE [   E   ] AND [   F   ]
│ ■ &ReadStatus = "success"
│ │ • READ Seat file WHERE Seat.BookingID = Booking.BookingID
│ │ ■ &ReadStatus = "success"
│ │ │ • Seat.BookingID ← "(null)"
│ │ │ • Seat.SeatStatus ← "available"
│ │ │ • WRITE the updated record into Seat file
│ │ │ • READ Seat file WHERE Seat.BookingID = Booking.BookingID
│ │ ■
│ │ • Booking.SeatStatus ← "canceled"
│ │ • WRITE the updated record into Booking file
│ │ • READ Booking file WHERE [   E   ] AND [   F   ]
│ ■
│ • READ Bus file WHERE [   D   ]
■
```

Answer group for D through F

a) `Booking.BookingID = Seat.BookingID`

b) `Booking.BusNo = Bus.BusNo`

c) `Booking.BusNo = Seat.BusNo`

d) `Booking.SeatStatus = "held"`

e) `Booking.SeatStatus = Seat.SeatStatus`

f) `Bus.DepartureDate ≤ &TodayPlus2`

g) `Bus.DepartureDate = &TodayPlus2`

Q6. Read the following description of programs and the programs themselves, and then answer Subquestions 1 through 3.

[Program Description]

(1) For enhancing the environment to attract higher number of tourists, it is decided to carry out waste collection along the coast. The entire coastline of City A resort is divided into `N` portions from 1 to `N`. Analysis reveals that in the `I`-th portion, there are `X[I]` tons of garbage; `I` is in the range from 1 to `N`.

(2) There is a garbage truck for use in trial operation. During a trip, the truck can collect and process a maximum of `T` tons of garbage. Because it is a trial run, the engineers are highly cautious, wishing to select a certain shoreline that includes sequential portions of coastline for convenient tracking and evaluation.

(3) The algorithm determines the number of feasible trips that can be selected for the truck to clean the coastline, considering that each trip covers sequential portions and that the total garbage in these portions is less than or equal to the truck capacity.
Program 1, named `TrashGen`, calculates and outputs the number of feasible trips. The argument specification for the program `TrashGen` is given in Table 1.

Table 1 Argument specification for the program `TrashGen`

Variable	Input/Output	Description
`N`	Input	The number of coastline portions.
`T`	Input	The maximum quantity of garbage (tons) that the collector can process. Here, `T` is a positive integer.
`X[]`	Input	An array of `N` elements where `X[I]` is the quantity of garbage (tons) at portion `I`.
`Seg`	Output	The number of feasible trips.

(4) For example, consider a sample case of waste collection shown in Figure 1.

`N`	9
`T`	10

	1	2	3	4	5	6	7	8	9
`X[]`	11	1	2	1	1	5	10	2	3

Figure 1 Sample case of waste collection

In the case shown in Figure 1, there are 19 feasible trips for the truck to clean the coastline, namely, {2}, {2,3}, {2,3,4}, {2,3,4,5}, {2,3,4,5,6}, {3}, {3,4}, {3,4,5}, {3,4,5,6}, {4}, {4,5}, {4,5,6}, {5}, {5,6}, {6}, {7}, {8}, {8, 9}, and {9}. Therefore, the program `TrashGen` returns 19.

[Program 1]

```
○ Program: TrashGen(Integer type: N,
                    Integer type: T,
                    Integer type: X[N])
  ○ Integer type: I, J, Seg, Sum

  • Seg ← 0
  • Sum ← 0
  ■ I: 1, [  A  ], 1
  │ • Sum ← 0
  │ ■ J: I, [  B  ], 1
  │ │ • Sum ← Sum + X[J]
  │ │ ▲ Sum ≤ T
  │ │ │ • [  C  ]
  │ │ ▼
  │ ■
  ■
  • Return Seg
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the above program.

Answer group for A

- a) `I < N`
- b) `I ≤ N`
- c) `I < N and Sum < T`
- d) `I ≤ N and Sum < T`
- e) `I < N and Sum ≤ T`
- f) `I ≤ N and Sum ≤ T`

Answer group for B

- a) `J < N and Sum ≤ T`
- b) `J ≤ N and Sum ≤ T`
- c) `J < N and Sum ≤ X[I]`
- d) `J ≤ N and Sum ≤ X[I]`
- e) `J < N or Sum ≤ T`
- f) `J ≤ N or Sum ≤ T`

Answer group for C

- a) `Seg ← Seg + 1`
- b) `Seg ← Seg + J - I`
- c) `Seg ← Seg + J - I - 1`
- d) `Seg ← Seg + J - I + 1`
- e) `Seg ← Seg + X[I]`
- f) `Seg ← Seg + X[J]`

Subquestion 2

Program 2, named `TrashGen2`, is implemented by modifying the program `TrashGen` to output certain information.

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description. Here, assume that the correct answers are inserted in blanks $\boxed{\text{A}}$ through $\boxed{\text{C}}$ in Program 2.

When the program `TrashGen2` is executed for the case shown in Figure 1, the following statement will be outputted. Here, the shaded part is not shown intentionally.

```
Sum of garbage from portion [    ] to portion [ D ] is [ E ].
```

[Program 2]

```
○ Program: TrashGen2(Integer type: N,
                     Integer type: T,
                     Integer type: X[N])
  ○ Integer type: I, J, M, From, To, Sum, Seg

  • M ← 0
  • Seg ← 0
  • From ← 0
  • To ← 0
  ■ I: 1, [ A ], 1
  │ • Sum ← 0
  │ ■ J: I, [ B ], 1
  │ │ • Sum ← Sum + X[J]                     ← α
  │ │ ▲ Sum ≤ T
  │ │ │ ▲ Sum > M
  │ │ │ │ • M ← Sum
  │ │ │ │ • From ← I
  │ │ │ │ • To ← J
  │ │ │ ▼
  │ │ │ • [ C ]
  │ │ ▼
  │ ■
  ■
  /* Output the statement. Here, "+" means the string concatenation operator. */
  • Print ("Sum of garbage from portion " + From +
          " to portion " + To + " is " + M + ".")
  • Return Seg
```

Answer group for D and E

a)	0	b)	1	c)	2	d)	3	e)	4
f)	6	g)	7	h)	8	i)	9	j)	10

Subquestion 3

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following description.

Until the program `TrashGen2` is terminated, the statement surrounded by the dashed rectangle that is pointed by "← α" will be executed $\boxed{\text{F}}$ times.

Answer group for F

a)	18	b)	19	c)	24	d)	26	e)	28
f)	39	g)	40	h)	52	i)	60	j)	61

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the (S) in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestions 1 and 2.

The body mass index (BMI) is a measurement that categorizes a person as underweight, normal, overweight, or obese based on their weight and height (See Table 1). The BMI is defined as the body weight (kg) divided by the square of the body height (m).

Table 1 The BMI category

Category	BMI	
	from	to
Underweight	-	18.5
Normal	18.5	25
Overweight	25	30
Obese	30	-

[Program Description]

(1) The program reads a series of physical data from the standard input. The input for the program is provided in the following order. The first line contains the number of persons `n` (`n` is 30 or fewer). Each of the next `n` lines contain a string and two floating point numbers, indicating the person name, his/her weight in kilograms, and height in meters, respectively. The person name does not exceed in 9 characters, and no space character is inserted. The values are separated by one or more space characters. Figure 1 shows an example of input data.

```
10
Alice 78.95 1.54
Chuck 93.06 1.78
Bob 47.20 1.64
Heidi 83.19 1.80
David 72.30 1.91
Mallory 70.58 1.53
Eve 76.53 1.67
Frank 65.17 1.72
Oscar 49.91 1.45
Grace 37.15 1.60
```

Figure 1 Example of input data

(2) The program prints `n` lines as output. Each line contains three values: the name, BMI value, and category of the person. The output is sorted by category (from underweight to obese). Within each category, the output is sorted in ascending order of the ASCII codes of the name. Figure 2 shows the output for the example input data shown in Figure 1.

```
Bob 17.55 Underweight
Grace 14.51 Underweight
David 19.82 Normal
Frank 22.03 Normal
Oscar 23.74 Normal
Chuck 29.37 Overweight
Eve 27.44 Overweight
Heidi 25.68 Overweight
Alice 33.29 Obese
Mallory 30.15 Obese
```

Figure 2 Output for the input data shown in Figure 1

(3) The following four user-defined functions are used in the program:

(i) `float calcBMI(float weight, float height)`
This function calculates the BMI value.

(ii) `enum BMIcategory calcCategory(float bmi)`
This function returns the category to `bmi`.

(iii) `void sortbyBMI(struct person p[], int from, int to)`
This function sorts between the `from`-th and `to`-th element of `p[]` in ascending order of `bmi`.

(iv) `void sortbyName(struct person p[], int from, int to)`
This function sorts between the `from`-th and `to`-th element of `p[]` in ASCII code order of `name`.

(4) The following library function is used in the program:

(i) `int strcmp(const char* s1, const char* s2)`
This function compares the string pointed by `s1` and the string pointed by `s2`. When `s1` < `s2`, it returns a negative value; when `s1` = `s2`, it returns 0; when `s1` > `s2`, it returns a positive value.

[Program]

```
#include <stdio.h>
#include <string.h>

#define NAME_LEN 10
#define MAX_NUM_PERSON 30
#define NUM_CATEGORY 4

enum BMIcategory {
   underweight, normal, overweight, obese
};

struct person {
   char name[NAME_LEN];
   float bmi;
   enum BMIcategory category;
};

float calcBMI(float, float);
enum BMIcategory calcCategory(float);
void sortbyBMI(struct person p[], int from, int to);
void sortbyName(struct person p[], int from, int to);

char* BMICategoryName[] = {"Underweight", "Normal",
                           "Overweight", "Obese"};

int main() {
   int i, n;
   struct person p[MAX_NUM_PERSON];
   int numPerson[NUM_CATEGORY];
   float weight, height;
   int from = 0;

   for (i = 0; i < NUM_CATEGORY; i++) {
      numPerson[i] = 0;
   }
   scanf("%d", &n);
   for (i = 0; i < n; i++) {
      scanf("%s%f%f", [     A     ], &weight, &height);
      p[i].bmi = calcBMI([     B     ]);
      p[i].category = calcCategory([     C     ]);
      numPerson[p[i].category]++;
   }
```

```
    [     D     ];
    for (i = 0; i < NUM_CATEGORY; i++) {
        [     E     ];
        from += numPerson[i];
    }

    for (i = 0; i < n; i++) {
        printf("%s %3.2f %s\n",
               p[i].name, p[i].bmi, BMICategoryName[p[i].category]);
    }
}

float calcBMI(float weight, float height) {
    return weight / (height * height);
}

enum BMIcategory calcCategory(float bmi) {
    if (bmi < 18.5) {
        return underweight;
    } else if (bmi < 25) {
        return normal;
    } else if (bmi < 30) {
        return overweight;
    } else {
        return obese;
    }
}

void sortbyBMI(struct person p[], int from, int to) {
    struct person tmp;
    int i, j;

    for (i = from; i <= to - 1; i++) {
        for (int j = i + 1; j <= to; j++) {
            if (p[i].bmi > p[j].bmi) {
                tmp = p[i];
                p[i] = p[j];
                p[j] = tmp;
            }
        }
    }
}
```

```
void sortbyName(struct person p[], int from, int to) {
   struct person tmp;
   int i, j;

   for (i = from; i <= to - 1; i++) {
      for (j = i + 1; j <= to; j++) {
         if (strcmp(p[i].name, p[j].name) > 0) {    //  ← α
            tmp = p[i];                              //  ← β
            p[i] = p[j];
            p[j] = tmp;
         }
      }
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A

a) `&p[i].bmi`
b) `category`
c) `name`
d) `p[i].bmi`
e) `p[i].category`
f) `p[i].name`

Answer group for B

a) `&p[i].weight, &p[i].height`
b) `&weight, &height`
c) `p[i].weight, p[i].height`
d) `weight, height`

Answer group for C

a) `bmi`
b) `numPerson[i]`
c) `p[i]`
d) `p[i].bmi`

Answer group for D

a) `sortbyBMI(p, 0, n)`
b) `sortbyBMI(p, 0, n - 1)`
c) `sortbyBMI(p, 0, numPerson[0])`
d) `sortbyName(p, 0, n)`
e) `sortbyName(p, 0, n - 1)`
f) `sortbyName(p, 0, numPerson[0])`

Answer group for E

a) `sortbyBMI(p, from, from + numPerson[i])`
b) `sortbyBMI(p, from, from + numPerson[i] - 1)`
c) `sortbyBMI(p, from, numPerson[i])`
d) `sortbyBMI(p, from, numPerson[i] - 1)`
e) `sortbyName(p, from, from + numPerson[i])`
f) `sortbyName(p, from, from + numPerson[i] - 1)`
g) `sortbyName(p, from, numPerson[i])`
h) `sortbyName(p, from, numPerson[i] - 1)`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank [ ] in the following description.

The program is executed with the input data shown in Figure 1.
Until the program is terminated, the statement marked by "$\leftarrow \alpha$" is executed [F] times, and the statement marked by "$\leftarrow \beta$" is executed [G] time(s).

Answer group for F

a) 4	b) 8	c) 18
d) 45	e) 55	f) 81

Answer group for G

a) 1	b) 3	c) 5
d) 7	e) 9	f) 11

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestions 1 and 2.

[Program Description]

(1) The `SpecialNumbers` class represents a set of unique integers. Its capacity can be specified at the time of creation and cannot be altered subsequently. It maintains the actual length of the set (i.e., the number of the unique integers).
It incorporates a method referred to as `conditionalAdd` that adds the specified new element to the set.
  (i) If the new element is unique in the set, the new element is added and the length of the set is increased.
  (ii) The sequence of the added numbers is internally maintained.
  (iii) It throws an `IllegalStateException` if the new element to be added is already present or the set has already attained its maximum capacity.

(2) The `getLength` method tells how many elements are present.

(3) The `getCapacity` method tells how many elements in total can be stored.

(4) The `nth` method returns the `n`-th value after arranging all the elements in ascending (increasing) order.
  (i) If `n` = 1, it returns the lowest number.
  If `n` = 2, it returns the second lowest number.
  If `n` = 3, it returns the third lowest number, etc.
  If `n` = `length`, it returns the highest number.
  (ii) The method corrects any out-of-boundary values of `n` to be as the boundary values. That is, if `n` < 1, it returns the lowest number, and if `n` > `length`, it returns the highest number.

(5) The `median` method calculates and returns the median value of the elements.
  (i) If the total number of elements is odd (e.g., five elements), the median is the middle element value after sorting. For example, if there are five elements as shown below, the median is 30, that is the middle element value of the sorted elements.

Elements (original order)	20	40	10	50	30
Elements (sorted order)	10	20	30	40	50

  (ii) If the total number of elements is even (e.g., six elements), the median is the average of the two middle elements after sorting. For example, if there are six elements as shown below, the median is 35, that is (30 + 40) / 2.

Elements (original order)	20	60	10	40	30	50
Elements (sorted order)	10	20	30	40	50	60

(6) The `getNumbers` method returns a new `int` array containing only the added elements, preserving the sequence.
(7) The `toString` method returns a `String` representing the added elements, preserving the sequence.

The following output is produced when the `Tester` class is executed with input values of 20, 40, 10, 50, and 30.

```
numbers: [20, 40, 10, 50, 30]
median: 30.0
nth(1): 10
nth(2): 20
nth(3): 30
nth(4): 40
nth(5): 50
nth(-5): 10
nth(100): 50
```

[Program 1]

```
import java.util.Arrays;
import java.util.NoSuchElementException;

public class SpecialNumbers {
   private final int[] numbers;
   private int length = 0;

   public SpecialNumbers(int capacity) {
      numbers = new int[capacity];
   }

   public void conditionalAdd(int newElement) {
      for (int i = 0; i < length; i++) {
         if (numbers[i] == newElement) {
            throw new IllegalStateException(newElement
                                          + " already exists");
         }
      }
      if (length == [    A    ]) {
         throw new IllegalStateException("Capacity is full");
      }
      // Successful entry of unique number
      numbers[[    B    ]] = newElement;
   }
```

```
public int getCapacity() {
   return [    A    ];
}

public double median() {
   if (isEmpty()) {
      throw new NoSuchElementException("no numbers");
   }
   // Create a copy with mentioned quantity of elements from beginning
   int[] temp = Arrays.copyOf(numbers, length);
   Arrays.sort(temp);      // Sorts the array in ascending order
   if (temp.length % 2 == 0) {
      return (temp[    C    ] + temp[    D    ]) / 2.0;
   } else {
      return temp[    C    ];
   }
}

public int nth(int n) {
   if (isEmpty()) {
      throw new NoSuchElementException("no numbers");
   }
   if (n > length) {
      n = length;
   } else if (n < 1) {
      n = 1;
   }
   // Create a copy with mentioned quantity of elements from beginning
   int[] temp = Arrays.copyOf(numbers, length);
   // Partial sort up to n-th element
   for (int i = 1; i <= n; ++i) {
      int minLocation = i - 1;
      for (int j = minLocation + 1; j < temp.length; ++j) {
         if (temp[j] < temp[minLocation]) {
            minLocation = [    E    ];
         }
      }
      if (i - 1 != minLocation) {
         int backup = temp[i - 1];
         temp[i - 1] = temp[minLocation];
         temp[minLocation] = backup;
      }
   }
   return temp[n - 1];
}
```

```
    public int[] getNumbers() {
        return [    F    ];
    }

    public int getLength() {
        return length;
    }

    private boolean isEmpty() {
        return getLength() == 0;
    }

    public String toString() {
        return Arrays.toString(getNumbers());
    }
}
```

[Program 2]

```
public class Tester {
    private static final int[] NUMBERS = { 20, 40, 10, 50, 30 };

    public static void main(String[] args) {
        SpecialNumbers sn = new SpecialNumbers(5);
        for (int e : NUMBERS) {
            sn.conditionalAdd(e);
        }
        System.out.println("numbers: " + sn);
        System.out.println("median: " + sn.median());
        for (int i = 1; i <= sn.getLength(); ++i) {
            System.out.println("nth(" + i + "): " + sn.nth(i));
        }
        System.out.println("nth(-5): " + sn.nth(-5));
        System.out.println("nth(100): " + sn.nth(100));
    }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank ☐ in Program 1.

Answer group for A and B

a) `capacity` b) `capacity + 1` c) `capacity++`
d) `length` e) `length + 1` f) `length - 1`
g) `length++` h) `numbers.length`

Answer group for C and D

a) `(temp.length + 1) / 2` b) `temp.length / 2`
c) `temp.length / 2 + 1` d) `temp.length / 2 - 1`

Answer group for E

a) `i` b) `i + 1` c) `i - 1`
d) `j` e) `j + 1` f) `j - 1`

Answer group for F

a) `Arrays.copyOf(numbers, length)`
b) `Arrays.copyOf(numbers, numbers.length)`
c) `new int[numbers.length]`
d) `new int[numbers]`
e) `numbers`
f) `numbers.clone()`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank ☐ in the following description.

In order to support a subsequence of the numbers, the following new constructor and method are added to the `SpecialNumbers` class.

(1) The new constructor creates a `SpecialNumbers` with the specified `int` array as `numbers`.
(2) The `subNumbers` method creates a `SpecialNumbers` containing a subsequence of this `SpecialNumbers`. Arguments `start` and `end` specify the first and last characters, respectively, of the subsequence. If `start` and/or `end` are out-of-range values, the

values are adjusted following the `nth` method convention. An `IllegalArgumentException` is thrown if `start` is greater than `end` after the adjustment.

The following is the source code of the additional constructor and method.

```
private SpecialNumbers(int[] numbers) {
   this.numbers = numbers;
}

public SpecialNumbers subNumbers(int start, int end) {
   if (start < 1) {
      start = 1;
   }
   if (end > getLength()) {
      end = getLength();
   }
   if (start > end) {
      throw new IllegalArgumentException();
   }
   int[] subarray = Arrays.copyOfRange(numbers, start - 1, end);
   SpecialNumbers subNumbers = new SpecialNumbers(subarray);
   subNumbers.length = subarray.length;
   return subNumbers;
}
```

In order to test the subsequence functionality, the following lines are added to the end of the `main` method in the `Tester` class.

```
   SpecialNumbers sub = sn.subNumbers(2, 5);
   System.out.println("sub: " + sub.median());
   SpecialNumbers subsub = sub.subNumbers(2, 4);
   System.out.println("subsub: " + subsub.median());
```

The following output is produced by the additional lines listed above.

```
sub: [    G    ]
subsub: [    H    ]
```

Answer group for G and H

a) `20.0` b) `25.0` c) `30.0`
d) `35.0` e) `40.0` f) `45.0`

24th ITPEC FE Afternoon Exam -- October 2017

Q	SQ	BQ	Correct Answer
1	1	A	d
	2	B	g
		C	d
		D	c
	3	E	a
2		A	c
		B	a
		C	b
		D	c
		E	c
3	1	A	b
		B	d
	2	C	a
		D	b
	3		d
4	1	A	a
	2	B	c
		C	c
	3	D	c
		E	c
		F	b
5	1	A	a
		B	e
		C	a
	2	D	g
		E	b/d
		F	d/b
6	1	A	b
		B	b
		C	a
	2	D	f
		E	j
	3	F	d
7	1	A	f
		B	d
		C	d
		D	b
		E	f
	2	F	b
		G	c
8	1	A	h
		B	g
		C	b
		D	d
		E	d
		F	a
	2	G	d
		H	c

[Note] Answers for E and F can be exchanged.

April 2017

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the spring Fundamental IT Engineer Examination conducted?

   Answer group
   a) March b) April c) May d) June

   Since the correct answer is " b) April ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

<table>
<tr><th colspan="2">Notation</th><th>Description</th></tr>
<tr><td colspan="2">○</td><td>Declares names, types, etc., of procedures, variables, etc.</td></tr>
<tr><td colspan="2">/* text */</td><td>Describes comments in the text.</td></tr>
<tr><td rowspan="7">Process</td><td>• variable ← expression</td><td>Assigns the value of the expression to the variable.</td></tr>
<tr><td>• procedure(argument, ...)</td><td>Calls the procedure and passes / receives the argument.</td></tr>
<tr><td>▲ conditional expression
process
▼</td><td>Indicates a one-way selection process.
If the conditional expression is true,
then the process is executed.</td></tr>
<tr><td>▲ conditional expression
process 1
―――
process 2
▼</td><td>Indicates a two-way selection process.
If the conditional expression is true,
then process 1 is executed.
If it is false, then process 2 is executed.</td></tr>
<tr><td>■ conditional expression
process
■</td><td>Indicates a pre-test iteration process.
While the conditional expression is true,
the process is executed repeatedly.</td></tr>
<tr><td>■
process
■ conditional expression</td><td>Indicates a post-test iteration process.
The process is executed, and then
while the conditional expression is true,
the process is executed repeatedly.</td></tr>
<tr><td>■ variable: init, cond, incr
process
■</td><td>Indicates an iteration process.
The initial value init (given by an expression) is stored in the variable at the start of the iteration process, and then while the conditional expression cond is true, the process is executed repeatedly.
The increment incr (given by an expression) is added to the variable in each iteration.</td></tr>
</table>

[Logical constants]

`true, false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	`+, −, not`	High
Multiplication, division	`×, ÷, %`	↑
Addition, subtraction	`+, −`	
Relational operation	`>, <, ≥, ≤, =, ≠`	
Logical product	`and`	↓
Logical sum	`or`	Low

Note: With division of integers, an integer quotient is returned as a result.
The "%" operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description of an exchange of cryptographic keys, and then answer Subquestions 1 and 2.

As a method of exchange of cryptographic keys in an insecure network, the Diffie-Hellman algorithm allows two users to securely exchange a key that can be used for subsequent symmetric encryption of messages.

To establish a shared secret key, the two users must first select two numbers: a prime number p and an integer g that is a primitive root of p. The number g is a primitive root of p when for each n (n = 1, 2, …, p – 1), and each value of n (n = 1, 2, …, p – 1) appears in g^n mod p. Table 1 shows an example of primitive root when p = 11 and g = 2. The bottom row shows the numbers 1, 2, …, 10.

Table 1 Example of primitive root when p = 11 and g = 2

n	1	2	3	4	5	6	7	8	9	10
2^n	2	4	8	16	32	64	128	256	512	1024
2^n mod 11	2	4	8	5	10	9	7	3	6	1

The Diffie-Hellman algorithm is shown in Figure 1 and in the following steps:

(1) Alice and Bob select a prime number p and an integer g that is a primitive root of p.

(2) Alice selects a random number x ($1 \le x \le p - 1$), and calculates R1 = g^x mod p.

(3) Bob selects another random number y ($1 \le y \le p - 1$), and calculates R2 = g^y mod p.

(4) Alice sends R1 to Bob. Note that Alice does not send the value of x.

(5) Bob sends R2 to Alice. Note that Bob does not send the value of y.

(6) Alice calculates the shared secret key K = $(\text{R2})^x$ mod p.

(7) Bob calculates the shared secret key K = $(\text{R1})^y$ mod p.

Thus, Alice and Bob obtain the shared secret key K for the session.

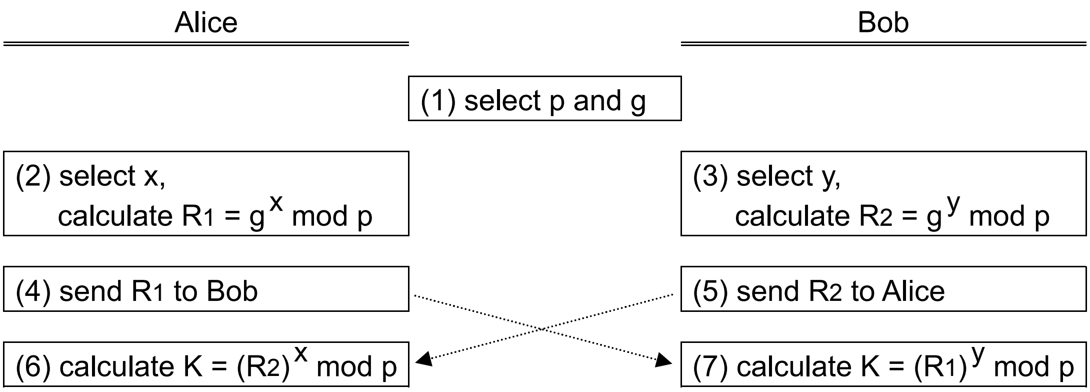


Figure 1 Diffie-Hellman key exchange algorithm

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

A shared secret key is being exchanged between Alice and Bob using this algorithm. Assume that they agreed upon the prime number $p = 11$ and the integer $g = 2$.
If Bob has received the public key R1 = 3 from Alice, the random number x Alice has selected is $\boxed{\text{A}}$. Also, if Alice has received the public key R2 = 9 from Bob, the random number y Bob has selected is $\boxed{\text{B}}$.
In this case, i.e. R1 = 3 and R2 = 9, the shared secret key obtained by both Alice and Bob is $\boxed{\text{C}}$, which they can use for subsequent symmetric encryption of messages.

Answer group

a) 2 b) 3 c) 4 d) 5
e) 6 f) 7 g) 8 h) 9

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

Alice and Bob decided to change the values of g, x, and y. Then, they exchanged the re-calculated R1 and R2, and finally obtained the shared secret key K.
Assume that an attacker Eve knows the three non-secret values; $p = 11$, $g = 7$, and R1 = 3.
Recently, Eve obtains the value $y = 3$ by illegal means. Then, it would be possible for Eve to determine the shared secret key K.
Eve finds out that the shared secret key K is $\boxed{\text{D}}$. Furthermore, by referring to Table 2, Eve finds out that the value of x is $\boxed{\text{E}}$, and the value of R2 is $\boxed{\text{F}}$.

Table 2 Table of 7^n mod 11

n	1	2	3	4	5	6	7	8	9	10
7^n mod 11	7	5	2	3	10	4	6	9	8	1

Answer group

a) 2 b) 3 c) 4 d) 5
e) 6 f) 7 g) 8 h) 9

Q2. Read the following description of a door controller circuit, and then answer Subquestions 1 and 2.

A sensor-based automatic door controller has three states of operation: “closed”, “opened” and “closing”. When the sensor detects a person coming towards the door or going through the door, the door enters the “opened” state. When the person has passed through the door, the sensor detects nothing and the door enters the “closing” state. Within the “closing” state, another person may approach towards the door, causing the change in the state of the door from “closing” to “opened”. The idle state “closed” implies that the door is completely closed.

Variable *S* represents the signal generated by the sensor, which is used as an input for the door controller circuit. The sensor generates the signal S = 1 when the sensor detects a person either approaching to or passing through the door; otherwise, it generates the signal S = 0. Figure 1 shows the operational state transition diagram of the door controller.

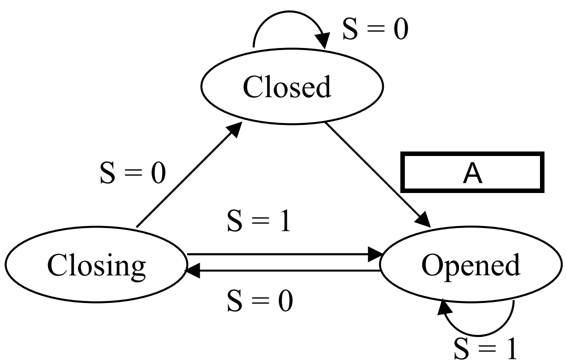


Figure 1 State transition diagram of the door controller

In designing the door controller circuit, 2-digit binary numbers 00, 01, and 10 are assigned to the states “closed”, “opened”, and “closing”, respectively. The higher bit is denoted by Q_H and the lower bit by Q_L. For example, the state “closing” (10) is represented as $Q_H = 1$ and $Q_L = 0$. With this notation, Table 1 is equivalent to Figure 1.

Table 1 State transition table of the door controller

Current state		Signal	Next state	
Q_H	Q_L	S	Q_H	Q_L
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	B	

A hardware engineer creates the program `SetNextState` that determines the next states of Q_H and Q_L based on Table 1. The program is executed at regular intervals triggered by the clock-generated pulses. Assume that during the program execution, the contents of `QH`, `QL`, and `S` are not changed nor fetched by external processes.

```
○ Global: Bit type: QH, QL, S
○ Program: SetNextState
  • QH ← [      C      ]
  • QL ← S
  • return
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the above description.

Answer group for A

a) S = 0 b) S = 1

Answer group for B

a) [0 | 0] b) [0 | 1] c) [1 | 0] d) [1 | 1]

Answer group for C

a) `((not QH) or QL) and (not S)`
b) `(QH and S) or (QL and (not S))`
c) `(not QL) and S`
d) `QL and (not S)`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank [ ] in the following description.

The door controller system consists of three components as shown in Figure 2.

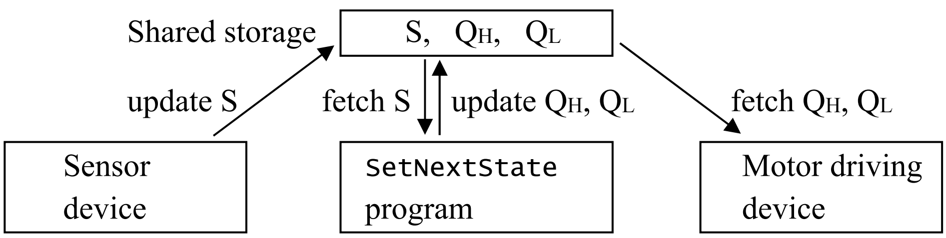


Figure 2 Door controller system

The door controller system operates normally on condition that the execution of the three components is well controlled. The description below explains a problem that may occur when the program `SetNextState` and the motor driving device operate independently.

The program `SetNextState` first updates Q_H and then updates Q_L. Therefore, there is a very short time lag between the updates of Q_H and Q_L. During this time lag, if the motor driving system fetches the values of Q_H and Q_L, it may result to a problem. As an example, the motor driving system receives an undefined state 11 (Q_H = 1 and Q_L = 1) when the program updates the state of the door [D].
By [E], for example, the time lag problem mentioned above can be resolved.

Answer group for D

a) from "closed" to "opened"
b) from "closing" to "closed"
c) from "closing" to "opened"
d) from "opened" to "closing"

Answer group for E

a) changing the data format of Q_H and Q_L so that they can be fetched or updated by one hardware instruction
b) changing the execution logic of the motor driving device so that it will ignore the undefined state 11
c) changing the execution sequence of the program `SetNextState` so that the program first updates Q_L and then updates Q_H
d) replacing the shared storage device with a new one so that Q_H and Q_L can be fetched or updated much faster than the current device

Q3. Read the following description of resource access control, and then answer Subquestion.

Resource access control is an important task that must be provided by operating systems. When designing a mechanism of process execution control that needs resource access control, the "semaphore" concept is useful.

Semaphore is a special variable that holds the number of copies of the specific resource that is currently available. Working on the specific semaphore named `S`, two functions `wait(S)` and `signal(S)` are provided. A process that needs the resource executes `wait(S)`, and a process that supplies the resource executes `signal(S)`.

(1) Function `wait(S)` performs the following operations:
    Decrement the value of `S` by 1.
    If `S` < 0 after the decrement, block the process that is executing `wait(S)`.

(2) Function `signal(S)` performs the following operations:
    Increment the value of `S` by 1.
    If `S` ≤ 0 after the increment, unblock one process that has been blocked by `wait(S)`.

Figures 1 through 3 show how semaphore `S` works upon three processes P1, P2 and R. The system shown in the Figures consists of a processor, ready queue, wait queue, and semaphore `S`. P1 and P2 need the resource and R supplies the resource. When P1 or P2 executes `wait(S)`, the process is interrupted and placed into the wait queue or ready queue depending on the value of `S`. When the process is loaded again for execution, the rest of the process is executed without executing `wait(S)`, which is deleted from the system. R executes `signal(S)` at the end of its process, and then R is deleted from the system.

Figure 1 shows the current state where P2 is running. R and P1 are in the ready queue, and no process is in the wait queue. The resource supplied by R is not available because `S` = 0.

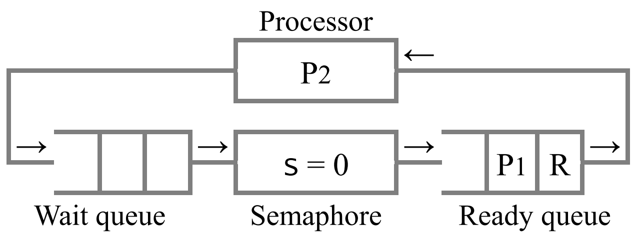


Figure 1 P2 is running

After P2 executes `wait(S)` in Figure 1, `S` is changed to `-1`, and P2 is blocked because `S` < 0. Subsequently, P2 is placed into the wait queue, and R is picked up from the ready queue for execution (Figure 2).

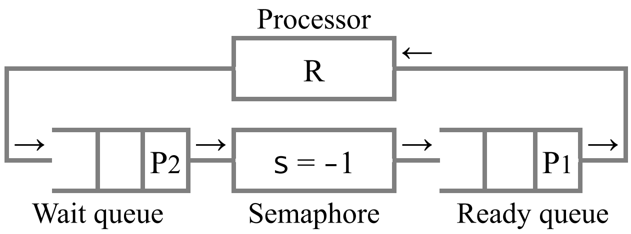


Figure 2 After P2 executes `wait(S)` in Figure 1

After R executes `signal(S)` in Figure 2, `S` is changed to 0, and R is ended and deleted. P2, which has been blocked by `wait(S)`, is unblocked because $S \leq 0$. Subsequently, P2 is moved to the ready queue, and then P1 is picked up from the ready queue for execution (Figure 3).

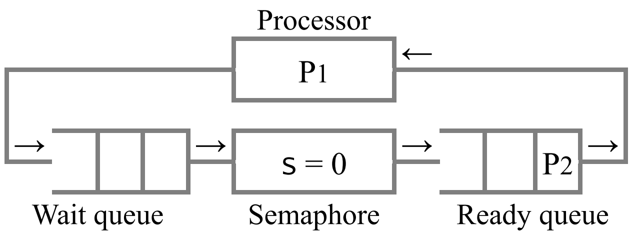


Figure 3 After R executes `signal(S)` in Figure 2

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in Figures 6 and 7.

Figures 4 through 7 show another example with four processes P1, P2, P3 and R. P1, P2 and P3 need the resource and R supplies the resource.

Figure 4 shows the current state where P2 is running. P3, R, and P1 are in the ready queue, and no process is in the wait queue. Here, P1, which is in the ready queue, has already executed `wait(S)`.

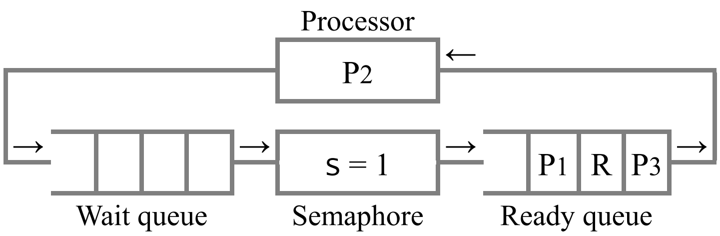


Figure 4 P2 is running

After P2 executes `wait(S)` in Figure 4, `S` is changed to 0, and P2 is not blocked because `S` ≥ 0. Subsequently, P2 is placed into the ready queue because it is not blocked, and P3 is picked up from the ready queue for execution (Figure 5).

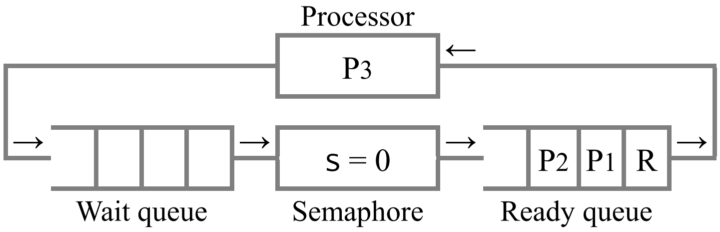


Figure 5 After P2 executes `wait(S)` in Figure 4

After P3 executes `wait(S)` in Figure 5, `S` is changed, and the queues are updated if necessary. R is picked up from the ready queue for execution (Figure 6).

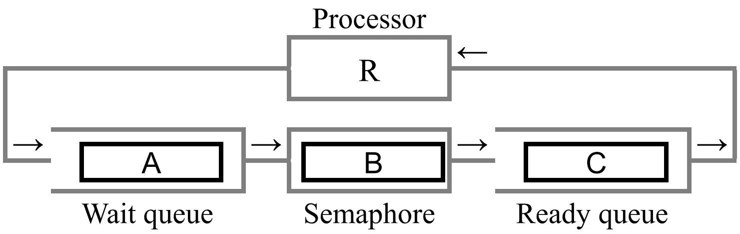
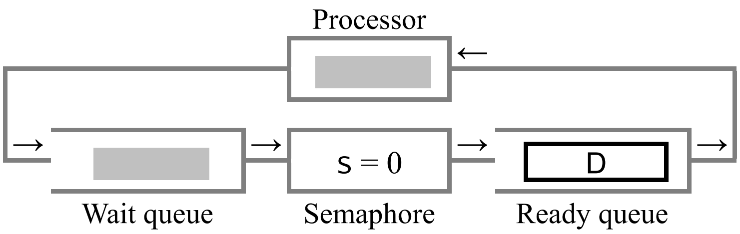


Figure 6 After P3 executes `wait(S)` in Figure 5

After R executes `signal(S)` in Figure 6, `S` is changed, and the queues are updated if necessary. The next process is picked up from the ready queue for execution (Figure 7).



Note: Shaded parts ▒ are not shown.

Figure 7 After R executes `signal(S)` in Figure 6

Answer group for A, C and D

a)

b)

			P2

c)

			P3

d)

		P2	P1

e)

		P3	P2

f)

	P3	P2	P1

Answer group for B

a) s = -1

b) s = 0

c) s = 1

d) s = 2

Q4. Read the following description of a relational database used in an airline company, and then answer Subquestions 1 and 2.

Fly4You is an airline company that operates domestic and international flights. The airline records information of its daily flights in the table Flight. The following table shows the structure of the table Flight with its sample data.

Table: Flight

FlightID	Segment Number	Origin Airport	Depart Time	Dest Airport	Arrive Time	Meal	Flying Time	Miles	Aircraft
4U221	1	MDL	08:30	RGN	09:55	-	1:25	357	Y747
4U221	2	RGN	12:30	DAC	14:15	L	2:15	605	Y747
4U321	1	BKK	08:00	KUL	11:10	B	2:10	748	Y747
4U380	1	ULN	09:45	NRT	15:30	L	4:45	1887	X757
4U380	2	NRT	18:15	MNL	22:20	D	5:05	1879	X757
4U536	1	MNL	10:30	SGN	12:50	L	3:20	1001	X757

Each flight is identified by flight ID. There are two types of flights: direct flight and one-stop flight. One-stop flight is a flight that stops at one intermediate airport before arriving at the final destination airport.
A direct flight is registered in one row of record with segment number 1. One-stop flight is registered in two rows of records with segment numbers 1 and 2. For example, in the above sample data, flight 4U221 is a one-stop flight from MDL to DAC via RGN. The first record (segment number 1) contains the flight information from the origin airport MDL to the intermediate airport RGN, and the second record (segment number 2) contains the flight information from the intermediate airport RGN to the destination airport DAC.

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following SQL statement.

To control flight schedules at intermediate airports, the management wants to generate a report that displays all of the one-stop flights registered in the table Flight.
For this purpose, the SQL statement "SQL1" is created.

```
-- SQL statement "SQL1"
SELECT FlightID, SegmentNumber, OriginAirport, DepartTime,
       DestAirport, ArriveTime
FROM Flight
WHERE FlightID IN ( SELECT FlightID FROM Flight
                    GROUP BY [    A    ]
                    HAVING [    B    ] = 2 )
```

When the table Flight contains the above sample data, SQL1 generates the following report:

FlightID	Segment Number	Origin Airport	Depart Time	Dest Airport	Arrive Time
4U221	1	MDL	08:30	RGN	09:55
4U221	2	RGN	12:30	DAC	14:15
4U380	1	ULN	09:45	NRT	15:30
4U380	2	NRT	18:15	MNL	22:20

Answer group

a) `COUNT(*)`
b) `FlightID`
c) `SegmentNumber`
d) `SUM(FlightID)`
e) `SUM(SegmentNumber)`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the following SQL statement.

Fly4You airline manages flight reservation information in table Reservation. The table Reservation maintains the latest number of seats reserved by seat classes. There are two types of seat classes; business class and economy class. A record of the table Reservation indicates the reservation status of one segment of one flight on a specific day.
Furthermore, different types of aircraft have different number of seats by seat classes. Fly4You airline manages the number of seats by seat classes by aircraft in the table Aircraft. The structures of the table Reservation and table Aircraft are as follows:

Reservation (FlightID, SegmentNumber, FlightDate,
BusinessSeatsReserved, EconomySeatsReserved)

Aircraft (Aircraft, BusinessSeats, EconomySeats)

To reduce flight costs, Fly4You airline wants to allocate smaller aircrafts for high-vacancy-rate flights routes. For this purpose, the SQL statement "SQL2" is created, which generates a report that displays all of the flight segments whose vacancy rate (percentage of seats that are not reserved) of economy class is 50% and higher on a specific day, in the descending order of the vacancy rate. Here, the specific date is given by the host variable `:SpDate`.

```
-- SQL statement "SQL2"
SELECT F.FlightID, F.SegmentNumber, OriginAirport, DepartTime,
       DestAirport, ArriveTime, [    C    ] AS VacantRate
FROM Reservation R
     JOIN Flight F
       ON [    D    ]
     JOIN Aircraft A
       ON F.Aircraft = A.Aircraft
WHERE [    E    ] AND [    C    ] >= 50
ORDER BY VacantRate DESC
```

An example of the report created by "SQL2" is as follows:

FlightID	Segment Number	Origin Airport	Depart Time	Dest Airport	Arrive Time	Vacant Rate
4U380	1	ULN	09:45	NRT	15:30	71
4U536	1	MNL	10:30	SGN	12:50	64
4U221	2	RGN	12:30	DAC	14:15	57
4U321	1	BKK	08:00	KUL	11:10	50

Answer group for C

a) `(100 – EconomySeatsReserved) / EconomySeats * 100`
b) `1 - EconomySeatsReserved / EconomySeats * 100`
c) `100 * EconomySeatsReserved / EconomySeats`
d) `100 - 100 * EconomySeatsReserved / EconomySeats`

Answer group for D and E

a) `F.Aircraft = A.Aircraft`
b) `FlightDate = :SpDate`
c) `R.FlightID = F.FlightID`
d) `R.FlightID = F.FlightID AND R.SegmentNumber = F.SegmentNumber`
e) `R.SegmentNumber = F.SegmentNumber`

Q5. Read the following description of processing orders for purchase, and then answer Subquestions 1 and 2.

Company X is a chain of stores located in several cities. Weekly, each of the stores places orders for their supplies to a purchasing office where suppliers also send their quotations. All orders are placed and fulfilled weekly.
Given that there are several stores in every city, the purchasing office consolidates orders for every city. The suppliers selected are in the same city the order is placed, and those with the lowest price are given priority. This practice yielded enormous profit to Company X. The manager creates a program to match the orders with the suppliers and create a file containing the orders to be placed.

[Program Description]

(1) The file Order contains the information on quantities to be ordered by city and by item.

File: Order (O_City, O_Item, O_Qty)

The fields from left to right are the city code, item code, and quantity. The quantity is the consolidated quantity ordered from each store in the city. The file Order is a sequential file. Records are sorted in ascending order of the city code and item code.

(2) The file Supply contains the information on stocked quantities and offered prices by city, by item, and by supplier.

File: Supply (S_City, S_Item, S_SupID, S_Qty, S_Price)

The fields from left to right are the city code, item code, supplier ID, quantity, and price. Quantity is the maximum stocked quantity the supplier can deliver for a week. The file Supply is a sequential file. Records are sorted in ascending order of the city code, item code, and price.

(3) The file Purchase will contain the information on quantities and prices to be purchased by city, by item, and by supplier.

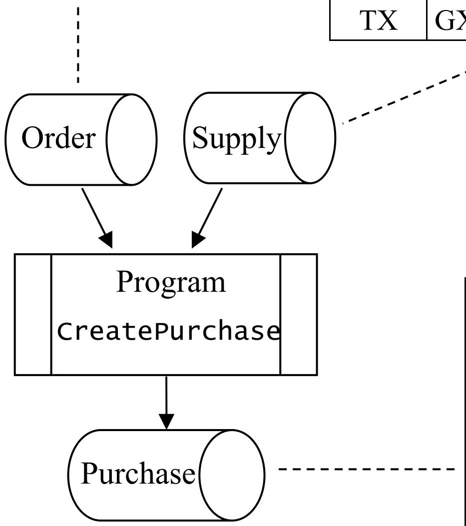
File: Purchase (P_City, P_Item, P_SupID, P_Qty, P_Price)

The fields from left to right are the city code, item code, supplier ID, quantity, and price. If two or more suppliers exist for a specific city and item, the supplier with the lowest price is given priority. The file Purchase is a sequential file created by the program.

(4) Figure 1 shows the process flow with sample data.

O_City	O_Item	O_Qty
TX	BA-CE-90	100
TX	DD-FE-10	200
TX	GX-YR-20	300

S_City	S_Item	S_SupID	S_Qty	S_Price
TX	BA-CE-90	ST-2370	200	75
TX	DD-FE-10	ST-2370	150	58
TX	DD-FE-10	YR-5678	100	60
TX	GX-YR-20	YR-5678	200	48
TX	GX-YR-20	ST-2370	50	50



P_City	P_Item	P_SupID	P_Qty	P_Price
TX	BA-CE-90	ST-2370	100	75
TX	DD-FE-10	ST-2370	150	58
TX	DD-FE-10	YR-5678	50	60
TX	GX-YR-20	YR-5678	200	48
TX	GX-YR-20	ST-2370	50	50

Figure 1 process flow with sample data

In Figure 1, for example, items BA-CE-90 and DD-FE-10 are processed as follows:

BA-CE-90: The purchasing office wants to purchase 100 BA-CE-90s, and supplier ST-2370 has enough (200) stocks; therefore, 100 BA-CE-90s are ordered from ST-2370.

DD-FE-10: The purchasing office wants to purchase 200 DD-FE-10s, but the first supplier ST-2370 has only 150 stocks; therefore, 150 DD-FE-10s are ordered from ST-2370, and the remaining 50 are ordered from the second supplier YR-5678.

(5) In the program, the following statements are used:

- `open file(`*filename*`) atEOF(`*eof_process*`)`

  Open the file *filename*. In subsequent `read` operations, if end of file is reached, *eof_process* is executed. The value `"|EOF|"` is higher than any other city code.

- `read(`*filename*`) into(`*field1*`,` *field2*`, …)`

  Read the next record from the file *filename*. The fields of the record are stored into the variables *field1*, *field2*, … .

- `write(`*filename*`) from(`*value1*`,` *value2*`, …)`

  Write the next record into the file *filename*. The fields of the record are constructed from the values *value1*, *value2*, … .

- `close file(`*filename*`)`

  Close the file *filename*.

[Program]

```
  ○ Program: CreatePurchase
    ○ String type: O_City, O_Item, S_City, S_Item, S_SupID
    ○ Numeric type: O_Qty, S_Qty, S_Price

    • open file(Order) atEOF(O_City ← "|EOF|")
    • open file(Supply) atEOF(S_City ← "|EOF|")
    • open file(Purchase)
    • read(Order) into(O_City, O_Item, O_Qty)
    • read(Supply) into(S_City, S_Item, S_SupID, S_Qty, S_Price)
    ■ O_City ≠ "|EOF|"
    | ▲ O_City = S_City and O_Item = S_Item
    | | ■ O_City = S_City and O_Item = S_Item and O_Qty > 0
    | | | ▲ O_Qty ≤ S_Qty
    | | | | • write(Purchase) from(S_City, S_Item,
    | | | |                          S_SupID, [    A    ], S_Price)
    | | | | • O_Qty ← 0
    | | | +------
    | | | | • write(Purchase) from(S_City, S_Item,
    | | | |                          S_SupID, S_Qty, S_Price)
    | | | | • O_Qty ← [    B    ]
α → | | | | • read(Supply) into(S_City, S_Item, S_SupID, S_Qty, S_Price)
    | | | ▼
    | | ■
β → | | • read(Order) into(O_City, O_Item, O_Qty)
    | ▼
    | ▲ [    C    ]
γ → | | • read(Order) into(O_City, O_Item, O_Qty)
    | ▼
    | ▲ [    D    ]
δ → | | • read(Supply) into(S_City, S_Item, S_SupID, S_Qty, S_Price)
    | ▼
    ■
    • close file(Purchase)
    • close file(Supply)
    • close file(Order)
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the program.

Answer group for A and B

a) `O_Qty`
b) `O_Qty - S_Qty`
c) `S_Qty`
d) `S_Qty - O_Qty`

Answer group for C and D

a) `(O_City < S_City) and (O_Item < S_Item)`
b) `(O_City < S_City) or (O_City = S_City and O_Item < S_Item)`
c) `(O_City < S_City) or (O_Item < S_Item)`
d) `(O_City > S_City) and (O_Item > S_Item)`
e) `(O_City > S_City) or (O_City = S_City and O_Item > S_Item)`
f) `(O_City > S_City) or (O_Item > S_Item)`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

To control out of stock of items, the purchasing office wants to print the following error messages (1) and (2):

(1) "Item *ii* for city *cc*: cannot be fulfilled fully"
(2) "Item *ii* for city *cc*: cannot be fulfilled partially"

(1) The case where the file Supply does not contain the record that has item *ii* and city *cc*.
(2) The case where the total of each supplier's quantity for item *ii* and city *cc* is less than the ordered quantity.

To implement message (1), insert the following statement to the program immediately before the `read` statement pointed by $\boxed{\text{ E }}$.

```
• print("Item ", O_Item, " for city ", O_City,
        ": cannot be fulfilled fully")
```

To implement message (2), insert the following statements to the program immediately before the `read` statement pointed by $\boxed{\text{ F }}$.

```
▲ O_Qty > 0
│ • print("Item ", O_Item, " for city ", O_City,
│         ": cannot be fulfilled partially")
▼
```

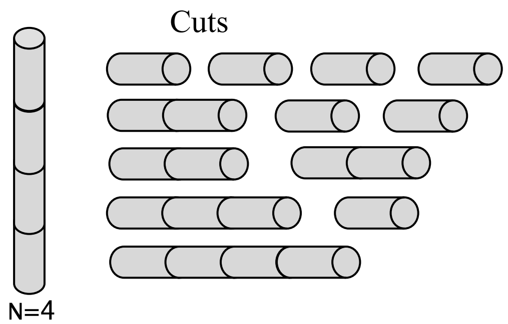
Answer group for E and F

a) α   b) β   c) γ   d) δ

Q6. Read the following description of a program and the program itself, and then answer Subquestions 1 and 2.

The rod cutting problem is a classic problem in combinatorial optimization. There is a rod with a length of `N` units. This rod can be cut into several pieces with length smaller than N. Here, the prices of all pieces are given. The problem is to determine the maximum revenue obtainable by cutting up the rod and selling the pieces. Figure 1 shows an example of a rod with a length of 4 and the total revenues corresponding to the possible combinations of cutting up the rod. In this case, the value of an optimal solution (maximum revenue) is 13, and the sizes of the pieces for an optimal solution to cut off are 1 and 3.

Length	1	2	3	4
Price	3	5	10	12



Revenues

`3 + 3 + 3 + 3 = 12`
`5 + 3 + 3 = 11`
`5 + 5 = 10`
`10 + 3 = 13` (maximum revenue)
`12`

Figure 1 Sample rod with a length of N = 4

[Program Description]

(1) `N` contains the length of the rod.

(2) Indexes of arrays `P[]` and `S[]` start at 1. An index of array `R[]` starts at 0.

(3) `P[i]` contains the price of the piece of length `i`.

(4) `R[]` contains a list of values of an optimal solution of each sub-problem. `R[i]` contains the value of an optimal solution when the length of the rod is `i`. In Figure 1, for example, when the length of the rod is 2, the optimal solution is 6 (3 + 3); therefore, `R[2]` is 6 in this case.

(5) The subprogram `rodCut()` finds an optimal solution of the problem using the bottom-up method of dynamic programming. To solve the problem, the subprogram `rodCut()` solves the same type of sub-problems in smaller sizes starting with the smallest and increasing according to their size. When solving a particular sub-problem, all of the smaller sub-problems have already been solved, and their solutions are saved in array `R[]`. To solve the problem of size 1, a solution of the problem of size 0 is needed. Therefore, the index of array `R[]` starts at 0.

(6) `S[]` contains a list of the optimal size of the first piece to cut off in each sub-problem. This array helps to reconstruct the sizes of the pieces in an optimal solution.

(7) Table 1 shows the execution results of the subprogram `rodCut()` in the case shown in Figure 1.

Table 1 Execution results in the case shown in Figure 1

N	1	2	3	4
P[N]	3	5	10	12
R[N]	3	6	10	13
S[N]	1	1	3	1

(8) The subprogram `print()` displays the contents of an array, and the subprogram `displayPiece()` displays the complete list of sizes for each piece in an optimal solution from array `S[]`. In the case shown in Figure 1, the sizes of the piece of an optimal solution to cut off are 1 and 3. Hence, the subprogram `displayPiece()` displays the result "`1 3 `".

[Program]

```
○ Program: RodCutSolver
  ○ Global: Integer type: N ← 4                      ┐
  ○ Global: Integer type: P[N] ← {3, 5, 10, 12}      ┘ ← α
  ○ Global: Integer type: R[N+1], S[N]

  ○ Subprogram: rodCut()
    ○ Integer type: I, J, Q

    • R[0] ← 0
    ■ I: 1, I ≤ N, 1
    | • Q ← 0
    | ■ J: 1, J ≤ [  A  ], 1
    | | ▲ Q < P[J] + R[  B  ]
    | | | • Q ← P[J] + R[  B  ]
    | | | • S[I] ← J
    | | ▼
    | ■
    | • R[I] ← Q
    ■

  ○ Subprogram: displayPiece()
    ■ N > 0
    | • print(S[N] + " ")
    | • N ← [  C  ]
    ■
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A and B

a) `I`
b) `I - J`
c) `J`
d) `J - I`
e) `N`
f) `N - I`
g) `N - J`

Answer group for C

a) `N - 1`
b) `N - S[N]`
c) `N - S[N - 1]`
d) `S[N] - S[N - 1]`
e) `S[N - S[N]]`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in Table 2. Here, assume that correct answers are inserted in blanks $\boxed{\text{A}}$ through $\boxed{\text{C}}$ in the program.

To simulate another case of data, two statements marked by "← α" are replaced with the following statements:

```
○ Global: Integer type: N ← 5
○ Global: Integer type: P[N] ← {3, 5, 10, 14, 16}
```

Then, the subprogram `rodCut()` is executed. Table 2 shows the execution results.

Table 2 Execution results in the case described above

N	1	2	3	4	5
P[N]	3	5	10	14	16
R[N]	3	6	$\boxed{\text{D}}$	14	$\boxed{\text{E}}$
S[N]	1	1	3	4	$\boxed{\text{F}}$

Answer group for D and E

a) 8
b) 10
c) 11
d) 13
e) 16
f) 17
g) 18
h) 19

Answer group for F

a) 1
b) 2
c) 4
d) 5

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestions 1 and 2.

[Program Description]

This program counts words from a sentence. Here, a word means a character string (hereinafter, a string) with no white-space in it. For instance, in the input string "I love C programming", there are four words, i.e. "I", "love", "C" and "programming".

(1) An input string is stored in the string `line` (see the program). The length of the input string is greater than 0 and less than 100. Each word in the input string will have at most 50 characters. The input string does not start with a white-space. The words in the input string are separated by exactly one white-space in between them and the last word of the input string does not have any white-space at the end of it.

(2) The program outputs all the unique words in the input string in ASCII-code order (uppercase characters come before lowercase characters) followed by the frequency of their occurrence separated by a white-space between them. The words are grouped in the ascending order of the frequency of the words, that is, the words with the lowest frequency will appear first.

(3) When the string `line` contains the following string,

```
I love C programming and my friend loves C programming too
```

the program outputs the following list of words:

```
I 1
and 1
friend 1
love 1
loves 1
my 1
too 1
C 2
programming 2
```

(4) In the program, there is one user defined structure named `Word`. This structure has an array of character variable `str` to store unique word, and an integer variable `frequency` to store the frequency of the unique word. Here, the array of structure `word[]` is a global variable.

(5) The following four functions are used in the program.

(i) `void analyse(char* line)`

This function separates each word in the string `line` and adds them in the array `word`.

(ii) `void setCount(char* s)`

This function checks the uniqueness of the current word `s`. If `s` is a new word, then it is added to the array `word`, and the `frequency` is set to 1. If `s` is not unique, then the `frequency` in the existing `word` entry is increased by 1. In both cases, if the `frequency` is greater than the `highestFreq`, then `highestFreq` is updated with the value of `frequency`.

(iii) `void sortByWords()`

This function sorts the array `word` in ASCII-code order of the variable `str`.

(iv) `void outputWords()`

This function outputs the words in the array `word`. Here, the words with lower frequency appear early in ASCII-code order.

(6) The following two library functions are used in the program.

(i) `int strcmp(const char* s1, const char* s2)`

This function compares the string pointed by `s1` and the string pointed by `s2`. When `s1` < `s2`, it returns a negative value; when `s1` = `s2`, it returns 0; when `s1` > `s2`, it returns a positive value.

(ii) `char* strcpy(char* s1, const char* s2)`

This function copies the string pointed by `s2` (including the null character) to the character array destination `s1`. It returns the pointer to the character array `s1`.

[Program]

```
#include <stdio.h>
#include <string.h>

#define MAX_WORD_LEN 50 + 1
#define MAX_INPUT_LEN 100 + 1
```

```
void analyse(char*);
void setCount(char*);
void sortByWords();
void outputWords();

struct Word {
   char str[MAX_WORD_LEN];
   int frequency;
} word[MAX_INPUT_LEN];

int wordCount, highestFreq;

int main() {
   char line[MAX_INPUT_LEN];

   strcpy(line, "Apple apple red apple green apple");
   analyse(line);
   sortByWords();
   outputWords();
   return 0;
}

void analyse(char* line) {
   int i = 0, j = 0;
   char s[MAX_WORD_LEN];

   wordCount = 0;
   highestFreq = 0;

   while (line[i]) {
      if ([      A      ]) {
         s[j] = '\0';
         setCount(s);
         j = 0;
         i++;
      } else {
         [      B      ];
      }
   }
   s[j] = '\0';
   setCount(s);
}
```

```
void setCount(char* s) {
   int i;
   for(i = 0; i < wordCount; i++) {
      if (!strcmp(word[i].str, s)) {
         word[i].frequency++;
         if (word[i].frequency > highestFreq)
            highestFreq = word[i].frequency;    // ← α
         return;
      }
   }
   strcpy(word[wordCount].str, s);              // ← β
   word[wordCount].frequency = 1;
   if (     C      ) {
      highestFreq = 1;
   }
   wordCount++;
}

void sortByWords() {
   int i, j;
   struct Word temp;
   for(i = 0; i < wordCount - 1; i++) {
      for(j = i + 1; j < wordCount; j++) {
         if (strcmp(word[i].str, word[j].str) > 0) {
            temp = word[j];
            word[j] = word[i];
            word[i] = temp;
         }
      }
   }
}

void outputWords() {
   int i, j;
   for(i = 1; i <=      D      ; i++) {
      for(j = 0; j <      E      ; j++) {
         if (word[j].frequency == i)
            printf("%s %d\n", word[j].str, word[j].frequency);
      }
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A

a) `line[i] == ' '`
b) `line[i] == '0'`
c) `line[i] == '\0'`
d) `s[j] == ' '`
e) `s[j] == '0'`
f) `s[j] == '\0'`

Answer group for B

a) `s[++j] = line[++i]`
b) `s[j++] = line[i++]`
c) `s[j++] = line[i]`
d) `s[j] = line[i++]`
e) `s[j] = line[i]`

Answer group for C

a) `highestFreq > 1`
b) `wordCount < highestFreq`
c) `wordCount == 0`
d) `wordCount == highestFreq`
e) `wordCount > 0`
f) `wordCount > highestFreq`

Answer group for D and E

a) `highestFreq`
b) `highestFreq - 1`
c) `wordCount`
d) `wordCount - 1`
e) `word[i].frequency`
f) `word[j].frequency`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

The program is executed with the input string `"Apple apple red apple green apple"`. Until the program is terminated, the statement marked by "$\leftarrow \alpha$" will be executed $\boxed{\text{F}}$ time(s), and the statement marked by "$\leftarrow \beta$" will be executed $\boxed{\text{G}}$ time(s).

Answer group

a) 1
b) 2
c) 3
d) 4
e) 5
f) 6

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestions 1 and 2.

[Program Description]

The class `CascadedCounter` represents a multi-purpose cascaded modular counter, which can be used to model a counting mechanism. These mechanisms are used in a digital clock, taxi-meter, up/down counter, and many others. For example, in case of using the model to count time, the hour, minute, and second will be represented by individual modular counting objects, which will be linked / cascaded so that when counting up, after 59 seconds, the seconds becomes 0 and the minute becomes 1. The methods to manipulate the model are invoked / called from the main method. All values that are set are automatically changed to modulo LIMIT. For example, the second is limited to 59 (modulo 60). Setting/increasing it to 60 will automatically make it zero.

The following is an example of the process of a multi-purpose cascaded modular counter:

(1) Create to count second and setting its limit to 59 (modulo 60),

second
0

(2) To count hour and minute, after attaching two more counters to second,

hour	minute	second
0	0	0

(3) After counting up once,

hour	minute	second
0	0	1

(4) After counting up 58 times,

hour	minute	second
0	0	59

(5) After counting up once,

hour	minute	second
0	1	0

The class `CascadedCounter` implements the following functions:

- Increase/decrease the value of counter to count up/down
- Attach additional modules to counter, e.g. attach "minute" in front of "second"
- Set the counter to certain value
- Reset the counter

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank ☐ in [Program 1].

At first, the class `Counter` that represents single counter and `Tester1` that is the tester class of class `Counter` are implemented.
The class `Counter` throws `IllegalArgumentException()` if a negative integer is given to variable `value` or non-positive integer is given to variable `limit`.
The following statements are outputted when class `Tester1` is executed.

```
second(s): 0
second(s): 59
second(s): 30
second(s): 59
second(s): 0
second(s): 1
```

[Program 1]

```
public class Counter {
   private int value, limit;
   private String label;

   Counter(int value, int limit, String label) {
      if (limit <= 0) {
         throw new IllegalArgumentException();
      }
      this.limit = limit;
      setValue(value);
      this.label = label;
   }

   public int getValue() {
      return value;
   }
```

```
    public void setValue(int value) {
        if (value < 0){
            throw new IllegalArgumentException();
        }
        this.value = [    A    ];
    }

    public void resetValue() {
        setValue(0);
    }

    public int getLimit() {
        return limit;
    }

    public void countUp() {
        setValue(getValue() + 1);
    }

    public void countDown() {
        if ([    B    ]) {
            setValue(getValue() - 1);
        } else {
            setValue(getLimit() - 1);
        }
    }

    public String toString() {
        return label + ": " + getValue();
    }
}
```

[Program 2]

```
public class Tester1 {
   public static void main(String[] args) {
      Counter c = new Counter(0, 60, "second(s)");
      System.out.println(c);
      c.countDown();
      System.out.println(c);
      c.setValue(30);
      System.out.println(c);
      c.setValue(59);
      System.out.println(c);
      c.countUp();
      System.out.println(c);
      c.countUp();
      System.out.println(c);
   }
}
```

Answer group for A

a) `limit - value` b) `value % limit`
c) `value - limit` d) `value / limit`

Answer group for B

a) `getValue() < getLimit()` b) `getValue() <= getLimit()`
c) `getValue() > 0` d) `getValue() >= 0`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank [ ] in [Program 3].

In addition to the classes from Subquestion 1, there are two classes in the following programs: `CascadedCounter` and `Tester2`.
The following statements are outputted when the class `Tester2` is executed.

```
hour(s): 0   minute(s): 0   second(s): 59
hour(s): 0   minute(s): 1   second(s): 0
hou(s): 0   minute(s): 59   second(s): 59
hour(s): 1   minute(s): 0   second(s): 0
hour(s): 0   minute(s): 59   second(s): 59
```

[Program 3]

```
public class CascadedCounter extends Counter {
   private Counter leftSide;

   CascadedCounter(int value, int limit, String label) {
      super(value, limit, label);
   }

   public void countUp() {
      if ([    C    ] && [    D    ]) {
         leftSide.countUp();
      }
      super.countUp();
   }

   public void countDown() {
      if ([    E    ] && [    D    ]) {
         leftSide.countDown();
      }
      super.countDown();
   }

   public void resetValue() {
      super.resetValue();
      if ([    D    ]) {
         leftSide.resetValue();
      }
   }

   public void connectLeft(Counter leftSide) {
      this.leftSide = leftSide;
   }

   public void disconnectLeft() {
      this.leftSide = null;
   }

   public String toString() {
       String str = "";
       if ([    D    ]) {
          str += leftSide;
       }
       str += "   " + ([    F    ]);
       return str;
   }
}
```

[Program 4]

```
public class Tester2 {
   public static void main(String[] args) {
      CascadedCounter sec = new CascadedCounter(59, 60, "second(s)");
      CascadedCounter min = new CascadedCounter(0, 60, "minute(s)");
      CascadedCounter hr = new CascadedCounter(0, 24, "hour(s)");
      sec.connectLeft(min);
      min.connectLeft(hr);
      System.out.println(sec);
      sec.countUp();
      System.out.println(sec);
      min.setValue(59);
      sec.setValue(59);
      System.out.println(sec);
      sec.countUp();
      System.out.println(sec);
      sec.countDown();
      System.out.println(sec);
   }
}
```

Answer group for C and E

a) `(getValue() + 1) == getLimit()`
b) `getValue() < getLimit()`
c) `getValue() <= getLimit()`
d) `getValue() == 0`
e) `getValue() == getLimit()`
f) `getValue() > 0`

Answer group for D

a) `leftSide != null`
b) `leftSide == null`
c) `super.leftSide != null`
d) `super.leftSide == null`

Answer group for F

a) `leftSide`
b) `leftSide.toString()`
c) `super`
d) `super.toString()`
e) `this`
f) `this.toString()`
g) `toString()`

23rd ITPEC FE Afternoon Exam -- April 2017

Q	SQ	BQ	Correct Answer
1	1	A	g
		B	e
		C	b
	2	D	d
		E	c
		F	a
2	1	A	b
		B	b
		C	d
	2	D	d
		E	a
3	A		c
	B		a
	C		d
	D		e
4	1	A	b
		B	a
	2	C	d
		D	d
		E	b
5	1	A	a
		B	b
		C	b
		D	e
	2	E	c
		F	b
6	1	A	a
		B	b
		C	b
	2	D	b
		E	f
		F	a
7	1	A	a
		B	b
		C	c
		D	a
		E	c
	2	F	b
		G	d
8	1	A	b
		B	c
	2	C	a
		D	a
		E	d
		F	d

October 2016

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the s of the question you select to answer in the "Selection Column" on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the spring Fundamental IT Engineer Examination conducted?

   Answer group
   a) September b) October c) November d) December

   Since the correct answer is " b) October ", mark your answer sheet as follows:

   [Sample Answer]

Sample	a	●	c	d	e	f	g	h	i	j

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc., of procedures, variables, etc.
`/* text */`		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 ――― process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then process 1 is executed. If it is `false`, then process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the iteration process, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	`+, −, not`	High
Multiplication, division	`×, ÷, %`	
Addition, subtraction	`+, −`	
Relational operation	`>, <, ≥, ≤, =, ≠`	
Logical product	`and`	
Logical sum	`or`	Low

Note: With division of integers, an integer quotient is returned as a result.
The “%” operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description of security vulnerability, and then answer Subquestions 1 and 2.

TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are cryptographic protocols that provide communications security between a client and server over the Internet. Although TLS is the successor of SSL, they are usually referred to together as SSL/TLS.

[The handshake protocol]

Figure 1 shows the flow of information and processes between the client and server before the secure SSL/TLS session is established.

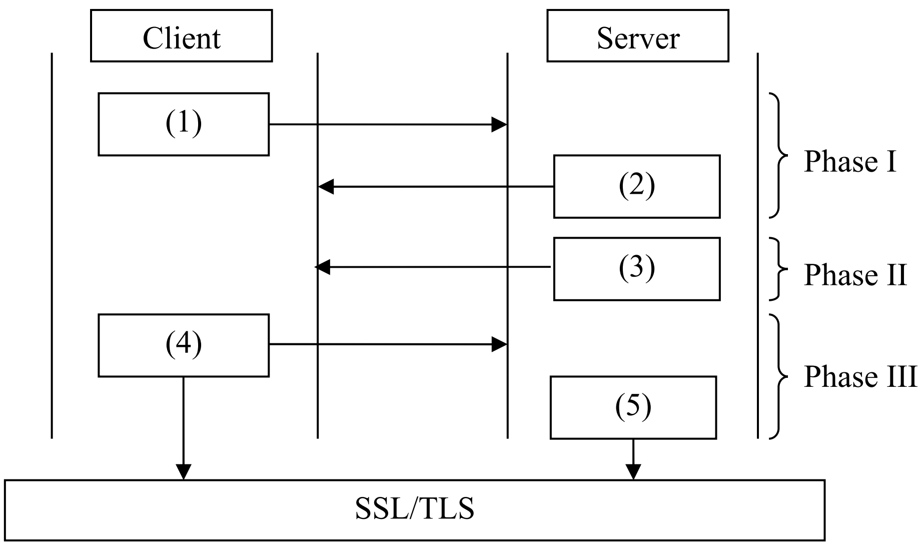


Figure 1 Overview of SSL/TLS handshake protocol

(1) In Phase I, the client sends the Client Hello message that contains the highest SSL version supported by the client, a random number to be used for master secret generation, and the list of encryption and compression algorithms supported by the client.
(2) Next, the server responds with a Server Hello message that contains the highest SSL version supported by the client and server, another random number for master secret generation, and the selected encryption and compression algorithm to be used.
(3) In Phase II, in most cases, the server sends its certificate (or certificate chain) to the client. At this stage, the $\boxed{\text{A}}$ is available to the client. If required, the server can send a certificate request to the client as well.

(4) In Phase III, if explicitly requested, the client sends its certificate and authenticates itself with the server. Then, depending on the key exchange algorithm, the client generates its contribution to the pre-master secret. The client encrypts it with the $\boxed{\quad\text{A}\quad}$ from the server's certificate, and sends it to the server. Now, the client has all the information required to generate the master secret key using the value generated on the client side and the information obtained in previous steps.

(5) Next, the server uses the $\boxed{\quad\text{B}\quad}$ to decrypt the value received from the client. It can now also generate the master secret key. Thus the handshake on the server side is complete.

[Heartbleed Bug]

Heartbleed is vulnerability in OpenSSL, an open source implementation of the SSL/TLS protocol. The name heartbleed is derived from the heartbeat extension used to maintain the SSL/TLS session alive. Because it is costly to initiate a new SSL/TLS session, maintaining alive current sessions between the server and client for a reasonable period is a viable solution especially in Web sites and servers with a high volume of SSL/TLS traffic.

The heartbeat protocol uses two message types: Heartbeat Request and Heartbeat Response. Either side can send a heartbeat request that contains a payload message and the size of the message itself, and the other side will respond with the same payload message to indicate that it is alive. Figure 2 shows the heartbeat requests and responses, with "Hello" as the payload message, of a normal client N, a malicious client M, and a possible victim client V.

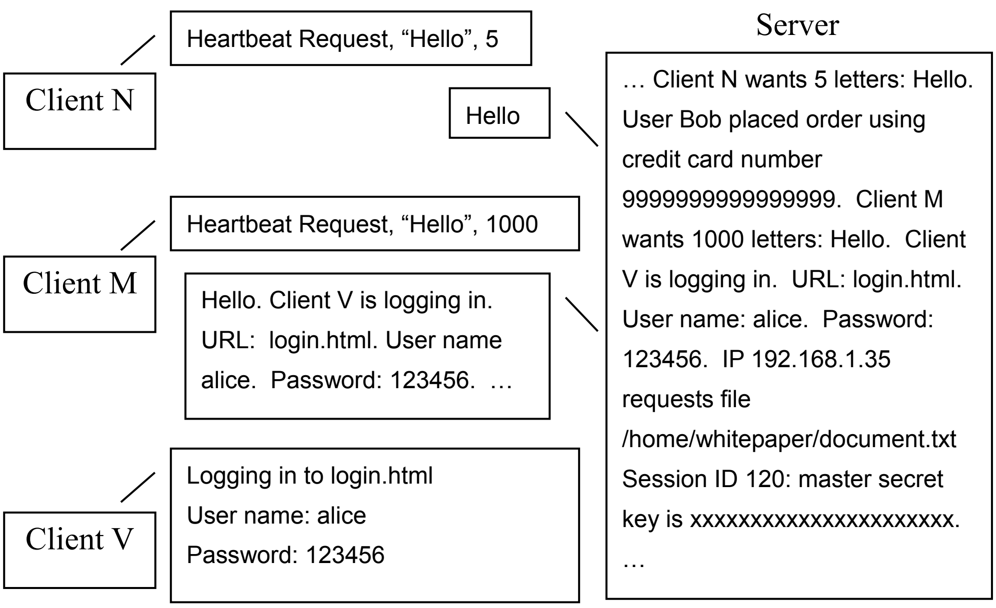


Figure 2 Example of heartbleed requests and responses

The heartbleed bug exploits the heartbeat response message by incorrectly specifying the size of the payload. For client M, the server failed to perform bound checking and returns 1000 bytes of its memory starting from the location of the payload instead of simply sending the payload itself. Because the payload size is 16-bit binary value, attackers can obtain up to [C] of the server memory with each heartbeat request. Therefore, it is possible for the server to leak sensitive information. In the worst-case scenario, if the [D] is leaked, attackers can easily decrypt all captured communications, both past and future. It also allows the attackers to impersonate as the server even if the bug is already patched.

The bug was discovered in 2014 in the older version of OpenSSL, and the new version 1.0.1g, which fixed the bug was released in April 2014. However, the bug was widespread because OpenSSL is integrated with various versions of operating systems that range from servers to mobile devices including popular Web server software. Therefore, users are urged to upgrade to the latest version of OpenSSL.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the above description. If needed, select the same answer twice or more.

Answer group for A, B and D

a) client's private key
b) client's public key
c) pre-master secret
d) master secret key
e) server's private key
f) server's public key

Answer group for C

a) 256 bytes
b) 32k bytes
c) 64k bytes
d) 256k bytes
e) 1M bytes
f) 4M bytes

Subquestion 2

From the answer group below, select <u>two</u> appropriate statements that concern the heartbleed bug.

Answer group

a) Both a client and server can send heartbeat requests. Thus, it is possible for a malicious user on the server side to read data from the client's memory as well.

b) It is a flaw in the design of the heartbeat extension. In order to avoid losing sensitive data, a user should avoid sending heartbeat requests to servers until further notice.

c) It is difficult to investigate an attack because an attacker only reads the contents of the memory and the attack leaves no trail of damage.

d) On vulnerable servers, an attacker can send illegal heartbeat request messages to alter the contents of the memory and gain root access to the operating system.

e) The bug was discovered in 2014 in the older version of OpenSSL. Therefore, it can be concluded that there were no attacks prior to the discovery.

Q2. Read the following description of arithmetic circuit for multiplication, and then answer Subquestions 1 and 2.

Hardware engineers are planning to design an arithmetic circuit that multiplies two fixed-point binary numbers in signed magnitude representation.

[Signed magnitude representation]

In signed magnitude representation, a fixed-point binary number is represented as follows:

Sign bit	Magnitude bits

The leftmost bit is the sign bit: 0 means positive, 1 means negative. The rest of the bits after the sign bit are the magnitude bits that represent the absolute value of the number.

Examples:

01001 represents +9 because the sign bit is 0 (+) and the magnitude bits are 1001 (9).

10011 represents −3 because the sign bit is 1 (−) and the magnitude bits are 0011 (3).

In signed magnitude representation, there are two zero representations: +0 and −0. When the magnitude bits are all 0s, the value is evaluated as 0 regardless of the sign bit value.

[Arithmetic circuit for multiplication]

(1) Calculation is performed in signed magnitude representation.

(2) A multiplicand and multiplier are 5 bits in length (including the sign bit), and the product is 9 bits in length (including the sign bit).

(3) There are 6 registers: Ms (1-bit register), Qs (1-bit register), Ps (1-bit register), MM (8-bit register), QM (4-bit register), and PM (8-bit register).

The following steps describe how the multiplication operation proceeds, with (−3) × (+9) = (−27), whose internal representation is 10011 × 01001 = $\boxed{\quad A \quad}$, as an example.

[1] Load the sign bit of the multiplicand to Ms.

[2] Load the magnitude bits of the multiplicand to bit positions 1 through 4 of MM. Set 0s to the rest of the bits of MM (shaded parts in Figure 1).

[3] Load the sign bit of the multiplier to Qs.

[4] Load the magnitude bits of the multiplier to QM.

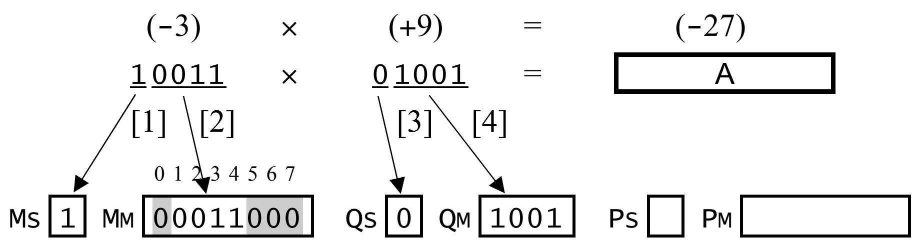


Figure 1 Execution results of steps [1] through [4]

[5] Test the rightmost bit of QM. If it is 1, copy the contents of MM to PM. If it is 0, load 00000000 to PM.

[6] Repeat [6-1] through [6-3] three times.

  [6-1] Shift the contents of PM 1-bit to the right.

  [6-2] Shift the contents of QM 1-bit to the right.

  [6-3] Test the rightmost bit of QM. If it is 1, add the contents of MM to PM.

[7] Set the result of the logical operation $\boxed{\quad B \quad}$ to PS.

[8] Finally, the sign bit of the product is obtained in PS, and the magnitude bits of the product are obtained in PM.

After the execution of [5]:	QM	1001	PM	00011000
After the first iteration of [6]:	QM	0100	PM	
After the second iteration of [6]:	QM	0010	PM	C
After the third iteration of [6]:	QM	0001	PM	

Note: Shaded parts are not shown.

Figure 2 Execution results of steps [5] and [6]

In real implementation, if PM and QM are combined into one longer register as shown below, steps [1] through [8] still work correctly, and the number of shift operations can be halved.

PM	QM

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\quad\quad}$ in the above description. Here, the symbols "&", "|", and "^" denote the logical operators AND, OR, and XOR, respectively.

Answer group for A

a) 000011011   b) 000110011

c) 100011011   d) 100110011

Answer group for B

a) (MS & QS)   b) (MS | QS)   c) (MS ^ QS)

Answer group for C

a) 00000011   b) 00000110

c) 00001100   d) 01100000

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in Figure 3.

In the multiplication operation, step [6] is executed 3 times. The hardware engineers work develop the design for a logic circuit for loop control as shown in Figure 3.
In Figure 3, the 4-bit counter (hereinafter, the counter) functions as follows:

- First, with the load signal, counter value 3 is set in the counter.
- Then, each time the counter receives the decrement signal, the counter decrements the counter value by 1, and outputs the decremented value on lines O_8, O_4, O_2 and O_1.
  - 1st time, (O_8, O_4, O_2, O_1) = (0, 0, 1, 0) because the decremented value is 2 (=0010).
  - 2nd time, (O_8, O_4, O_2, O_1) = (0, 0, 0, 1) because the decremented value is 1 (=0001).
  - 3rd time, (O_8, O_4, O_2, O_1) = (0, 0, 0, 0) because the decremented value is 0 (=0000).

The logic circuit in Figure 3 finally outputs the signal Z, which is a binary value (0 or 1). The logic circuit outputs Z = 1 when (O_8, O_4, O_2, O_1) = (0, 0, 0, 0). Otherwise, it outputs Z = 0.

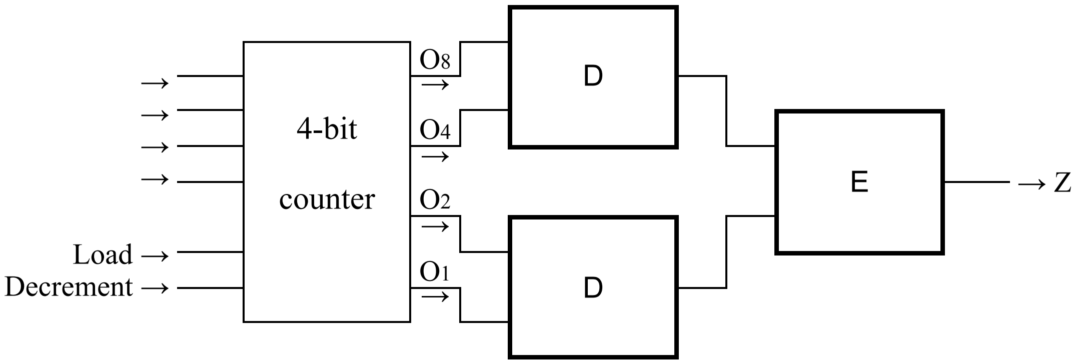
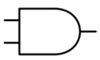
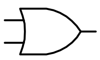
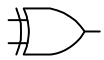
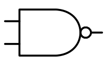
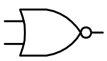


Figure 3 Logic circuit for loop control

Answer group for D and E

a)  (AND gate)   b)  (OR gate)   c)  (XOR gate)

d)  (NAND gate)   e)  (NOR gate)

Q3. Read the following description of a database system, and then answer Subquestions 1 through 3.

The Lucky Dog Grooming Parlor is a pet care shop that provides full pet-styling salon services with several facilities, especially for dogs. The shop maintains information about each pet in a table named PetTable with attributes that include each dog's ID, name, breed, pet owner's name, and the balance due on services. The table structure is as follows:

PetTable (<u>DogID</u>, DogName, Breed, OwnerName, BalanceDue)

Subquestion 1

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following SQL statement.

Some pet owners own more than one dog. The shop manager wants to generate a report that displays a list of pet owners who own more than one dog. For this purpose, the SQL statement "SQL1" is created.

```
-- SQL statement "SQL1"
SELECT  OwnerName, DogName, Breed, BalanceDue
   FROM  PetTable
   WHERE [      A      ]
ORDER BY OwnerName
```

An example of the report created by "SQL1" is as follows:

OwnerName	DogName	Breed	BalanceDue
Henry Chauncey	Buddy	Great Dane	1000
Henry Chauncey	Abe	Bulldog	300
Mike Barz	Baxter	Boxer	1000
Mike Barz	Fluffy	Poodle	0
Mike Barz	Love	Poodle	100

Answer group for A

a) `COUNT(OwnerName)>1`

b) `DogID IN (SELECT DogID FROM PetTable HAVING(COUNT(DogID)>1))`

c) `DogID IN (SELECT DogID FROM PetTable HAVING(COUNT(OwnerName)>1))`

d) `HAVING(COUNT(DogID)>1)`

e)
```
OwnerName IN (SELECT OwnerName FROM PetTable
              GROUP BY OwnerName HAVING(COUNT(OwnerName)>1))
```

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following SQL statement. If needed, select the same answer twice or more.

The Lucky Dog Grooming Parlor also wants to maintain information about each pet owner in a table named OwnerTable with attributes that include each owner's ID, name, address, township, and telephone number. The table structure is as follows:

OwnerTable (<u>OwnerID</u>, OwnerName, Address, Township, TelephoneNo)

To obtain a relationship between OwnerTable and PetTable, the attribute OwnerName in PetTable is replaced by the attribute OwnerID. Consequently, the table structure of PetTable is as follows:

PetTable (<u>DogID</u>, DogName, Breed, OwnerID, BalanceDue)

The shop manager wants to give 10% discount on the current balance due for pet owners who come from any township different from the "Wild Rose" township, which is where the shop is located. For this purpose, the SQL statement "SQL2" is created.

```
-- SQL statement "SQL2"
UPDATE [    B    ]
   SET BalanceDue = BalanceDue - (BalanceDue * 0.1)
   WHERE [    C    ] IN (SELECT [    D    ]
                             FROM [    E    ]
                             WHERE Township <> 'Wild Rose')
```

The following table shows how the balance due is updated by "SQL2".

DogID	DogName	Breed	BalanceDue before update	BalanceDue after update	Township
1	Buddy	Great Dane	1000	1000	Wild Rose
2	Abe	Bulldog	300	300	Wild Rose
3	Acridus	Great Dane	1500	1350	Schaumburg
4	Bam Bam	Bulldog	1000	900	Schaumburg
5	Baxter	Boxer	1000	900	Dubuque
6	Fluffy	Poodle	0	0	Dubuque
7	Love	Poodle	100	90	Dubuque

Answer group for B through E

a) `BalanceDue` b) `DogID` c) `OwnerID`
d) `OwnerTable` e) `PetTable` f) `Township`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank [ ] in the following SQL statement.

The shop manager also wants to generate a report that displays a list of the number of dogs and the total amount of balance dues by each township. For this purpose, the SQL statement "SQL3" is created.

```
-- SQL statement "SQL3"
SELECT [   F   ]
   FROM [   G   ]
GROUP BY OwnerTable.Township
```

An example of the report created by "SQL3" is as follows:

NoOfDogs	TotalAmount	Township
2	1300	Wild Rose
2	2250	Schaumburg
3	990	Dubuque

Answer group for F and G

a) `COUNT(DogID) AS NoOfDogs,`
   `SUM(BalanceDue) AS TotalAmount, Township`

b) `COUNT(DogID) AS NoOfDogs,`
   `TOTAL(BalanceDue) AS TotalAmount, Township`

c) `OwnerTable`

d) `OwnerTable INNER JOIN PetTable`
   `ON OwnerTable.OwnerID = PetTable.OwnerID`

e) `PetTable IN (SELECT OwnerID FROM OwnerTable`
   `HAVING PetTable.OwnerID = OwnerTable.OwnerID)`

Q4. Read the following description of network connectivity problems, and then answer Subquestions 1 and 2.

Company ABC has established its internal LANs to share resources within the office and have access to the Internet. The office is divided into two sections X and Y with 50 PCs in total. It also has a Web server hosted within the premises of an Internet service provider. For optimizing security and performance, the two sections are separated by a router.
Table 1 lists the PCs in sections X and Y. Figure 1 shows the network connectivity map.

Table 1 List of PCs connected via Switch/Hub

PCs	Connecting Switch/Hub	Section
PC01 to PC20	Switch1	X
PC21 to PC30	Hub	X
PC31 to PC50	Switch2	Y

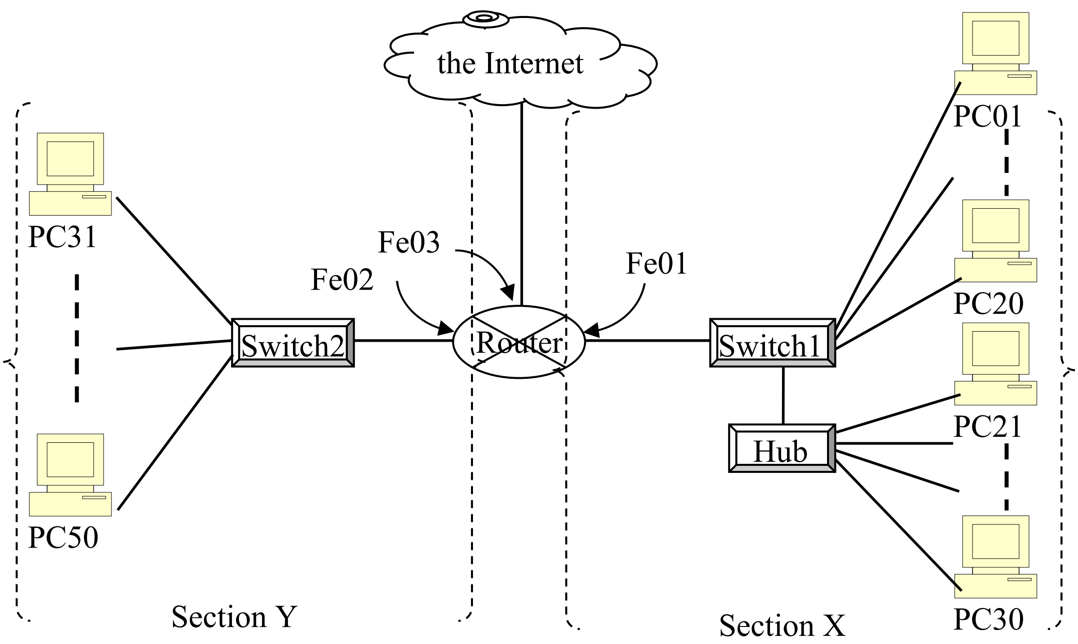


Figure 1 Network connectivity map

The PCs on both LANs are configured by class C private IP addresses with default subnet mask. Table 2 lists partial information on the device name, IP address, subnet mask, MAC address, and the default gateway of the devices.

Table 2 List of devices on internal LANs

Device Name	IP Address	MAC Address	Default gateway	Section
Fe01	192.168.1.1/24	AA:AA:AA:12:34:5A		X
Switch1	192.168.1.11/24	AA:AA:12:5E:7D:6D		
PC01	192.168.1.101/24	AA:0F:3C:33:44:01	192.168.1.1	
…	…	…	192.168.1.1	
PC20	192.168.1.120/24	AA:0F:3C:33:44:20	192.168.1.1	
PC21	192.168.1.121/24	AA:0F:3C:33:44:21	192.168.1.11	
…	…	…	192.168.1.11	
PC30	192.168.1.130/24	AA:0F:3C:33:44:30	192.168.1.11	
Fe02	192.168.3.2/24	AA:AA:AA:12:34:5B		Y
Switch2	192.168.3.22/24	AA:AA:12:5E:7D:6E		
PC31	192.168.2.231/24	AA:0F:3C:33:44:31	192.168.3.2	
…	…	…	192.168.3.2	
PC50	192.168.2.250/24	AA:0F:3C:33:44:50	192.168.3.2	
Fe03	203.129.30.57/30	AA:AA:AA:12:34:5C		Internet
Web server web.example.com	198.51.100.189/28	AB:CD:EF:12:34:56		

Subquestion 1

From the answer group below, select the appropriate answer to be inserted in each blank [ ] in the following description.

After the network configuration was completed, the employees of both sections started sharing resources within their own sections. However, most employees currently encounter problems with connectivity among sections and to the Internet. Company ABC hired a network engineer to determine the problems.

The network engineer made a plan to test connectivity using ping commands. He arranged three test cases, and started the testing.

The results of test case (1) are as follows:

Test case	Source	Destination	Command	Result	Cause
(1)	PC01 – PC20	PC21	ping 192.168.1.121	ping successful	
	PC21 – PC30	PC01	ping 192.168.1.101	ping successful	
	PC01	Router Fe01	ping 192.168.1.1	ping successful	
	PC21	Router Fe01	ping 192.168.1.1	ping successful	
	PC01 – PC20	Web Server	ping 198.51.100.189	ping successful	
	PC21 – PC30	Web Server	ping 198.51.100.189	ping failed	A

The network engineer resolved cause $\boxed{\text{A}}$, and then proceeded with test case (2).

The results of test case (2) are as follows:

Test case	Source	Destination	Command	Result	Cause
(2)	PC01 – PC20	PC31	ping 192.168.2.231	ping failed	B
	PC21 – PC30	PC31	ping 192.168.2.231	ping failed	
	PC31 – PC50	PC01	ping 192.168.1.101	ping failed	
	PC31 – PC50	PC21	ping 192.168.1.121	ping failed	
	PC31 – PC49	PC50	ping 192.168.2.250	ping successful	
	PC31 – PC50	Web Server	ping 198.51.100.189	ping failed	B
	PC31 – PC50	Router Fe02	ping 192.168.3.2	ping failed	

The network engineer resolved cause $\boxed{\text{B}}$, and then he proceeded with test case (3).

The results of test case (3) are as follows:

Test case	Source	Destination	Command	Result	Cause
(3)	PC01 – PC50	Web Server	ping web.example.com	ping failed	C

Answer group for A through C

a) Router port Fe02 was not registered with Fe01 and Fe03.

b) The DNS server address was not configured in any of the PCs.

c) The gateway addresses of all the PCs of Company ABC were not assigned to 203.129.30.57.

d) The gateway addresses of PC21 to PC30 were assigned to 192.168.1.11/24 instead of 192.168.1.1/24.

e) The IP address of Fe03 was not assigned to 192.168.2.2/24.

f) The IP address of Switch1 was not assigned to 192.168.1.1/24.

g) The IP address of Switch1 was not assigned to 192.168.3.2/24.

h) The IP addresses of PC31 to PC50 were wrongly configured to 192.168.2.xxx instead of 192.168.3.xxx.

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following description.

An employee working in Section Y has transferred to Section X. He had to move with his PC32 because it contains much information with regard to common company benefits. He connected his PC to Switch1. However, he is neither able to browse the Web sites on the Internet nor use the shared resources in Section X. He noticed that the activity indicator of his PC's Ethernet port, as well as the corresponding switch port, is blinking normally, as other PCs do. He asked the network engineer for a solution to the problem. The network engineer advised him to $\boxed{\quad D \quad}$. The employee found this helpful.

Answer group for D

a) change his PC's DNS address to 203.129.30.57 and default gateway to 192.168.1.1
b) change his PC's IP address to 192.168.1.132 and DNS to 203.129.30.57
c) change his PC's IP address to 192.168.1.132 and default gateway to 192.168.1.1
d) replace the straight through UTP cable with a crossover UTP cable
e) restart Switch1 to reset the MAC address table of the switch

Q5. Read the following description of test design for software, and then answer Subquestions 1 through 3.

Company *N*, a system integrator, is reviewing its testing method in order to reduce the number of bugs left uncorrected in the programs developed by the company.

[Description of the testing method used in Company *N*]

Company *N* mainly tests the programs developed by the company using control flow testing, which is a white box testing method.

Control flow testing focuses on the smallest units that form a program, such as commands, paths, and decision conditions. Test cases and test data are prepared in accordance with the coverage criteria defined during test planning, and the behavior of the developed programs is checked.

The coverage criteria include statement coverage, where all the statements are executed at least once in the test, and decision condition coverage (hereinafter, branch coverage), where all the paths after all the branches are executed at least once.

Company *N* uses branch coverage as its coverage criteria.

[Description of the decision condition of branch coverage used by Company *N*]

A decision condition for a branch includes a single condition that evaluates only one condition, and a multiple condition that evaluates two or more single conditions combined with "and" or "or". The following example illustrates single conditions and a multiple condition:

Example $$\underbrace{\underbrace{(\,a > b\,)}_{\text{Single condition}} \text{ and } \underbrace{(\,a < c\,)}_{\text{Single condition}}}_{\text{Multiple condition}}$$

Here, when a program is executed, short-cut evaluation is applied to a multiple condition. In short-cut evaluation, single conditions that constitute a multiple condition are evaluated in sequence from left to right. Once the result of the multiple condition is determined, the remaining single conditions are not evaluated. For example, in the case of a multiple condition in which two single conditions are combined with "and", if the evaluation result of the first single condition is false, the evaluation result of the multiple condition is false, regardless of the evaluation result of the second single condition. Therefore, the second single condition is not evaluated.

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description that concerns the decision condition of the branch coverage used by Company *N*.

Figure 1 shows a sample program to be tested, and Table 1 lists sample test cases for this program. When the program shown in Figure 1 is tested according to the decision condition of the branch coverage used by Company *N* with the test cases, the test result reveals that $\boxed{\text{A}}$ in test case 1, and $\boxed{\text{B}}$ in test case 2.

```
○ Program (Integer type: x, Integer type: a, Integer type: b,
           Integer type: c, Integer type: d)
■ x > 10
│ • func1()
│ ▲ (a < 10) or (b < 20)
│ │ • func2()
│ ├──────
│ │ • func3()
│ ▼
│ ▲ (c > 10) and (d > 10)
│ │ • func4()
│ │ • exit        /* Get out of the iteration process */
│ ├──────
│ │ • func5()
│ ▼
│ • func6()
■
```

Figure 1 Sample program to be tested

Table 1 Sample test cases

	Test data				
Variable	x	a	b	c	d
Test case 1	11	9	19	10	10
Test case 2	11	10	20	11	11

Answer group for A and B

a) `b < 20` is not evaluated
b) `b < 20` and `c > 10` are not evaluated
c) `b < 20` and `d > 10` are not evaluated
d) `c > 10` is not evaluated
e) `c > 10` and `d > 10` are not evaluated
f) `d > 10` is not evaluated
g) all single conditions are evaluated

Subquestion 2

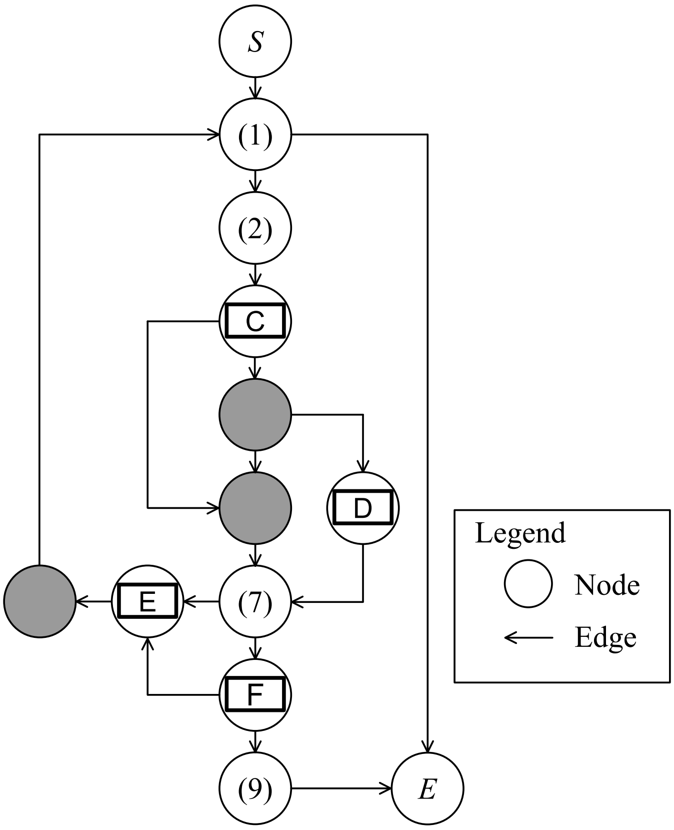
The control structure of a program can be described with a control flow graph. In a control flow graph, processes are divided into serial instructions, iteration instructions, and branch instructions, and each of them is placed in a process block (hereinafter, node) that is connected with a directed line segment (hereinafter, edge) in the sequence of process execution. Here, a multiple condition is divided into the respective single conditions, and they are placed in the control flow graph.

Figure 2 is prepared by assigning node numbers (1) through (11) to the sample program to be tested in Figure 1. Figure 3 shows the corresponding control flow graph. The node numbers in Figure 3 correspond to the node numbers in Figure 2. Nodes *S* and *E* in Figure 3 are special nodes that indicate the entry and exit of the program respectively, and there are no corresponding processes in the sample program to be tested.

From the answer group below, select the appropriate node number to be inserted in each blank $\boxed{\qquad}$ in the control flow graph in Figure 3.

```
○ Program (Integer type: x, Integer type: a, Integer type: b,
           Integer type: c, Integer type: d)
■ (1) x > 10
│ (2) • func1()
│ ▲ (3) (a < 10) or (4) (b < 20)
│ │ (5) • func2()
│ ├──────
│ │ (6) • func3()
│ ▼
│ ▲ (7) (c > 10) and (8) (d > 10)
│ │ (9) • func4()
│ │ • exit        /* Exit iteration process */
│ ├──────
│ │ (10) • func5()
│ ▼
│ (11) • func6()
■
```

Figure 2 Sample program in Figure 1 with node numbers



Note: Shaded parts are not shown.

Figure 3 Control flow graph that corresponds to sample program in Figure 2

Answer group for C through F

a) (3)	b) (4)	c) (5)	d) (6)
e) (8)	f) (10)	g) (11)	

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

For the testing of the program shown in Figure 1, in the case of the branch coverage used by Company *N*, the minimum number of test cases required is $\boxed{\text{ G }}$.
Furthermore, there is a method for making test cases by extracting paths from a control flow graph. The minimum number of paths (*S*) that cover all the edges and nodes of a control flow graph is determined with the following expression:

$$S = \text{Number of edges} - \text{Number of nodes} + 2$$

By conducting a test for *S* test cases that correspond to the extracted paths, it is possible to assure higher coverage than branch coverage.
With regard to the concerning the control flow graph in Figure 3, the value of *S* is $\boxed{\text{ H }}$. In order to reduce the number of bugs left uncorrected in its programs, Company *N* decides to test the programs with test cases based on control flow graphs.

Answer group for G and H

a) 2 b) 3 c) 4
d) 5 e) 6 f) 7

Q6. Read the following description of a program and the program itself, and then answer Subquestions 1 and 2.

The bin-packing problem is a classic problem in combinatorial optimization. There are items with different sizes, and bins with the same capacity. The problem is to determine the bins of each item, and the items that must be placed into the bin so that the total size of the items in each bin does not exceed the bin capacity, and the number of non-empty bins is minimal. In order to solve the problem, heuristic methods have been developed. Next-Fit, First-Fit, Best-Fit, the most popular methods, programs, and examples are given below.

[Program Description]

(1) `N` contains the total number of items.

(2) `C` contains the capacity of the bins. Each bin has the same capacity.

(3) `A[i]` contains the size of `i`-th item. The size of each item does not exceed the bin capacity `C`. Figure 1 shows an example of the items.

(4) Indexes of arrays start at 1.

(5) All of the following three methods place the items in the order in which they arrive.

  (a) The subprogram `NextFit` places the next item into the current bin if the item fits. If it does not fit, that bin is closed and a new bin is started. Figure 2 shows the execution results of the `NextFit` program with the items shown in Figure 1.

  (b) The subprogram `FirstFit` places the next item into the lowest numbered bin in which the item fits. If it does not fit into any open bin, a new bin is started. Figure 3 shows the execution results of the `FirstFit` program with the items shown in Figure 1.

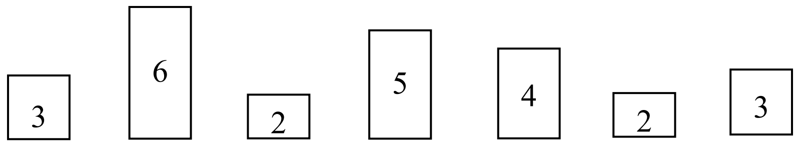
  (c) The subprogram `BestFit` places the next item into the bin that will leave the least capacity left over after the item is placed in the bin. If it does not fit into any open bin, a new bin is started. Figure 4 shows the execution results of the `BestFit` program with the items shown in Figure 1.

(6) `B[i]` contains the sum of the size of all items placed into `i`-th bin.

(7) `S[]` is a solution to the problem. `S[i]` contains the bin number into which i-th item is placed.

(8) The subprogram `OutputResult` displays the elements of array `S[]`.

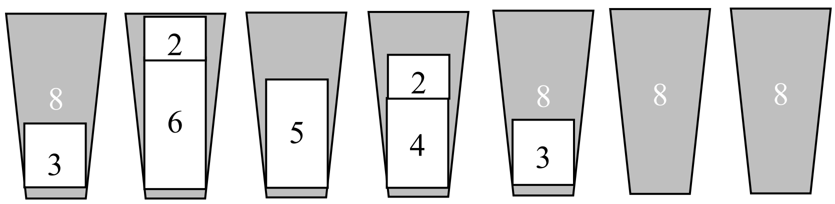
Items:



	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
Value	3	6	2	5	4	2	3

Figure 1 Example of items for programs

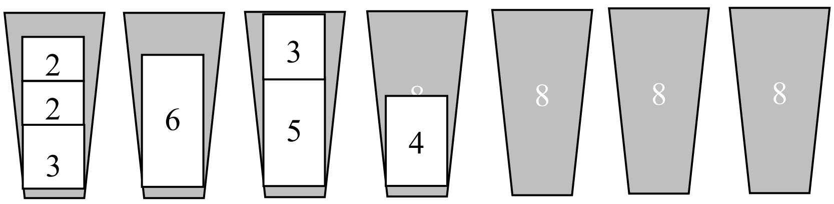
Next-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
Value	3	8	5	6	3	0	0

Figure 2 Execution results of `NextFit` program (C = 8)

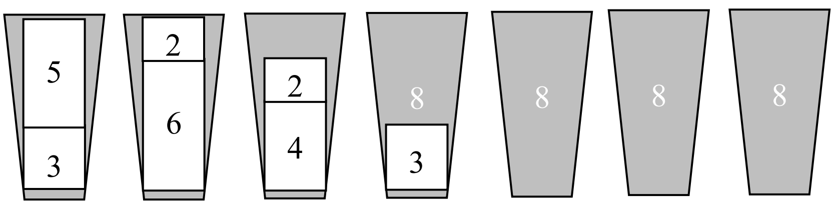
First-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
Value	7	6	8	4	0	0	0

Figure 3 Execution results of `FirstFit` program (C = 8)

Best-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
value	8	8	6	3	0	0	0

Figure 4 Execution results of `BestFit` program (C = 8)

[Program]

```
○ Global: Integer type: N ← 7, C ← 8
○ Global: Integer type: A[N] ← {3, 6, 2, 5, 4, 2, 3}
○ Global: Integer type: B[N], S[N]

○ Subprogram: OutputResult()
  ○ Integer type: I

  ■ I: 1, I ≤ N, 1
  │ • print(S[I])  /* output S[I] */
  │ • print(" ")   /* output " "  */
  ■

○ Subprogram: NextFit()
  ○ Integer type: I, J

  ■ I: 1, I ≤ N, 1
  │ • B[I] ← 0
  ■
  • J ← 1
  ■ I: 1, I ≤ N, 1
  │ ▲ C - B[J] < A[I]
  │ │ • J ← J + 1
  │ ▼
  │ • S[I] ← J
  │ • B[J] ← B[J] + A[I]
  ■
  • OutputResult()

○ Subprogram: FirstFit()
  ○ Integer type: I, J

  ■ I: 1, I ≤ N, 1
  │ • B[I] ← 0
  ■
  ■ I: 1, I ≤ N, 1
  │ ■ J: 1, J ≤ N, 1
  │ │ ▲ C - B[J] ≥ A[I]
  │ │ │ • S[I] ← [    A    ]
  │ │ │ • B[J] ← B[J] + A[I]
  │ │ │ • J ← N + 1
  │ │ ▼
  │ ■
  ■
  • OutputResult()
```

```
○ Subprogram: BestFit()
  ○ Integer type: I, J, K

  ■ I: 1, I ≤ N, 1
  │ • B[I] ← 0
  ■
  ■ I: 1, I ≤ N, 1
  │ • K ← 0
  │ ■ J: 1, J ≤ N, 1
  │ │ ▲ C - B[J] ≥ A[I]
  │ │ │ ▲ ([    B    ]) or (C - B[J] < [    C    ])
  │ │ │ │ • K ← J
  │ │ │ ▼
  │ │ ▼
  │ ■
  │ • S[I] ← [    D    ]
  │ • B[K] ← [    E    ]
  ■
  • OutputResult()
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A and D

a) `A[I]`	b) `I`	c) `J`
d) `K`	e) `S[I] + 1`	f) `S[I] + A[I]`
g) `S[I] + B[J]`	h) `S[I] + J`	

Answer group for B

a) `I = J`	b) `I ≠ J`	c) `K = 0`
d) `K = N`	e) `K > 0`	f) `K > J`

Answer group for C

a) `A[J]`	b) `A[K]`	c) `B[J]`
d) `B[K]`	e) `C`	f) `C – A[I]`
g) `C – B[J]`	h) `C – B[K]`	

Answer group for E

a) `B[J] + A[I]`	b) `B[J] + B[K]`	c) `B[K] + A[I]`
d) `B[K] + A[J]`		

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description. Here, it is assumed that the correct answers from Subquestion 1 are inserted into all the blanks in the [Program]. If needed, select the same answer twice or more.

In order to simulate another case, the first 3 lines of the program are replaced with the following 3 lines, and then the subprograms are executed.

```
○ Global: Integer type: N ← 10, C ← 10
○ Global: Integer type: A[N] ← {4, 8, 5, 7, 6, 1, 4, 2, 2, 1}
○ Global: Integer type: B[N], S[N]
```

The following table shows the execution results.

Subprogram	Printed result
NextFit()	$\boxed{\text{F}}$
FirstFit()	$\boxed{\text{G}}$
BestFit()	$\boxed{\text{H}}$

Answer group for F through H

a) 1 2 1 2 3 4 3 4 4 5
b) 1 2 1 3 4 1 4 2 3 3
c) 1 2 1 3 4 3 4 2 3 2
d) 1 2 3 4 5 5 6 6 6 6
e) 1 2 3 4 5 6 7 8 9 10
f) 4 8 5 7 7 9 0 0 0 0
g) 10 10 10 10 0 0 0 0 0 0

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the (S) in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestion.

This program is an Armstrong number calculation that will check whether an input number is an Armstrong number. An Armstrong number is an *n*-digit number that is equal to the sum of the *n*-th powers of its digits.

For example,

$371 = 3^3 + 7^3 + 1^3$

$1634 = 1^4 + 6^4 + 3^4 + 4^4$

[Program Description]

(1) The program receives a number that is a base-10 number.

(2) The input number is a positive integer. The number of digits of the input number is 8 or fewer, because the sum of the 8-th powers of its digits does not exceed the maximum value of long integer.

(3) The program calculates the number of digits.

(4) The program computes the summation of individual digits raised to the power number of digits. If the summation is equal to the input number, the number is an Armstrong number.

(5) The following 3 functions are used in the program.

(i) `long calc_power(int base, int power)`
The function calculates the power number of digits.

(ii) `int number_digits(long input_number)`
The function calculates the number of digits.

(iii) `int check_armstrong (long n_input)`
The function calculates the summation of the input digits and checks whether the input number is an Armstrong number.

(6) The following is a sample output of the Armstrong number calculation program.

```
Enter a number:
8208
8208 has 4 digits.
8208 is an Armstrong number.
```

[Program]

```
#include <stdio.h>

long calc_power(int, int);
int number_digits(long);
int check_armstrong(long);

int main () {
   long n_input;

   printf("Enter a number:\n");
   scanf("%ld", &n_input);

   printf("%ld has %d digits.\n", n_input, number_digits(n_input));

   if (check_armstrong(n_input))
      printf("%ld is an Armstrong number.\n", n_input);
   else
      printf("%ld is not an Armstrong number.\n", n_input);
   return 0;
}

long calc_power(int base, int power) {
   int i;
   long pw = 1;

   for (i = 0; [    A    ]; i++)
      pw = pw * base;

   return pw;
}

int number_digits(long input_number) {
   int n_digits = 0;

   while ([    B    ]) {
      n_digits++;
      [    C    ];
   }

   return n_digits;
}
```

```
int check_armstrong(long n_input) {
   long tmp, sum = 0;
   int remainder, digits;

   digits = number_digits(n_input);
   tmp = n_input;

   while (tmp != 0) {
      remainder = tmp % 10;
      sum = sum + [    D    ];
      tmp = tmp / 10;
   }

   return [    E    ];
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the above program.

Answer group for A

a) `i < base + power`
b) `i < base * power`
c) `i < power`
d) `i <= base + power`
e) `i <= base * power`
f) `i <= power`

Answer group for B

a) `input_number != 0`
b) `input_number != 1`
c) `input_number == 0`
d) `input_number == 1`
e) `n_digits < input_number`
f) `n_digits <= input_number`

Answer group for C

a) `input_number = input_number + 10`
b) `input_number = input_number - 10`
c) `input_number = input_number * 10`
d) `input_number = input_number / 10`
e) `input_number = input_number % 10`

Answer group for D

a) `calc_power(digits, remainder)`
b) `calc_power(digits, remainder) * calc_power(n_input, remainder)`
c) `calc_power(n_input, remainder)`
d) `calc_power(remainder, digits)`
e) `calc_power(remainder, n_input)`
f) `calc_power(remainder, n_input) * calc_power(remainder, digits)`

Answer group for E

a) `digits != sum`
b) `digits == sum`
c) `n_input != sum`
d) `n_input - sum`
e) `n_input == sum`
f) `sum`
g) `sum - n_input`

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestion.

[Program Description]

An on-line course gives students badges based on points earned in order to encourage them to study well. To earn points, the students need to take a quiz and answer 10 multiple-choice questions with 4 options (A, B, C and D). Each correct answer is worth 50 points.

GOLD	SILVER	BRONZE
500	300	100

The badges are categorized as GOLD, SILVER and BRONZE with 500, 300, and 100 points, respectively. If the student earns at least 100 points, a BRONZE badge appears, if the points reach at least 300, a SILVER badge appears, and for 500 points, a GOLD badge appears.

The following list shows the execution results of this program.

```
Quiz:
Question 1...
Question 2...
Question 3...
Question 4...
Question 5...
Question 6...
Question 7...
Question 8...
Question 9...
Question 10...

Student: ID: 123, Name: Thomas Anderson
 Answers: A B C A B B C D C A
 Badge earned: SILVER 300
```

[Program 1]

```
public [   A   ] Badge {
   GOLD(500), SILVER(300), BRONZE(100), NONE(0);
   private int value;

   private Badge(int value) {
      this.value = value;
   }

   public int getValue(){
      return value;
   }

   public static Badge getBadge(int score) {
      for (Badge badge : values()) {
         if (score >= badge.getValue())
            return badge;
      }
      return NONE;
   }
}
```

[Program 2]

```
public class Student {
   private int id;
   private String name;
   private Badge earnedBadge;
   private char[] answers;

   public Student(int id, String name, [   B   ] answers) {
      this.id = id;
      this.name = name;
      this.answers = answers;
   }

   public [   B   ] getAnswers() {
      return answers;
   }

   public void setBadge(Badge earnedBadge) {
      this.earnedBadge = earnedBadge;
   }
```

```java
    public Badge getBadge() {
        return earnedBadge;
    }

    public String toString() {
        String str = String.format("ID: %s, Name: %s%n Answers: ",
                                   id, name);
        for (char answer : answers) {
            str += answer + " ";
        }
        str += String.format("%n Badge earned: %s %d", [    C    ]);
        return str;
    }
}
```

[Program 3]

```java
public class Quiz {
    private String[] questions = new String[10];
    private char[] answers = new char[10];

    public Quiz(String[] questions, char[] answers) {
        this.questions = questions;
        this.answers = answers;
    }

    public void setAnswers(char[] answers) {
        this.answers = answers;
    }

    public char[] getAnswers() {
        return answers;
     }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i<questions.length; i++)
            sb.append(questions[i]).append("\n");
        return sb.toString();
    }
}
```

[Program 4]

```
public class Badges {

   public static int checkQuiz(char[] correctAnswer,
                               char[] studentAnswer ) {
   int score = 0;
   for (int i = 0; i<correctAnswer.length; i++)
      if (correctAnswer[i] == studentAnswer[i])
         [      D      ];
   return score;
   }

   public static void main(String[] args) {
      String[] questions = {"Question 1...", "Question 2...",
                            "Question 3...", "Question 4...",
                            "Question 5...", "Question 6...",
                            "Question 7...", "Question 8...",
                            "Question 9...", "Question 10..."};
      char[] answers = {'A', 'B', 'C', 'A', 'D',
                        'B', 'C', 'D', 'D', 'A'};
      Quiz quiz = new Quiz(questions, answers);
      System.out.println("Quiz:\n" + quiz.toString());

      int id = 123;
      String name = "Thomas Anderson";
      char[] studentAnswer = {'A', 'B', 'C', 'A', 'B',
                              'B', 'C', 'D', 'C', 'A'};
      Student student = new Student(id, name, studentAnswer);

      int score = [      E      ];
      student.setBadge(Badge.getBadge(score));
      System.out.println("Student: " + student.toString());
   }
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above programs.

Answer group for A

a) `abstract`
b) `class`
c) `enum`
d) `interface`

Answer group for B

a) `Answers`
b) `char`
c) `char[]`
d) `String`

Answer group for C

a) `earnedBadge`
b) `earnedBadge, earnedBadge.getValue()`
c) `earnedBadge.getValue()`
d) `earnedBadge.getValue(), earnedBadge`

Answer group for D

a) `score *= 10`
b) `score *= 50`
c) `score += 10`
d) `score += 50`
e) `score ^= 10`
f) `score ^= 50`

Answer group for E

a) `checkQuiz(Quiz.getAnswers(), Student.getAnswers())`
b) `checkQuiz(Quiz.getAnswers(), student.getAnswers())`
c) `checkQuiz(quiz.getAnswers(), Student.getAnswers())`
d) `checkQuiz(quiz.getAnswers(), student.getAnswers())`

22nd ITPEC FE Afternoon Examination, Oct. 2016

Question No.	Subquestion No.		Correct Answer
1	1	A	f
		B	e
		C	c
		D	e
	2		a/c
			c/a
2	1	A	c
		B	c
		C	b
	2	D	b
		E	e
3	1	A	e
	2	B	e
		C	c
		D	c
		E	d
	3	F	a
		G	d
4	1	A	d
		B	h
		C	b
	2	D	c
5	1	A	c
		B	g
	2	C	a
		D	d
		E	f
		F	e
	3	G	b
		H	e
6	1	A	c
		B	c
		C	h
		D	d
		E	c
	2	F	d
		G	b
		H	b
7	1	A	c
		B	a
		C	d
		D	d
		E	e
8	1	A	c
		B	c
		C	b
		D	d
		E	d

April 2016

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the spring Fundamental IT Engineer Examination conducted?

   Answer group
   a) March b) April c) May d) June

   Since the correct answer is " b) April ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc., of procedures, variables, etc.
/* text */		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then process 1 is executed. If it is `false`, then process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true, false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	+, −, not	High
Multiplication, division	×, ÷, %	↑
Addition, subtraction	+, −	│
Relational operation	>, <, ≥, ≤, =, ≠	│
Logical product	and	↓
Logical sum	or	Low

Note: With division of integers, an integer quotient is returned as a result.
The "%" operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description concerning a security vulnerability, and then answer Subquestions 1 and 2.

SQL injection is an attack where attackers manipulate SQL queries being used in Web applications on the Internet or other sources. The extent of damage ranges from revealing confidential or personal data to completely destroying the database, depending on the access rights exploited by the attack. Thus, it is important to review the source codes and perform the $\boxed{\quad A \quad}$ on the developed application to prevent SQL injection attacks.
In general, a Web application consists of a Web server acting as a front-end server providing Web pages and a database server acting as a back-end server supplying data to the Web server. The Web server usually passes the data entered by the users to the database server in the form of a query string. The database server then returns a set of data to the Web server for further processing.

[Example of non-secured application]

Attackers usually make SQL injection attacks when input data are not checked and are used as part of an SQL statement. For instance, Figure 1 shows a simple sign-in form used in a Web application, and Figure 2 shows an SQL-based table named User that is used to verify the user name and password for the sign-in process.

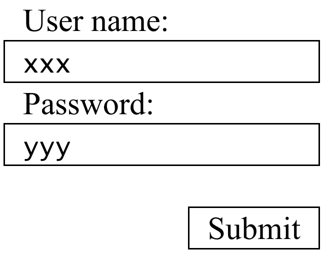


Figure 1 Simple sign-in form

User
UserID UserName Password …

Figure 2 Table named User

Communication between the Web server and the database server is summarized as follows:
(1) The Web server constructs the following SQL statement (*1):

```
SELECT UserID FROM User                                   … (*1)
  WHERE UserName='xxx' AND Password='yyy'
```

Here, the character strings in the shaded parts are copied from the input fields shown in Figure 1. This query tells the database server to return the UserID on the row whose UserName is “xxx” and Password is “yyy” in the table named User.

(2) The Web server sends the query constructed in step (1) to the database server. The database server then returns the result data to the Web server.
(3) The Web server calls an appropriate function to check if the result data have at least one row or not.
(4) If at least one row is obtained in step (3), the Web server calls an appropriate function to obtain the UserID from the result data, and completes the sign-in process successfully.

This Web application contains SQL injection vulnerabilities. It is possible for an attacker to sign-in to the application without a password or even a user name by entering a tricky input string in the user name and/or password field.

[SQL injection - Case 1]
The attacker modifies the SQL statement (*1) to the SQL statement (*2) as follows:

```
SELECT UserID FROM User                                        … (*2)
  WHERE UserName='admin' -- ' AND Password=''
```

This query tells the database server to return the UserID on the row where the UserName is admin. Here, "`--`" indicates the start of a comment field, and all the characters after "`--`" are ignored as a comment. As a result, the UserID of the UserName `admin` is returned to the application without checking the password allowing the attacker to sign-in to the application with ease.
The attacker can construct the SQL statement (*2) by entering [B] in the user name field and "" (no input; handled as a null string) in the password field in Figure 1.

[SQL injection - Case 2]
The attacker modifies the SQL statement (*1) to the SQL statement (*3) as follows:

```
SELECT UserID FROM User                                        … (*3)
  WHERE UserName='' AND Password='' OR 'A'='A' LIMIT 1 -- '
```

This query tells the database server to return the UserID on the rows where the UserName and Password are both null or `'A'='A'`. The former condition, the UserName and Password are both null, will not match any of the rows; however, the latter condition, `'A'='A'`, is always true and will match all of the rows. Furthermore, as this type of query is expected to return only a single row, the condition `LIMIT 1` is added. As a result, the UserID on the first row is returned to the application, allowing the attacker to sign-in to the application with ease.
The attacker can construct the SQL statement (*3) by entering [C] in the user name field and [D] in the password field in Figure 1.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above description.

Answer group for A

a) implementation of security patches

b) TCP/IP vulnerability scanning

c) virus checking

d) vulnerability assessment

Answer group for B

a) “`'admin -- '`”

b) “`'admin' -- `”

c) “`admin' -- `”

d) “`admin' -- '`”

Answer group for C and D

a) “”

b) “`' OR 'A'='A' LIMIT 1 -- '`”

c) “`' OR 'A'='A' LIMIT 1 -- `”

d) “` OR 'A'='A' LIMIT 1 -- '`”

Subquestion 2

From the answer group below, select the appropriate answer to be inserted in the blank $\boxed{}$ in the following description.

To strengthen the protection against SQL injection attacks such as cases 1 and 2 described in Subquestion 1, one should $\boxed{\quad\text{E}\quad}$.

Answer group for E

a) develop important applications as desktop applications instead of as Web applications since a desktop application is impervious to SQL injection attacks

b) register all passwords stored in a database as hash values instead of as plain text

c) use a packet-filtering firewall to deny all database access from anywhere other than the associated Web server

d) verify characters entered into a Web form so that none of the illegal characters exists

Q2. Read the following description concerning logic expressions, and then answer Subquestions 1 and 2.

Note: In this question, the symbols "•", "+" and "¬" are used to indicate the logical operators AND, OR and NOT respectively.

Table 1 shows a truth table for a logic circuit that has three input lines (I_1, I_2 and I_3) and one output line (Out). The truth table in Table 1 outputs value 1 when one or two of the input lines have value 1. Otherwise, it outputs value 0.

Table 2 shows the Karnaugh map that is equivalent to the truth table in Table 1. A Karnaugh map is useful for determining a single logic expression that outputs the specified value for all the cases in the truth table.

The single logic expression can be obtained from the Karnaugh map as follows:
In Table 2, there are 6 cells that have output value 1.

(1) Group these cells into three sets of two adjacent cells, as indicated by the thick lines.
(2) Obtain the logic expression that outputs value 1 for each set. For example, the set of the shaded cells ($\boxed{}$) outputs value 1 when $I_1 = 1$ and $\neg I_2 = 1$ (i.e., $I_2 = 0$) regardless of the value of I_3, and this condition can be expressed by the logic expression ($I_1 \bullet \neg I_2$).
(3) Connect the logic expressions obtained in (2) by the operator "+".

In this manner, the single logic expression $\boxed{\quad A \quad}$, that outputs the specified value for all the cases in the truth table, is obtained.

Table 1 Truth table

I_1	I_2	I_3	Out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 2 Karnaugh map

		I_2, I_3			
		0, 0	0, 1	1, 1	1, 0
I_1	0	0	1	1	1
	1	1	1	0	1

Table 3 shows another truth table for a logic circuit. Table 3 outputs value 1 when all three input lines have value 0 or all three input lines have value 1. Otherwise, it outputs value 0.

Table 3 Another truth table

I_1	I_2	I_3	Out
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

The single logic expression for Table 3 is as follows:

$$(\neg I_1 \bullet \neg I_2 \bullet \neg I_3) + (I_1 \bullet I_2 \bullet I_3) \qquad \ldots (*1)$$

The expression (*1) has 4 "•" operators, 1 "+" operator and 3 "¬" operators. When a logic circuit is assembled based on the expression (*1), 8 logic gates (4 AND gates, 1 OR gate and 3 NOT gates) are required. If the expression (*1) is changed to $\boxed{\quad B \quad}$, the number of logic gates required can be reduced to 6.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\qquad}$ in the above description.

Answer group for A

a) $(I_1 \bullet \neg I_2) + (I_1 \bullet \neg I_3) + (I_2 \bullet \neg I_3)$
b) $(I_1 \bullet \neg I_2) + (I_1 \bullet \neg I_3) + (\neg I_2 \bullet I_3)$
c) $(I_1 \bullet \neg I_2) + (\neg I_1 \bullet I_3) + (I_2 \bullet \neg I_3)$
d) $(I_1 \bullet \neg I_2) + (\neg I_1 \bullet I_3) + (\neg I_2 \bullet I_3)$

Answer group for B

a) $\neg(I_1 + I_2 + I_3) + (I_1 + I_2 + I_3)$
b) $\neg(I_1 + I_2 + I_3) + (I_1 \bullet I_2 \bullet I_3)$
c) $\neg(I_1 \bullet I_2 \bullet I_3) + (I_1 + I_2 + I_3)$
d) $\neg(I_1 \bullet I_2 \bullet I_3) + (I_1 \bullet I_2 \bullet I_3)$

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

There are three electric devices: TV, boiler and iron. The power consumption of the boiler is 1,000W, the iron is 1,500W, and the TV is 600W.
The house owner is going to configure an electric power control system that issues a warning when simultaneous use of electric power exceeds 2,000W.

Table 4 shows a truth table for a system that has three input lines (I_B, I_I and I_T) and one output line (Out). I_B, I_I and I_T indicate the on/off status of the boiler, iron and TV respectively. Value 0 indicates the device is turned off, and value 1 indicates the device is turned on. The truth table in Table 4 outputs value 1 when simultaneous use of the electric power exceeds 2,000W. Otherwise, it outputs value 0.

Table 4 Truth table for the system

I_B	I_I	I_T	Total(W)	Out
0	0	0	0	0
0	0	1	600	0
0	1	0	1,500	0
0	1	1	2,100	1
1	0	0	1,000	0
1	0	1	1,600	0
1	1	0	2,500	1
1	1	1	3,100	1

Table 5 Karnaugh map for the system

		I_I, I_T			
		0, 0	0, 1	1, 1	1, 0
I_B	0	0	0	$\boxed{\text{C}}$	
	1	0	0		

Note: grouping result is not shown

Table 5 shows the Karnaugh map that is equivalent to the truth table in Table 4. Because Table 5 has three cells that have output value 1, those cells are grouped into two sets: a set of two adjacent cells, and a set of a single cell. As a result, one of the single logic expression obtained from Table 5 is: $\boxed{\text{D}}$ + ($I_B \bullet I_I \bullet \neg I_T$)

Answer group for C

a)

0	1
1	1

b)

1	0
1	1

c)

1	1
0	1

d)

1	1
1	0

Answer group for D

a) ($I_B \bullet I_I$)
b) ($\neg I_B \bullet I_I$)
c) ($I_I \bullet I_T$)
d) ($I_I \bullet \neg I_T$)

Q3. Read the following description concerning a database for a car-rental service, and then answer Subquestions 1 and 2.

Company U, a car-rental service company, has various types of cars such as saloon, wagon, mini-bus, and 45-seater bus. Its office is located at the center of Capital city.
For each rental trip, Company U provides a driver together with a car including fuel.

Company U has a booking department that customers can contact to make a car rental. Customers give the following information to the booking department when they make a booking for the car-rental service:

- Type of car, or number of persons
- Start date and end date they want to use the service
- Area they want to go to

Examples of booking information:

- Company K makes a booking for a 45-seater bus from May 1 to May 4, 2016 (4 days) for their staff vacation trip to City L.
- Mr. M makes a booking for 4 persons from May 5 to May 6, 2016 (2 days) for the vacation trip to Beach N.

With these data, the booking staff sends an inquiry to the database to select an available car with the required capacity (number of persons). He also has to select an available driver who has the driving experience both with the selected car and in the area to go.

Company U provides the database management system with the following tables:

Table: Car
 Fields: CarNo, CarName, CarTypeID, Capacity, CarStatus
Table: CarType
 Fields: CarTypeID, CarTypeName
Table: Area
 Fields: AreaID, AreaName
Table: Driver
 Fields: DriverID, DriverName, CarLicenseNo
Table: DriverCarType
 Fields: DriverID, CarTypeID, CarExperience
Table: DriverArea
 Fields: DriverID, AreaID, AreaExperience
Table: TripHistory
 Fields: TripID, CarNo, BookingDate, DriverID, CustomerName, AreaID, StartDate, EndDate, Capacity, Fees, TripStatus

Description of the main fields:

CarNo	Car-license plate number
CarTypeID	1: Saloon, 2: Wagon, 3: SUV, 4: Mini-bus, 5: 45-seater bus, …
Capacity	Number of persons the car can carry
CarStatus	1: Ready to rent, 2: Maintenance (not ready to rent)
AreaID	1: Capital city, 2: City L, 3: Beach N, 4: Northern states, …
CarExperience	1 to 10: the larger the number, the greater the experience
AreaExperience	1 to 10: the larger the number, the greater the experience
TripID	Automatically assigned unique number
BookingDate	Date on which car-rental booking is made
StartDate	Date on which car rental starts
EndDate	Date on which car rental ends
Fees	Car-rental fees the customer must pay
TripStatus	1: Booked, 2: Finished, 3: Canceled

A driver may drive two or more types of cars and may also drive in two or more different areas. A driver who has finished a trip takes a day off the next day.

The table DriverCarType records what types of cars a driver can drive, and the table DriverArea records the experience of a driver in an area.

The booking staff selects the available car
- that is not in maintenance status
- that is ready to rent on the specific period from the start date to the end date
- whose capacity is greater than or equal to what the customer wants

and, selects an available driver who
- has an experience of driving the selected car
- has experience driving in the selected area
- has not already been assigned to another trip on the specific period
- is not on a day off

and then, adds a new record to the table TripHistory with the trip status “1” (booked).

The trip status of the record in the table TripHistory is set to
- “1” (booked) when the booking staff adds the new record to the table
- “2” (finished) when the car-rental service for the customer is finished
- “3” (canceled) if the customer cancels the booking he/she has previously made

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in Figure 3.

The booking staff makes a request to the booking department to provide a service report that lists the drivers who are (were, will be) actually working on a specific date together with the car name and the area name. Figure 1 shows an example of the service report.

TripID	DriverName	CarName	AreaName
1604141	Mr. Y. Young	Silver Wagon #5	Capital City
1604592	Mr. T. Tall	White Mini-bus #2	Northern States
……			

Figure 1 Example of the service report

The booking department first develops an SQL statement shown in Figure 2 that creates the view vTrip.

```
CREATE VIEW vTrip AS
  SELECT t.TripID, d.DriverName, c.CarName, a.AreaName,
         t.TripStatus, t.StartDate, t.EndDate
  FROM ((TripHistory t JOIN Driver d ON t.DriverID = d.DriverID)
                       JOIN Car    c ON t.CarNo    = c.CarNo   )
                       JOIN Area   a ON t.AreaID   = a.AreaID
```

Figure 2 SQL statement that creates the view vTrip

Using the view vTrip, the booking department then develops an SQL statement shown in Figure 3 that creates the service report.
Here, the specific date is given by the host variable `:SpDate`.

```
SELECT TripID, DriverName, CarName, AreaName
FROM vTrip
WHERE [      A      ]
  AND [      B      ]
```

Figure 3 SQL statement that creates the service report

Answer group for A

a) `:SpDate BETWEEN StartDate AND EndDate`
b) `StartDate < :SpDate AND :SpDate < EndDate`
c) `StartDate < :SpDate OR :SpDate < EndDate`
d) `StartDate = :SpDate OR :SpDate = EndDate`

Answer group for B

a) `TripStatus <> 1` b) `TripStatus <> 2` c) `TripStatus <> 3`

d) `TripStatus = 1` e) `TripStatus = 2` f) `TripStatus = 3`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank [ ] in Figure 5.

The booking staff receives a telephone call from a customer. The customer wants to make a booking for a saloon from May 7 to May 9, 2016 (3 days), for the business trip to City L.

The booking staff needs to obtain an experience report that lists the drivers who have the experience in both driving in City L and driving a saloon, and who are ready to work from May 7 to May 9, 2016, in descending order of area experience. Figure 4 shows an example of the experience report.

DriverName	AreaExperience	CarExperience
Mr. Y. Young	9	6
Ms. C. Charm	7	7

……

Figure 4 Example of the experience report

The booking department develops an SQL statement, as shown in Figure 5, that creates the experience report. Here, it is assumed that dates are expressed in the format 'yyyy-mm-dd', and it is always guaranteed that StartDate ≤ EndDate.

```
SELECT DISTINCT d.DriverName, a.AreaExperience, c.CarExperience
FROM (Driver d JOIN DriverArea    a ON d.DriverID = a.DriverID)
               JOIN DriverCarType c ON d.DriverID = c.DriverID
WHERE d.DriverID [    C    ] (SELECT DriverID FROM TripHistory AS t
                              WHERE t.StartDate [    D    ]
                                AND t.EndDate   [    E    ]
                                AND t.TripStatus [    F    ] )
      AND a.AreaID = 2
      AND c.CarTypeID = 1
ORDER BY a.AreaExperience DESC
```

Figure 5 SQL statement that creates the experience report

Answer group for C

a) `EXISTS`
b) `IN`
c) `NOT EXISTS`
d) `NOT IN`

Answer group for D and E

a) `< '2016-05-09'`
b) `< '2016-05-10'`
c) `< '2016-05-11'`
d) `> '2016-05-05'`
e) `> '2016-05-06'`
f) `> '2016-05-07'`

Answer group for F

a) `<> 1`
b) `<> 2`
c) `<> 3`
d) `= 1`
e) `= 2`
f) `= 3`

Q4. Read the following description concerning a distributed file system, and then answer Subquestions 1 and 2.

A distributed file system (hereinafter, DFS) allows files across a network to be shared by different clients using a network of servers and workstations. The location of the file becomes transparent to the client/workstation while the server keeps track of the files with each workstation. A DFS ensures the integrity of the file even when several clients open the same file and do updates. A remote-service mechanism is one methodology to maintain file integrity. Figure 1 presents the steps of the remote-service mechanism.

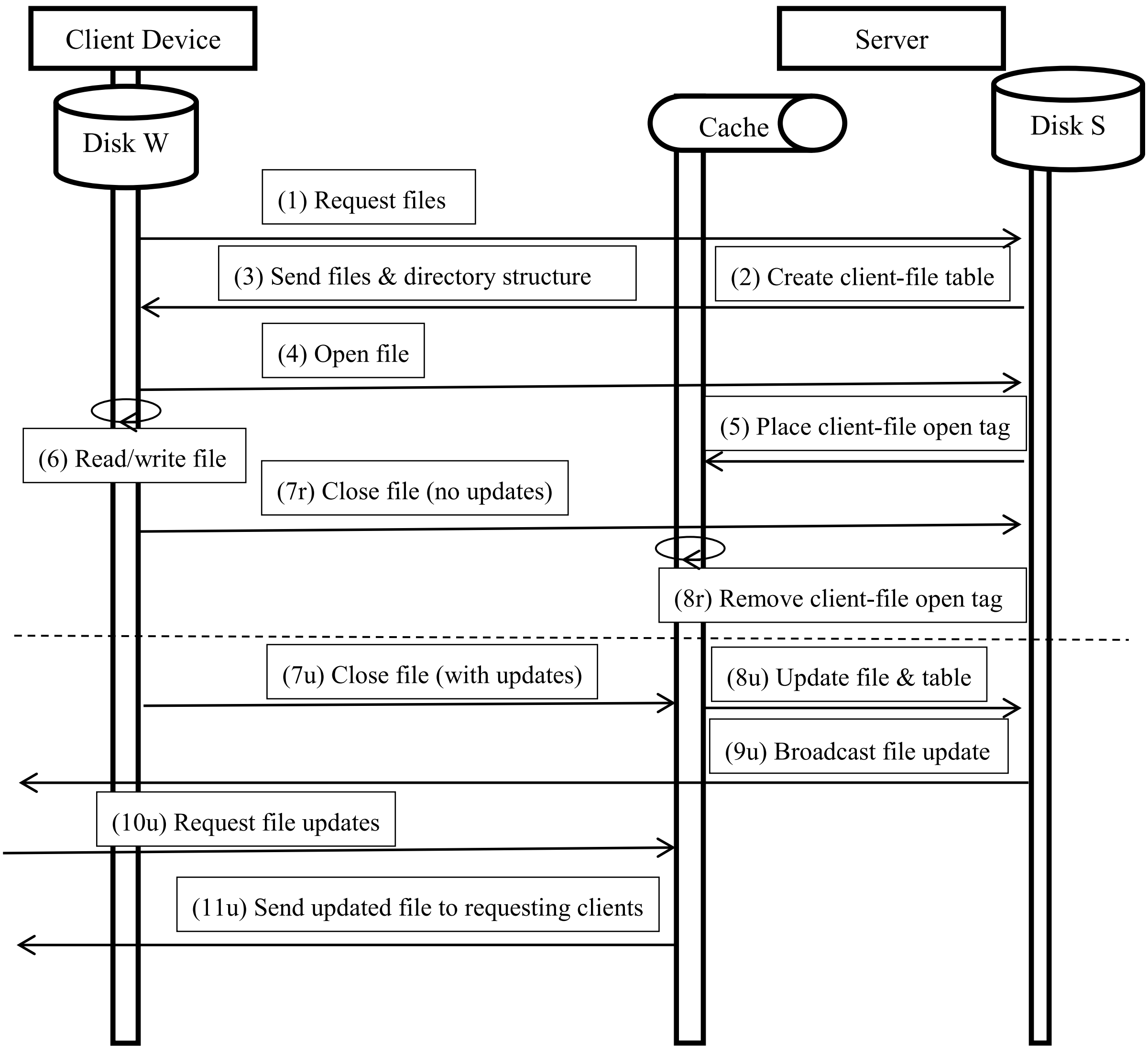


Figure 1 Remote-service mechanism

[Remote-Service mechanism]

(1) The workstation sends a request to the server to acquire files it needs.

(2) The server creates a client-file table in the cache to keep track of what files the client has.

(3) The server sends the files and directory structure to the workstation. If there was a previous connection, and the workstation already has the files in Disk W, only the updates are sent. The cache contains a partial copy of what is in Disk S including files and client-file tables.

(4) Once a workstation opens a file, notification is sent to the server.

(5) The server places an open tag in the client-file table. If there are updates to the file, they are sent to the workstation.

(6) The workstation is at liberty to do file operations on its end without informing the server.

In the case where the file is not updated:

(7r) Closing of the file (no update) is sent to the server.

(8r) The server removes the open tag.

In the case where the file is updated:

(7u) Closing of the file (with update) is sent to the server.

(8u) The cache and Disk S are updated. An update tag is placed in the client-file table.

(9u) A broadcast of the file update is sent to all clients with the file open.

(10u) Clients request for the file updates.

(11u) The server sends the updated file.

When an administrator makes changes to the directory structure, a broadcast is made and all client sessions will update their copies.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the following description.

Sales agents make use of a client device with a product information application. The application retrieves product files from the server. No updates are done by the client device. Each device can hold a limited amount of files. The server uses [A] to know the files each device has. In the event that some of the files are updated by the server, the [B] is used to broadcast that a file update has occurred. This informs the client devices that their copy is no longer vaild.

A workstation with an empty Disk W, which can occur for a new workstation or after a workstation crash, will retrieve files and directory structure from the server. As the number of workstations with an empty Disk W increases and the files they use are the same, both the [C] and [D] will increase, due to the process of step [E] in Figure 1.

Answer group for A and B

a) Cache (Server)
b) client-file table
c) Disk S
d) Disk W
e) open tag on the client-file table
f) update tag on the client-file table

Answer group for C and D

a) broadcasting operation
b) hit ratio of cache
c) network traffic
d) operation of placing update tag
e) read access to Disk W
f) write access to Disk S

Answer group for E

a) (1)
b) (3)
c) (4)
d) (11u)

Subquestion 2

[Lost Update problem]

At the time that a workstation is modifying the files, no communication with the server occurs. Only after the file is closed, the changes take effect on the server and cascaded to the other workstations. When two or more workstations change the same file and close it at the same time, only one will be processed by the server at any given time. The server broadcasts the update and the next workstation's update is received for processing. The design assumes that the workstation will consolidate the broadcasted update with its own update before closing the file. When the number of clients using the same file increases and the amount of updates is large of each one, updates are lost or overwritten by succeeding file close/update.

From the answer group below, select the appropriate description that will minimize the risk of a lost update.

Answer group

a) Different servers will have copies of the same file to distribute the file across the network, and one server will serve to consolidate the different copies.
b) Increasing the size of cache will increase the throughput of the server, but it will not change the network traffic performance.
c) The server's cache will accept the updates one at a time, and do the consolidation of updates instead of the workstation.
d) Workstations frequently close files after every small updates and re-open the file, even though this increases network traffic for large files.

Q5. Read the following description concerning network configurations, and then answer Subquestions 1 and 2.

Figure 1 shows the current network configuration of Company X. The network is connected to the Internet through an ADSL modem with a leased-line. Local IP addresses 192.168.1.0/24 are used for the internal network. In the internal network, a LAN and a wireless LAN are configured. A wireless access point supports many modern security standards such as encryption with 128-bit keys. For switching, a layer-2 switch is used.

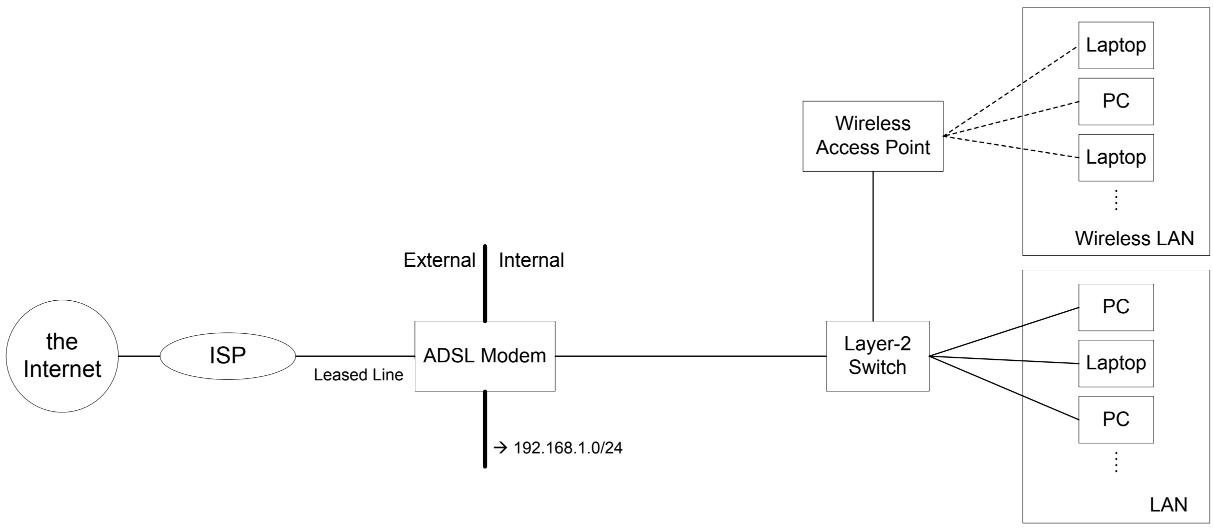


Figure 1 Current network configuration

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the following description.

With the current network configuration in Figure 1, the network address of the LAN is [A], and a maximum of [B] local IP addresses can be assigned to the PCs and laptops within the LAN and wireless LAN. Here, the ADSL modem uses local IP address 192.168.1.1, and the wireless access point uses local IP address 192.168.1.2

Answer group for A

a) 192.168.1.0	b) 192.168.1.1
c) 192.168.1.2	d) 192.168.1.255

Answer group for B

a) 251 b) 252 c) 253 d) 254

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

With the expansion of business, Company X decided to enhance the network configuration. Figure 2 shows the planned network configuration of Company X based on the following design principles:

- Place all laptops within the wireless LAN.
- Place all PCs and new internal-use servers within the LAN.
- Change the Layer-2 switch to a Layer-3 switch.
- Between the ADSL modem and Layer-3 switch, configure a network that has global IP addresses 203.0.113.240/29, and place new external-use servers within this network.

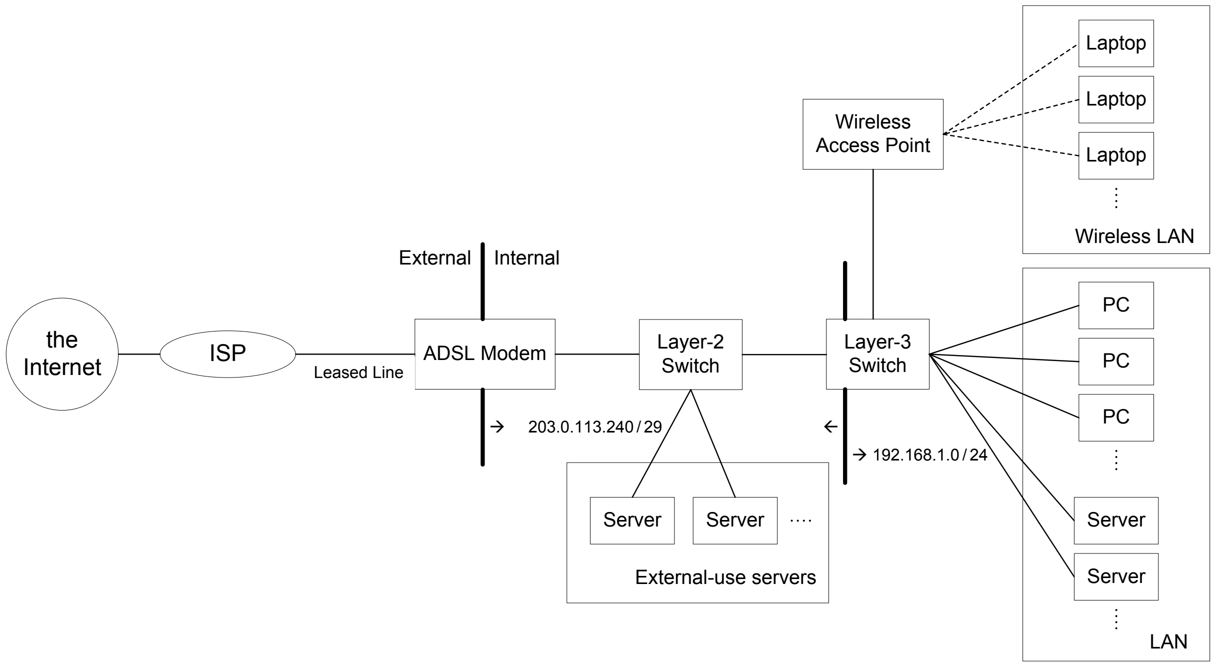


Figure 2 Planned network configuration

In this planned network configuration, Company X can use a maximum of $\boxed{\quad C \quad}$ global IP addresses for the external-use servers. Here, the ADSL modem and the Layer-3 switch will use one global IP address each.

Currently, to use Wireless LAN safely, Company X uses WPA (Wi-Fi Protected Access), and a user enters a pass-phrase when connecting to the wireless network.

Company X is now investigating the use of more secure WPA2. WPA2 has two modes: WPA2-Personal (or called WPA2-PSK) and WPA2-Enterprise (or called WPA2-EAP).
WPA2-Personal provides encryption of network traffic, and it does not require an authentication server.
WPA2-Enterprise provides enterprise-grade authentication, and it requires an authentication server.

If WPA2-Personal is used, the authentication method is the same as the current one, and the risk of [D] still remains.
If WPA2-Enterprise is used, the level of security will be improved, but the installation of [E] server is required.

Finally, Company X decided to use WPA2-Enterprise for wireless access authentication.

Answer group for C

a) 4	b) 6	c) 12	d) 14

Answer group for D

a) collision of packets
b) interference of radio wave
c) leakage of pass-phrase
d) leakage of radio wave

Answer group for E

a) DHCP
b) DNS
c) FTP
d) RADIUS

Q6. Read the following description of programs and the programs themselves, and then answer Subquestions 1 through 3.

The integer function `BitTest` checks the bit values in the specified bit positions in an 8-bit data, and then returns the result. The integer function `BitCount` returns the number of bits that have a value of 1 in an 8-bit data.
Here, in this question, the operators "&" and "`|`" obtain the logical product and logical sum, respectively, for each pair of bits in the corresponding bit positions of two 8-bit logical-type data, and obtain an 8-bit logical-type result. The notation "........"`B` expresses an 8-bit logical-type constant.

[Description of Program 1]
The integer function `BitTest` is declared as follows:

```
○ Integer function:
      BitTest (8-bit Logical type: Data, 8-bit Logical type: Mask)
```

The 8-bit data to be checked are stored in the input argument `Data`, and the bit position information to be checked is stored in the input argument `Mask`. Bits in `Data` corresponding to the bit positions that have value of 1 in `Mask` are checked, and the following return value is returned. Here, in `Mask`, there is at least one bit that has a value of 1.

Return value 0: All checked bits are 0's
1: 0 and 1 are mixed in the checked bits
2: All checked bits are 1's

For example, in Example 1 in Figure 1, the 3 bits (bit numbers 7 to 5) in `Mask` are 1's; thus, the values of the 3 bits (bit numbers 7 to 5) in `Data` are checked; because 0 and 1 are both included in these, the return value of 1 is returned. In Example 2, the 2 bits (bit numbers 4 and 0) in `Mask` are 1's; thus, the values of the 2 bits (bit numbers 4 and 0) in `Data` are checked; because both are 1's, the return value of 2 is returned.

(Example 1)

Bit number	7	6	5	4	3	2	1	0
`Data`	0	1	0	1	0	1	0	1
	|	|	|					
`Mask`	1	1	1	0	0	0	0	0

Return value 1

(Example 2)

Bit number	7	6	5	4	3	2	1	0
`Data`	0	0	1	1	0	0	1	1
				|				|
`Mask`	0	0	0	1	0	0	0	1

Return value 2

Figure 1 Examples of the `BitTest` operation

[Program 1]

```
○ Integer function:
       BitTest (8-bit Logical type: Data, 8-bit Logical type: Mask)
○ Integer type: RC     /* return value */
▲   A
│ • RC ← 2         /* return value is 2 */
├──
│ ▲   B
│ │ • RC ← 0       /* return value is 0 */
│ ├──
│ │ • RC ← 1       /* return value is 1 */
│ ▼
▼
• return RC        /* returns RC as the return value */
```

[Description of Programs 2 and 3]

The integer function `BitCount` is declared as follows:

```
○ Integer function: BitCount (8-bit Logical type: Data)
```

The 8-bit data to be checked are stored in the input argument `Data`.

Program 2, which uses a basic algorithm, and Program 3, which emphasizes efficiency of processing, are created as programs for this purpose.

For each line of Programs 2 and 3, the required processing time in a certain time unit (1, 2, …) when that line of the program is executed once on a certain processing system is shown. Assume that the processing time for the last lines of the selection process and iteration processes is included in the processing time for their respective first lines.

Here, the operator "–" performs subtraction by treating both operands as 8-bit unsigned integers.

[Program 2]

(Processing time)

```
     ○ Integer function: BitCount (8-bit Logical type: Data)
     ○ 8-bit Logical type: Work
     ○ Integer type: Count, Loop
 1   • Work ← Data
 1   • Count ← 0
 4   ■ Loop: 0, Loop < 8, 1
 3   │ ▲ rightmost bit of Work is 1
 1   │ │ • Count ← Count + 1
     │ ▼
 1   │ • logically shifts Work 1 bit to the right
     ■
 2   • return Count            /* returns Count as the return value */
```

[Program 3]

(Processing time)

```
       ○ Integer function: BitCount (8-bit Logical type: Data)
       ○ 8-bit Logical type: Work
       ○ Integer type: Count
  1    • Work ← Data
  1    • Count ← 0
  2    ■ bits that have value 1s exist in Work
  1    │ • Count ← Count + 1
  3    │ • Work ← Work & (Work − 1)  ←———————— α
       ■
  2    • return Count              /* returns Count as the return value */
```

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank $\boxed{\qquad\qquad}$ in Program 1.

Answer group for A and B

a) `(Data & Mask) = "00000000"B`

b) `(Data & Mask) = Data`

c) `(Data & Mask) = Mask`

d) `(Data | Mask) = "00000000"B`

e) `(Data | Mask) = Mask`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank $\boxed{\quad\quad}$ in the following description.

Program 1 presumes that there is at least one bit that has a value of 1 in `Mask`. Here, this presumption is removed, and the goal is to return a value of 0 when all the bits in `Mask` have a value of 0. To achieve this, the following change proposals, (1) to (3), are made to replace the processing part of Program 1. Note that change proposal (1) makes no change to Program 1. Here, the correct answers for Subquestion 1 are assumed to be inserted in blanks $\boxed{A}$ and $\boxed{B}$.

Change proposal (1)
(no change)

```
▲ [ A ]
│ • RC ← 2
├──────
│ ▲ [ B ]
│ │ • RC ← 0
│ ├──────
│ │ • RC ← 1
│ ▼
▼
• return RC
```

Change proposal (2)

```
▲ [ B ]
│ • RC ← 0
├──────
│ ▲ [ A ]
│ │ • RC ← 2
│ ├──────
│ │ • RC ← 1
│ ▼
▼
• return RC
```

Change proposal (3)

```
• RC ← 1
▲ [ B ]
│ • RC ← 0
▼
▲ [ A ]
│ • RC ← 2
▼
• return RC
```

Of these change proposals, the one that operates correctly is $\boxed{C}$.

Answer group for C

a) Change proposal (1)
b) Change proposal (2)
c) Change proposal (3)

Subquestion 3

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the following description.

The processing efficiency of Programs 2 and 3 is considered. Table 1 shows the results of the comparison of the processing time of Programs 2 and 3.

Table 1 Comparison of the processing time of Programs 2 and 3

	Minimum	Maximum
Program 2	72	80
Program 3	D	54

In Program 3, an efficient algorithm is used to update the value of the variable `Work` in the line **α**. For example, when the content of the argument `Data` in Program 3 is `"01101010"B`, the content of the variable `Work`, at the time that the process in the line **α** finishes its second execution during the iteration process, is "[E]"`B`. Through the process of bit conversion in this manner, the number of repetitions in the iteration process is the same as the number of bits that have a value of 1 in the data to be checked.

Answer group for D

a) 6 b) 10 c) 20 d) 22

Answer group for E

a) `00000011` b) `00000110` c) `00001010`
d) `01010000` e) `01100000` f) `10100000`

Concerning questions **Q7** and **Q8**, **select one** of the two questions. Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question. If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestion.

[Program Description]

A palindrome is a sequence of one or more characters that read the same from the left as it does from the right. For example, Z, EVE and ABBA are palindromes, but ADAM is not. The program reads a positive integer with at most 9 digits, and determines the number of unique integer palindromes. The unique integer palindromes must be arranged in terms of the number of digits and in ascending order.

(1) Table 1 shows some examples of a positive integer and the unique integer palindromes.

Table 1 Examples of a positive integer and the unique integer palindromes

Positive integer	Unique integer palindromes
1230	0, 1, 2 and 3
12103	0, 1, 2, 3 and 121
11221	1, 2, 11, 22 and 1221
4123214	1, 2, 3, 4, 232, 12321 and 4123214

(2) The algorithm used in this program is as follows:
Step 1: Ask the user to input a positive integer.
Step 2: Count the number of digits in the inputted positive integer.
Step 3: Extract all unique integer palindromes.
Step 4: Sort the integer palindrome list in ascending order and display the result.

(3) When executing this program, it will print out the following lines:

```
Enter a positive integer: 112233221
There are 9 unique palindromes.
[ 1, 2, 3, 11, 22, 33, 2332, 223322, 12233221 ]
```

(4) Five user defined functions are used:

(i) `int numOfDigits(long number)`
This function returns the number of digits in `number`.

(ii) `long reverseDigits(long number)`
This function returns a number with its digits in reversed order of `number`.

(iii) `int isExisting(long pList[], long number, int pCount)`
This function returns 1 if an integer `number` already exists in the integer palindrome list `pList`; otherwise, it returns 0. Here, `pCount` is the array length of `pList`.

(iv) `void extractNumber(long number, int length,`
`long *pList, int *pCount)`
This function extracts an integer palindrome from a positive integer number and adds this integer to the palindrome list `pList`.

(v) `void displayPalindromes(long pList[], int pCount)`
This function sorts the integer palindrome list `pList` in ascending order and displays all the integer palindromes in `pList`.

[Program]

```
#include <stdio.h>
#define MAXLEN 20

int  numOfDigits(long);
long reverseDigits(long);
int  isExisting(long*, long, int);
void displayPalindromes(long*, int);
void extractNumber(long, int, long*, int*);

int main(void) {
   int  pCount = 0, length = 0, i;
   long input, pList[MAXLEN];

   for(i=0; i< MAXLEN; i++) {
      pList[i] = -1;
   }
   printf("Enter a positive integer: ");
   scanf("%ld", &input);
   length = numOfDigits(input);
   extractNumber(input, length, [     A     ], &pCount);
   if (pCount > 1)
      printf("\nThere are %d unique palindromes.", pCount);
   else
      printf("\nThere is %d unique palindrome.", pCount);
   displayPalindromes([     A     ], pCount);
   return 0;
}

int numOfDigits(long number) {
   int  length = 0;

   while ([     B     ]) {
      number /= 10;
      length++;
   }
   return length;
}
```

```
long reverseDigits(long number) {
   long reversedNum = 0;

   while ([    B    ]) {
      reversedNum = (reversedNum * 10) + (number % 10);
      number /= 10;
   }
   return reversedNum;
}

int isExisting(long* pList, long num, int pCount) {
   int  i;

   for (i = 0; [    C    ]; i++)
      if(pList[i] == num || num == -1)
         return 1;
   return 0;
}

void displayPalindromes(long* pList, int pCount) {
   long hold;
   int  count, pass;

   for (pass = 1; pass < pCount; pass++) {
      for (count = 0; count < pCount - 1; count++) {
         if ([    D    ]) {
            hold = pList[count];
            [    E    ];
            pList[count+1] = hold;
         }
      }
   }
   printf("\n[");
   for (count = 0; count < pCount; count++) {
      if (count != pCount - 1)
         printf(" %ld,", pList[count]);
      else
         printf(" %ld", pList[count]);
   }
   printf(" ]");
}
```

```
void extractNumber(long number, int length,
                   long *pList, int *pCount) {
   int  i, j, k, l, index = 0;
   long tempNum, extractedNum, extractedRNum;

   for (i = 1; i <= length; i++) {
      for (j = 0, tempNum = number; j < length - i + 1; j++) {
         extractedNum = 0;
         extractedRNum = 0;
         tempNum = number;
         for (k = 0, l = 0; k < length && l < i; k++) {
            if (k >= j) {
               extractedNum = [    F    ];
               l++;
            }
            tempNum /= 10;
         }
         extractedRNum = reverseDigits(extractedNum);
         if (extractedNum == extractedRNum) {
            if (!isExisting([    A    ], extractedNum, *pCount)) {
               pList[index] = extractedNum;
               index++;
               *pCount = *pCount + 1;
            }
         }
      }
   }
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A

a) `&pList[20]` b) `*pList[0]` c) `**pList`
d) `pList` e) `pList[0]` f) `pList[20]`

Answer group for B

a) `number / 10 != 0` b) `number / 10 < 0` c) `number / 10 == 0`
d) `number != 0` e) `number < 0` f) `number == 0`

Answer group for C

a) `i <= pCount` b) `i < pCount` c) `i < pCount - 1`
d) `i <= pList[i]` e) `i < pList[i]` f) `i < pList[i] - 1`

Answer group for D

a) `pList[count] > pList[count+1]`
b) `pList[count] != pList[count+1]`
c) `pList[count] == pList[count+1]`
d) `pList[count] > pList[count-1]`
e) `pList[count] != pList[count-1]`
f) `pList[count] == pList[count-1]`

Answer group for E

a) `pList[count] = pList[count+1]` b) `pList[count] = pList[count+2]`
c) `pList[count] = pList[count-1]` d) `pList[count] = pList[count-2]`

Answer group for F

a) `(extractedNum % 10) + (tempNum * 10)`
b) `(extractedNum % 10) + (tempNum / 10)`
c) `(extractedNum * 10) + (tempNum % 10)`
d) `(extractedNum * 10) + (tempNum / 10)`
e) `(extractedNum / 10) + (tempNum % 10)`
f) `(extractedNum / 10) + (tempNum * 10)`

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestions 1 through 3.

[Program Description]

A system development team of a university is building a prototype for a university information system. The system manages SMS, e-mail and allowance information of the students and staffs of the university. The prototype is being analyzed and debugged using tester classes with methods for testing purposes. These methods are invoked from the main method in the tester classes. As for some statements in the main method, they may

- be executed properly by invoking the appropriate methods through polymorphism
- cause a runtime exception
- cause compile errors

So, the output, exceptions and errors need to be determined.

In the programs, there are four classes: `Person`, `Student`, `Staff` and `PostGradStudent`. Figure 1 shows that `Student` and `Staff` inherit from `Person` and that `PostGradStudent` inherits from `Student`.

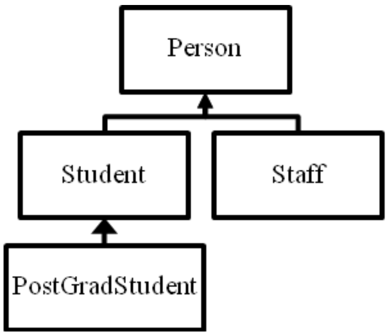


Figure 1 Inheritance relation

[Program 1]

```
public class Person {
    static double LIVING_COST = 1000.0;
    double allowance;
    String cellPhoneNumber, emailAddresses[];

    Person() {
        this("0123456", LIVING_COST);
        //0123456 is the contact number for information desk
    }
    Person(String cellPhoneNumber, double allowance) {
        [    A    ].cellPhoneNumber = cellPhoneNumber;
        [    A    ].allowance = allowance;
    }
    public void setEmail(String email1){
        emailAddresses = [    B    ] { email1 };
    }
    public void setEmail(String email1, String email2){
        emailAddresses = [    B    ] { email1, email2 };
    }
    public void setEmail(String email1, String email2, String email3){
        emailAddresses = [    B    ] { email1, email2, email3 };
    }
    public void sendSMS() {
        System.out.println(this + "Person sendSMS");
        sendEmail();
    }
    public void sendEmail() {
        System.out.println(this + "Person sendEmail");
    }
    public String toString(){
        return "I am Person: ";
    }
}
```

[Program 2]

```
public class Student extends Person {
    public void sendEmail() {
        System.out.println(this + "Student sendEmail");
    }
    public void depositAllowance() {
        System.out.println(this + "Student depositAllowance");
        allowance += LIVING_COST;
    }
}
```

[Program 3]

```
public class Staff extends Person {
   public void sendSMS() {
      System.out.println(this + "Staff sendSMS");
   }
   public void sendEmail() {
      System.out.println(this + "Staff sendEmail");
      sendSMS();
   }
   public String toString(){
      return "I am Staff: ";
   }
}
```

[Program 4]

```
public class PostGradStudent extends Student {
   public void sendSMS() {
      super.sendSMS();
      System.out.println(this + "PostGradStudent sendSMS");
   }
   public void sendEmail() {
      System.out.println(this + "PostGradStudent sendEmail");
   }
   public void depositAllowance() {
      System.out.println(this + "PostGradStudent depositAllowance");
      allowance += 2 * LIVING_COST;
   }
}
```

[Program 5]

```
public class Tester1 {
   public static void main(String[] args) {
      Person person1 = new Student();
      Person person2 = new PostGradStudent();
      Student student1 = new Student();
      Student student2 = new PostGradStudent();
      Staff staff1 = new Staff();
      Object obj1 = new Student();
      staff1.sendSMS();
      person1.sendEmail();
      person2.sendEmail();
      student1.sendEmail();
      student2.sendEmail();
      student1.depositAllowance();
      student2.depositAllowance();
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\quad\quad}$ in Program 1.

Answer group for A

a) `Object`
b) `Person`
c) `super`
d) `this`

Answer group for B

a) `new String[]`
b) `new String[3]`
c) `new String`
d) `new String ...`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{\quad\quad}$ in the following description.

The purpose of [Program 5] (class `Tester1`) is to check whether the programs are executed properly by invoking the appropriate methods through polymorphism.

When [Program 5] is executed, the following lines of output data are displayed:

```
I am Staff: Staff sendSMS
I am [  C  ]: [  D  ] sendEmail
I am [  C  ]: [  E  ] sendEmail
I am [  C  ]: [  D  ] sendEmail
I am [  C  ]: [  E  ] sendEmail
I am [  C  ]: [  D  ] depositAllowance
I am [  C  ]: [  E  ] depositAllowance
```

Answer group for C, D and E

a) `Person`
b) `PostGradStudent`
c) `Staff`
d) `Student`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank $\boxed{\quad\quad}$ in Table 1.

A person can be a `Student`, `PostGradStudent` or `Staff`. Based on the actual object, the behaviors of the methods `depositAllowance`, `sendEmail` and `sendSMS` may vary. So, to investigate the behavior of the methods, [Program 6] (class `Tester2`) is created.

[Program 6]

```
public class Tester2 {
   public static void main(String[] args) {
      Person person1 = new Student();
      Person person2 = new PostGradStudent();
      Student student1 = new Student();
      Student student2 = new PostGradStudent();
      Staff staff1 = new Staff();
      Object obj1 = new Student();
      [                    ]
   }
}
```

In the blank [ ] in [Program 6], one of the statements shown in Table 1 is inserted one at a time. In this way, [Program 6] is executed three times with a different statement in the blank.

Table 1 shows the statements to be inserted in the blank and their execution results.

Table 1 Statements to be inserted in the blank and their execution results

Statements to be inserted in the blank [ ]	Execution result
`((PostGradStudent)staff1).depositAllowance();`	F
`((Student)person2).depositAllowance();`	G
`((PostGradStudent)obj1).sendEmail();`	H

Answer group for F, G and H

a) The statement causes a compile error; hence the program is not executable.
b) The statement causes a runtime exception; hence the program terminates abnormally.
c) "`I am Person: PostGradStudent depositAllowance`" is displayed.
d) "`I am Person: PostGradStudent sendEmail`" is displayed.
e) "`I am Person: Staff depositAllowance`" is displayed.
f) "`I am Person: Staff sendEmail`" is displayed.
g) "`I am Person: Student depositAllowance`" is displayed.
h) "`I am Person: Student sendEmail`" is displayed.

Question No.	Subquestion No.		Correct Answer
1	1	A	d
		B	c
		C	a
		D	c
	2	E	d
2	1	A	c
		B	b
	2	C	b
		D	c
3	1	A	a
		B	c
	2	C	d
		D	c
		E	d
		F	c
4	1	A	b
		B	f
		C	b/c
		D	c/b
		E	b
	2		d
5	1	A	a
		B	b
	2	C	a
		D	c
		E	d
6	1	A	c
		B	a
	2	C	b
	3	D	a
		E	e
7	1	A	d
		B	d
		C	b
		D	a
		E	a
		F	c
8	1	A	d
		B	a
	2	C	a
		D	d
		E	b
	3	F	a
		G	c
		H	b

October 2015

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the Ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the autumn Fundamental IT Engineer Examination conducted?

   Answer group
   a) September   b) October   c) November   d) December

   Since the correct answer is " b) October ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated.

[Declaration, comment, and process]

<table>
<tr><th colspan="2">Notation</th><th>Description</th></tr>
<tr><td colspan="2">○</td><td>Declares names, types, etc. of procedures, variables, etc.</td></tr>
<tr><td colspan="2">/* text */</td><td>Describes comments in the text.</td></tr>
<tr><td rowspan="7">Process</td><td>• variable ← expression</td><td>Assigns the value of the expression to the variable.</td></tr>
<tr><td>• procedure(argument, ...)</td><td>Calls the procedure and passes / receives the argument.</td></tr>
<tr><td>▲ conditional expression
  process
▼</td><td>Indicates a one-way selection process.
If the conditional expression is true,
then the process is executed.</td></tr>
<tr><td>▲ conditional expression
  process 1
―
  process 2
▼</td><td>Indicates a two-way selection process.
If the conditional expression is true,
then the process 1 is executed.
If it is false, then the process 2 is executed.</td></tr>
<tr><td>■ conditional expression
  process
■</td><td>Indicates a pre-test iteration process.
While the conditional expression is true,
the process is executed repeatedly.</td></tr>
<tr><td>■
  process
■ conditional expression</td><td>Indicates a post-test iteration process.
The process is executed, and then
while the conditional expression is true,
the process is executed repeatedly.</td></tr>
<tr><td>■ variable: init, cond, incr
  process
■</td><td>Indicates an iteration process.
The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is true, the process is executed repeatedly.
The increment incr (given by an expression) is added to the variable in each iteration.</td></tr>
</table>

[Logical constants]

`true, false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	+, −, not	High
Multiplication, division	×, ÷, %	↑
Addition, subtraction	+, −	
Relational operation	>, <, ≥, ≤, =, ≠	
Logical product	and	↓
Logical sum	or	Low

Note: With division of integers, integer quotient is returned as a result.
The % operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description concerning email security, and then answer Subquestions 1 through 3.

Email (electronic mail) is the most heavily used network-based distributed application. All Internet users worldwide communicate with others who are connected directly or indirectly to the Internet via email, regardless of system architecture, host operating system, vendor platform, and communication suites.

PGP (Pretty Good Privacy) is an encryption program that provides authentication for data communication by adding a digital signature to an email message. To generate a digital signature from the email body, (1) the SHA-1 (Secure Hash Algorithm-1) is used to generate a 160-bit hash code of the message, and (2) the hash code is then encrypted with RSA algorithm using the sender's private key. Then, the result is appended to the message. To save bandwidth during email transmission and the space taken by the file, the ZIP algorithm having a compression ratio of approximately 2.0 is used to compress the message before it is sent.

Figures 1 and 2 show the message authentication operations at a sender site and a receiver site, respectively.

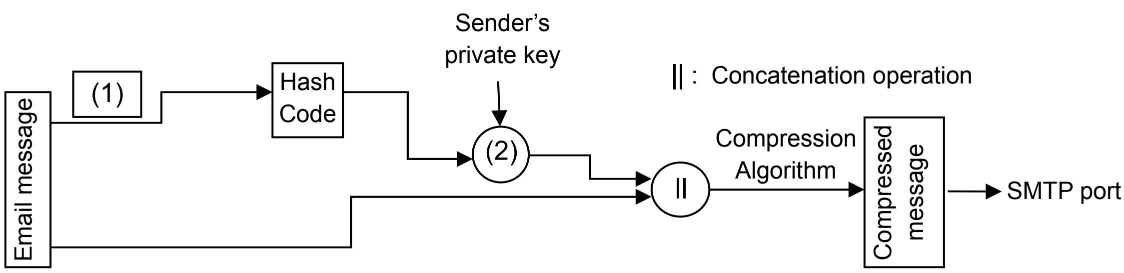


Figure 1. Message authentication operations at a sender site

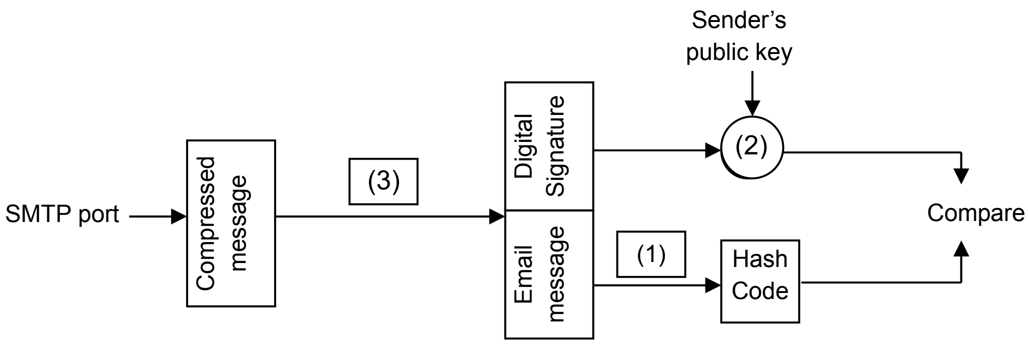


Figure 2. Message authentication operations at a receiver site

At the receiver site, the digital signature is separated from the original email message. RSA is used with the sender's public key to decrypt and recover the hash code.

The receiver generates a new hash code for the email message and compares it with the decrypted hash code. If the two codes match, the message is accepted as authentic.

The addition of a signature to an email ensures the possessor's authenticity, but does not ensure the confidentiality of the email. A symmetric encryption algorithm is used to encrypt email messages in 64-bit cipher feedback mode. In PGP, the symmetric key is used only once, and a new random 128-bit session key is generated for each message. An email is transmitted in the store and forward manner, and thus, using a handshake to ensure that both sides have the same session key is not practical. Because it is to be used only once, the key is bound to the message and transmitted with it. For protection, the session key is encrypted by using the receiver's public key.
Figure 3 shows the sequences of operations performed at a sender site and a receiver site.

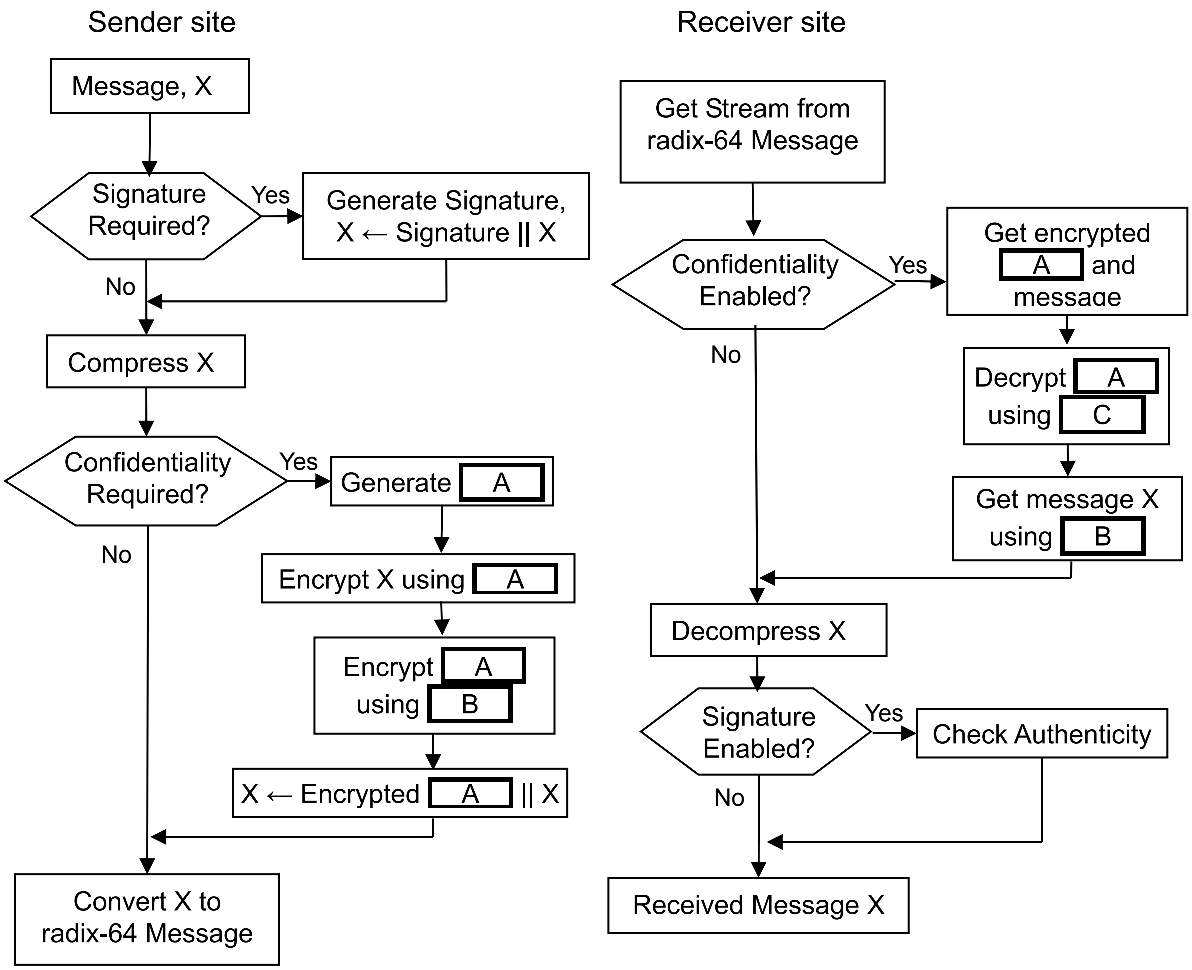


Figure 3. Operations at sender and receiver sites for ensuring both authentication and confidentiality of the email body

The digital signature is generated from the plaintext, rather than the encrypted or compressed message. This ensures that, for the purpose of signature verification, a third-party need not be concerned with the symmetric key or compression algorithm. Message encryption is applied after compression to strengthen cryptography security: because the compressed message has less redundancy than the original plaintext, crypt-analysis is more difficult.

The resulting encrypted message at the sender site consists of a stream of arbitrary 8-bit binary codes. However, many email systems permit the use only of blocks that consist of ASCII text. To accommodate this restriction, PGP provides a service that comprises converting an 8-bit binary stream to a stream of printable ASCII characters using radix-64 conversion. Each group of three 8-bit binary codes is mapped into four ASCII characters expanding the length of the message by 33%. Fortunately, the session key and signature portions of the message are relatively compact.

Subquestion 1

From the answer group below, select the correct combination of the answers for (1), (2), and (3) in Figures 1 and 2.

Answer group

	(1)	(2)	(3)
a)	Compression algorithm	RSA algorithm	ZIP algorithm
b)	RSA algorithm	Compression algorithm	SHA-1
c)	RSA algorithm	ZIP algorithm	Compression algorithm
d)	SHA-1	RSA algorithm	Compression algorithm
e)	SHA-1	RSA algorithm	Decompression algorithm
f)	ZIP algorithm	SHA-1	Decompression algorithm

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in Figure 3.

Answer group

a) 64-bit cipher key
b) Compression key
c) Hash key
d) Receiver's private key
e) Receiver's public key
f) Sender's private key
g) Sender's public key
h) Session key

Subquestion 3

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following description.

An email with a message size of 9 kB is transmitted enabling both authenticity and confidentiality. The ZIP algorithm is used to compress the message of the email. The size of the digital signature and encrypted session key is small and can be ignored. In this case, the size of the ASCII stream to be transmitted will be $\boxed{\text{D}}$ (in nearest kB).

Answer group

a) 4.5 kB

b) 6 kB

c) 9 kB

d) 12 kB

Q2. Read the following description concerning scheduling algorithms, and then answer Subquestion.

Modern multiprogramming-based operating systems allow more than one process to be loaded into the memory at a time. The loaded processes share the CPU among them through time multiplexing. In general, an operating system chooses a process for the execution based on a scheduling algorithm. The scheduling is in fact the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a certain strategy. Some of the most popular scheduling algorithms are:

- First Come First Serve (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling
- Round Robin(RR) Scheduling

Table 1 shows the descriptions of the FCFS, SJF, and RR scheduling algorithms.

Table 1. Descriptions of FCFS, SJF, and RR algorithms

Algorithm	Description
FCFS	Processes are executed on a first come, first serve basis. The process manager selects the process to be executed based on arrival time.
SJF	The process manager selects the process that has the shortest execution time among the waiting processes. In SJF, the scheduling is initiated when all the processes are ready for execution.
RR	The process manager provides a fixed time period, known as a quantum, for each process. When a process has been executed for the given quantum, the process is halted and the next process in the queue is chosen for execution for the given quantum. The performance of RR varies with the quantum and the number of processes in the queue.

Several criteria are applied to evaluate the performance of scheduling algorithms, such as waiting time, response time, and turnaround time. Table 2 shows the definitions of the waiting time, response time, and turn-around time.

Table 2. Definitions of waiting time, response time, and turn-around time

	Definition
Waiting time	The time the process has to wait before it receives a time slice for its execution.
Response time	The amount of time between the submission of a request and the first response (i.e., the time the process takes to start responding).
Turn-around time	The total of the waiting time and the processing time.

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\qquad}$ in the following description.

There are four processes, named P1, P2, P3, and P4. Table 3 shows the arrival time and processing time of each process. The processing time is the execution time required to complete the process. The arrival times in Table 3 are applicable to the FCFS and RR algorithms. The arrival times in Table 3 are not applicable to the SJF algorithm, because the scheduling starts when all the processes are ready for execution. Assume that processes P1 through P4 make their first response immediately after they start.

Table 3. Arrival time and processing time of each process

Process	Arrival time	Processing time
P1	0	5
P2	1	3
P3	2	8
P4	3	6

Figure 1 shows the execution sequence of the processes and Table 4 shows the execution results of the processes, in the case of the FCFS algorithm.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

P1	P2	P3	P4

Figure 1. Execution sequence of the processes in the FCFS algorithm

Table 4. Execution results of the processes in the FCFS algorithm

Process	Waiting time	Response time	Turn-around time
P1	0 =(0−0)	0 =(0−0)	5 =(0+5)
P2	4 =(5−1)	4 =(5−1)	7 =(4+3)
P3	A		
P4	13 =(16−3)	13 =(16−3)	19 =(13+6)

Figure 2 shows the execution sequence of the processes and Table 5 shows the execution results of the processes, in the case of the SJF algorithm.

|0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10 |11 |12 |13 |14 |15 |16 |17 |18 |19 |20 |21 |22

P2	P1	P4	P3

Figure 2. Execution sequence of the processes in SJF algorithm

Table 5. Execution results of the processes in SJF algorithm

Process	Waiting time	Response time	Turn-around time
P1	3	3	8
P2	0	0	3
P3	14	14	22
P4	B		

Figure 3 shows the execution sequence of the processes and Table 6 shows the execution results of the processes, in the case of the RR algorithm.
Here, the quantum used in the RR algorithm is 3.

|0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10 |11 |12 |13 |14 |15 |16 |17 |18 |19 |20 |21 |22

P1	P2	P3	C

Figure 3. Execution sequence of the processes in RR algorithm

Table 6. Execution results of the processes in RR algorithm

Process	Waiting time	Response time	Turn-around time
P1	D		
P2	2 =(3−1)	2 =(3−1)	5 = (2+3)
P3	12 =(6−2)+(14−9)+(20−17)	4 =(6−2)	20 =(12+8)
P4	11 =(9−3)+(17−12)	6 =(9−3)	17 =(11+6)

Answer group for A (Note: detailed expressions are not shown)

a)

2	2	10

b)

3	3	16

c)

6	6	14

d)

8	8	18

Answer group for B

a)

5	5	11

b)

8	8	14

c)

13	13	19

d)

16	16	22

Answer group for C (Note: select the correct sequence of the processes)

a)

P1	P2	P3	P4	P1

b)

P2	P1	P4	P3	P1

c)

P4	P1	P3	P4	P3

d)

P4	P3	P1	P3	P4

Answer group for D (Note: detailed expressions are not shown)

a)

3	14	22

b)

9	0	14

c)

12	0	14

d)

12	0	17

Q3. Read the following description concerning a gymnastic center's database, and then answer Subquestions 1 through 3.

Given the busy life of people nowadays, exercise is an essential activity for maintaining and improving health. Training at a gymnastic center does not consist of simply doing random exercises, but is planned based on a clear training strategy and effective training methods.

A gymnastic center introduced indexes to measure the progress of the training, but the data thus far have not been utilized effectively. Therefore, the gymnastic center decides to develop a gymnastic index management system to provide better services for its customers. Figure 1 shows an E-R diagram of the gymnastic index management system (incomplete), and Table 1 shows the description of each entity in Figure 1.

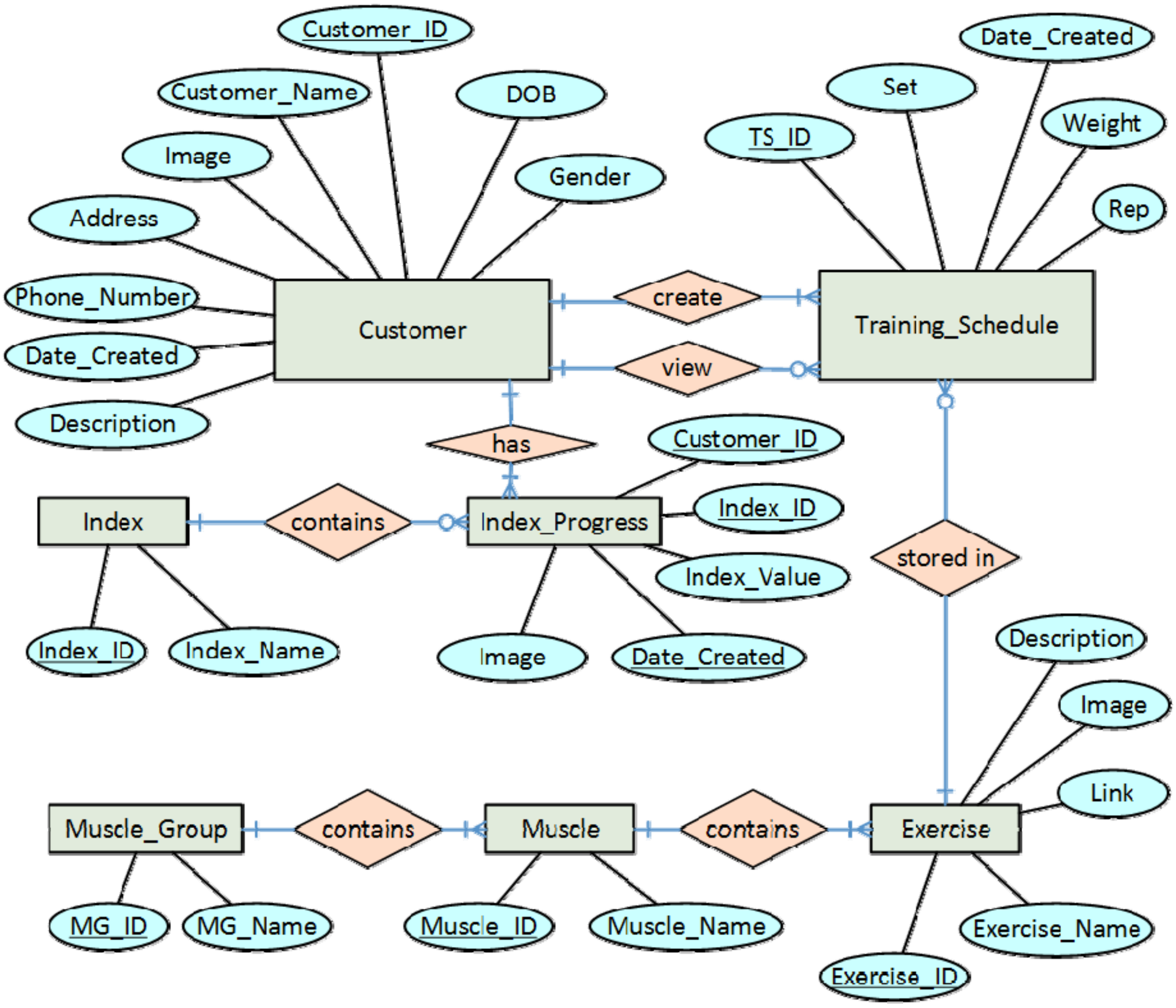


Figure 1. E-R diagram of the gymnastic index management system (incomplete)

Table 1. Description of each entity in Figure 1

Entity name	Description
Customer	Information about each registered customer.
Index	Name of each index.
Index_Progress	Index value and image of a customer at a difference time. A customer can monitor the changes in his/her index values and images.
Muscle_Group	Major muscle group in the human body. "Muscle Group" is abbreviated to "MG".
Muscle	Name of each muscle that belongs to the major muscle group.
Exercise	Information about each exercise that is associated with the muscle.
Training_Schedule	Training schedule of exercises for a customer. "Training Schedule" is abbreviated to "TS".

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [ ] in the following description.

Figure 1 is incomplete because it has four missing attributes which are primary keys.
In order to complete the E-R diagram, Figure 1 should be revised as follows:

(1) Add the attribute [A] to the entity Exercise.
(2) Add the attribute [B] to the entity Muscle.
(3) Add the attributes Customer_ID and Exercise_ID to the entity [C].

Answer group for A and B

a) Customer_ID
b) Exercise_ID
c) Index_ID
d) MG_ID
e) Muscle_ID
f) TS_ID

Answer group for C

a) Index
b) Index_Progress
c) Muscle_Group
d) Training_Schedule

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank [ ] in Figure 2.

Figure 2 shows an SQL statement that lists all the fitness index progress of a specific customer during the specified period from the beginning date to the end date.
Here, `:FromDate` and `:ToDate` are host variables that contain the beginning date and the end date, respectively, and `:CustID` is a host variable that contains the ID of the specific customer.

```
SELECT Customer.Customer_ID, Customer_Name, Index_Name, Index_Value
FROM   Customer, Index, Index_Progress
WHERE  Customer.Customer_ID = :CustID
       AND Customer.Customer_ID = Index_Progress.Customer_ID
       [          D          ]
[          E          ]
```

Figure 2. SQL statement

Figure 3 shows an example of the output list. The list is arranged in ascending order of the date created.

Customer ID	Customer Name	Index Name	Index Value
C00001	Peter	Pectorals	7
C00001	Peter	Deltoids	5
C00001	Peter	Obliques	12
C00001	Peter	Deltoids	8

Figure 3. Example of the output list

Answer group for D

a) `AND Customer.Date_Created = Index_Progress.Date_Created`
   `AND Index_Progress.Date_Created BETWEEN :FromDate AND :ToDate`

b) `AND Customer.Date_Created = Index_Progress.Date_Created`
   `AND Index_Progress.Date_Created IN (:FromDate, :ToDate)`

c) `AND Index.Index_ID = Index_Progress.Index_ID`
   `AND Index_Progress.Date_Created BETWEEN :FromDate AND :ToDate`

d) `AND Index.Index_ID = Index_Progress.Index_ID`
   `AND Index_Progress.Date_Created IN (:FromDate, :ToDate)`

Answer group for E

a) `GROUP BY Index_Progress.Date_Created`
b) `GROUP BY Index_Progress.Index_ID, Index_Progress.Date_Created`
c) `ORDER BY Index_Progress.Date_Created`
d) ORDER BY Index_Progress.Index_ID, Index_Progress.Date_Created DESC

Q4. Read the following description concerning Analog/Digital conversion, and then answer Subquestion.

Pulse Code Modulation (PCM) is a method of converting an analog signal into digital data. The PCM encoder has three processes: (1) Sampling → (2) Quantizing → (3) Encoding.
An Analog/Digital conversion of n bits is described below, using the voltage of a direct current as an example:

(1) Sampling
In sampling, a voltage that is a continuous-time analog signal is measured at a fixed time interval. In Figure 1, the time axis is divided into t0, t1, … at equally-spaced time intervals of d, and the voltage at a given time is represented as v(t0), v(t1), … .

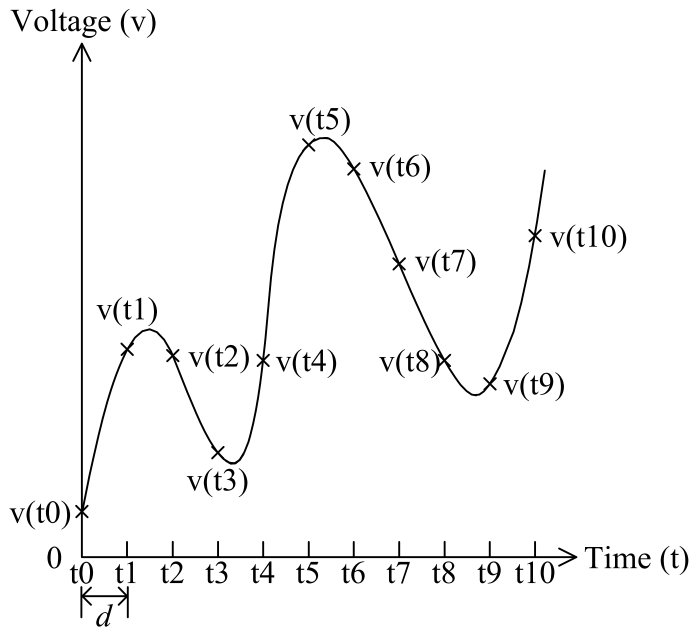


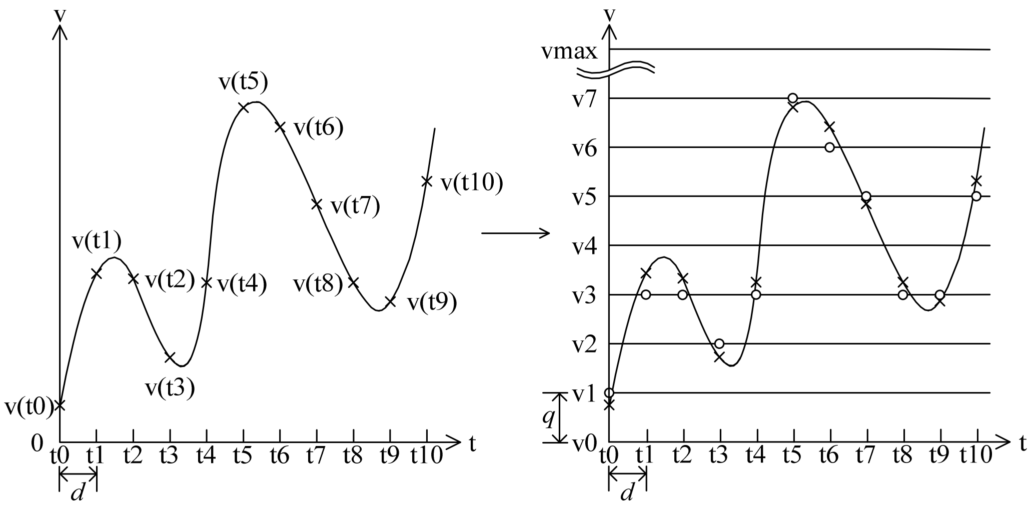
Figure 1. Example of sampling

(2) Quantization
In quantization, the voltages v(t0), v(t1), … obtained by the sampling in (1) are approximated with integral multiple values of step size q.
In order to perform quantization, first, the FSR, which is the maximum width of the voltage to be measured, is decided. Next, in order to perform quantization with n bits, step size q is determined by equally dividing the FSR into $(2^n - 1)$ parts. In this case, step size q is FSR $\div (2^n - 1)$.

In quantization, when v0 = 0, v1 = q, …, vmax = $(2^n - 1) \times q$ represent the voltage values used as approximate values in ascending order, with regard to the voltage v(tm) of the analog signal measured in sampling at the time tm, a non-negative integer N that satisfies the following condition is found in quantization, and the voltage $N \times q$ is taken as the measured voltage v(tm). This is called n bit quantization:

$$N \times q - q \div 2 \le \text{v(tm)} < N \times q + q \div 2$$

As shown on the right in Figure 2, after the voltage axis is divided into voltages v0, v1, …, vmax with step size q, for each of v(t0), v(t1), …, the closest voltage from v0 to vmax is taken as the measured value. For example, the measured value of v(t3) is v2.



Note: "○" is an approximated value of v(tm) in the integral multiples of q

Figure 2. Example of quantization

(3) Encoding

In encoding, the voltages v0, v1, …, vmax used in quantization in (2) are assigned binary codes. Each measured value is represented using these binary codes.

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description.

The encoding of the voltages v(t0), v(t1), …, v(t10) as shown on the left in Figure 2 is considered. Table 1 shows a case where the voltages v0, v1, …, v7 shown on the right in Figure 2 are assigned binary codes 000, 001, …, 111 in sequence.

Table 1. Voltages and corresponding codes

Voltage(v)	Code
v0	000
v1	001
v2	010
…	…
v7	111

Table 2 shows the results when the values measured for v(t0), v(t1), …, v(t10) shown on the left in Figure 2 are encoded in accordance with Table 1.

Table 2. Encoding of values measured at respective times

Time	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
Code	001	011	011	010			A				B

Note: The shaded parts are not shown.

When the voltage range of an analog signal is between 0 and 9V, if the FSR is set to 9V and quantization with 4 bits is performed, q is [C] V. The measured value of the voltage 7.49…V of the analog signal is [D] V, and when the measured value is encoded with the binary codes 0000, 0001, …, and 1111 in sequence as in Table 1, the binary code is [E].

Answer group for A and B

a) 011 b) 100 c) 101
d) 110 e) 111

Answer group for C and D

a) 0.5625 b) 0.6 c) 1.2 d) 2.25
e) 5.5 f) 7.0 g) 7.2 h) 7.5
i) 7.8 j) 8.0

Answer group for E

a) 1010 b) 1011 c) 1100
d) 1101 e) 1110

Q5. Read the following description concerning a program design, and then answer Subquestions 1 and 2.

Company *X* is a software development company. It accepts a project for developing an Online Job Matching System (OJMS). OJMS is a social Web site that offers services to three types of person: users, applicants and employers.
Users can search jobs, but can not apply for jobs. Applicants can search and apply for jobs, and update their profiles. Employers can post jobs.
If a user wants to apply for a job, he/she must be registered in this system. After registration, he/she becomes an applicant. When an applicant wants to search and apply for jobs, or update the profile, he/she must login to this system.
Figure 1 shows the use case diagram of OJMS, focused on the user/applicant portion.

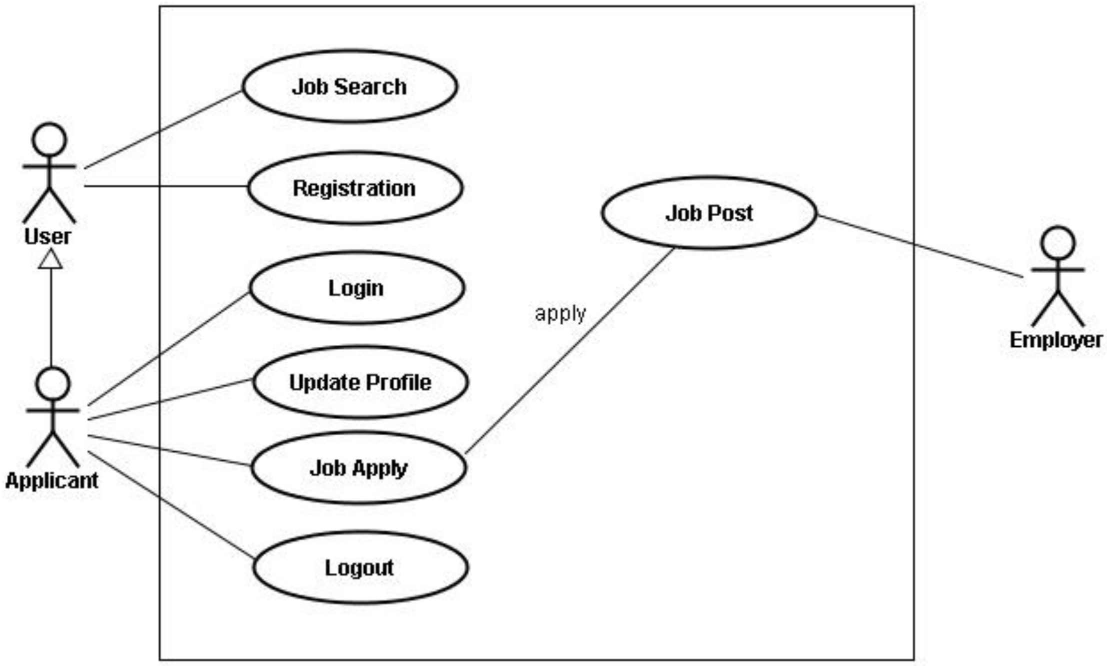


Figure 1. Use case diagram of OJMS

The following list shows the description of each use case shown in Figure 1.

Use case	Description
Job Search:	A user or an applicant searches jobs and views the details of the jobs. An applicant, if he/she wishes, can go directly to the Job Apply page.
Registration:	A user registers himself/herself on this system by entering information, such as user name, password, email, and phone number. After the user's successful completion of the registration, the system will send an email to the user, and he/she becomes an applicant.

Login:	An applicant logs in to the system. When he/she has logged in, the menu page is displayed.
Update Profile:	An applicant changes his/her registered information.
Job Apply:	An applicant applies for a selected job. When the applicant has completed the job application operation, the system sends an email to the applicant and go to the Logout page to exit. When the applicant cancels the job application operation, go to the Login page to display menu.
Job Post:	An employer posts jobs.
Logout:	An applicant logs out from the system.

Figure2 shows the high-level OJMS navigation map for users and applicants.

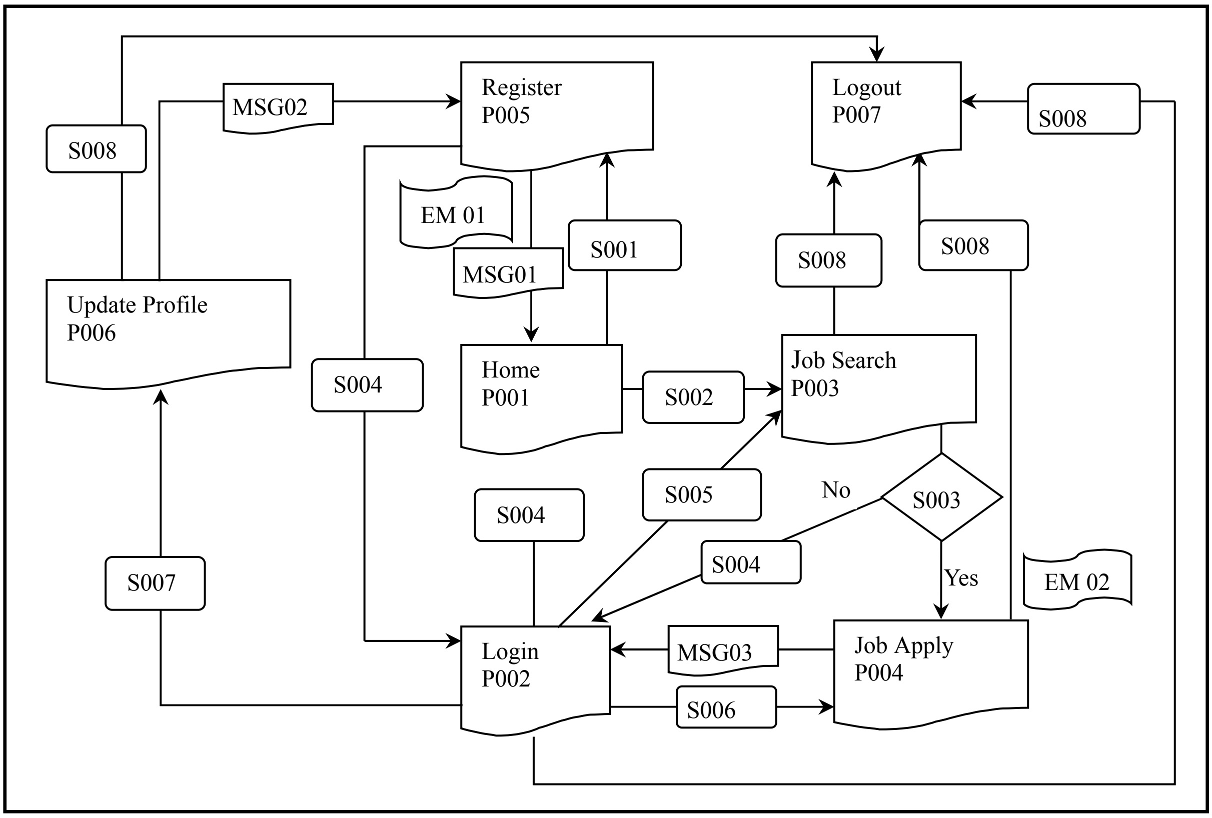


Figure 2. High-level OJMS navigation map

Table 1 shows the logical pages, Table 2 shows the screen transactions, Table 3 shows the display messages, and Table 4 shows the email messages.

Table 1. Logical pages

P001	Home Page
P002	Login Page
P003	Job Search Page
P004	Job Apply Page
P005	Register Page
P006	Update Profile Page
P007	Logout Page

Table 2. Screen transactions

S001	Create New Account
S002	Job Search (actor: A)
S003	Is actor B
S004	Login
S005	Job Search (actor: C)
S006	Job Apply (actor: C)
S007	Update Profile
S008	Logout

Table 3. Display messages

MSG01	"Registration completed"
MSG02	"Profile successfully updated"
MSG03	" D "

Table 4. Email messages

EM01	"Registration completed"
EM02	" E "

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [ ] in Tables 2, 3 and 4.

Here, the same answer can be selected twice or more if needed.

Answer group for A through C

a) applicant
b) either applicant or user
c) employer
d) user

Answer group for D and E

a) Job application canceled
b) Job application completed
c) Job application is for applicants only
d) Login is not yet complete

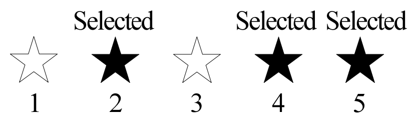
Subquestion 2

From the answer group below, select the incorrect sequence of page transition.

a) Home → Login → Job Apply → Logout
b) Home → Login → Update Profile
c) Home → Register → Home → Login
d) Home → Update Profile → Register → Login

Q6. Read the following description of a program and the program itself, and then answer Subquestion.

A program obtains all combinations when selecting *K* elements out of *N* elements. For example, when selecting 3 elements out of 5 elements, the program obtains all 10 combinations.
In the program, an array `S` containing `N` elements (element number: 1 to `N`) is prepared. In order to express a specific combination, the program sets 1s for the selected `K` elements in array `S` and 0s for the remaining elements. For example, when the three elements 2, 4, and 5 are selected from the five elements 1 to 5 as shown in Figure 1 (1), the program expresses the combination as shown in Figure 1 (2).

(1) 

(2) Array *S*

(Element number)	1	2	3	4	5
	0	1	0	1	1

Figure 1. Example of selecting 3 elements out of 5 elements, and its expression in the program

[Program Description]
The program comprises the main program `Main`, as well as the functions `Init` and `Next` for obtaining the combinations.

○`Main program: Main`

Description: The program obtains all 10 combinations for the case `N` = 5 and `K` = 3 (selecting 3 out of 5 elements). The result is set in array `S` sequentially.

○`Integer type function: Init(Integer type: S[], Integer type: N, Integer type: K)`

Arguments: `S[]` is an output argument. `N` and `K` are input arguments.

Description: When 1 ≤ `K` ≤ `N`, the function sets the first `K` elements of array `S` to 1, and sets the remaining `N` − `K` elements to 0. The function then returns 0 as the return value. Otherwise, the function returns −1 as the return value without setting any values in array `S`.

○`Integer type function: Next(Integer type: S[], Integer type: N)`

Arguments: `S[]` is an input / output argument. `N` is an input argument.

Description: The combination obtained most recently is already set in the first `N` elements of the received array `S`. By performing a predetermined operation for the received combination, the function obtains a new combination and sets it in array `S`, and then returns 0 as the return value. However, if the received combination is the final form that can be obtained by the algorithm used in this function, the function returns −1 as the return value without setting any values in array `S`.

[Program]

```
○Main program: Main
○Integer type: S[5], K, N, R          /*  1 ≤ K ≤ N  */
• K ← 3                    /*  Number of elements to be selected  */
• N ← 5                    /*  Number of total elements  */
• R ← Init(S, N, K)
■ R = 0                                          α
│ • R ← Next(S, N)
■

○Integer type function: Init (Integer type: S[],
                              Integer type: N, Integer type: K)
○Integer type: L
▲ 1 ≤ K and K ≤ N
│ ■ L: 1, L ≤ N, 1
│ │ ▲ L ≤ K
│ │ │ • S[L] ← 1
│ │ ├──────────
│ │ │ • S[L] ← 0
│ │ ▼
│ ■
│ • return 0
├──────────
│ • return −1
▼

○Integer type function: Next(Integer type: S[], Integer type: N)
○Integer type: C, L, R
• C ← 0
• L ← 1
• R ← −1
■ L < N and R = −1
│ ▲ S[L] = 1
│ │ ▲ S[L+1] = 0
│ │ │ • S[L] ← 0
│ │ │ • S[L+1] ← 1
│ │ │ • Init(S, L−1, C)
│ │ │ • R ← 0
│ │ ├──────────
│ │ │ • C ← C + 1
│ │ ▼
│ ▼
│ • L ← L + 1
■
• return R
```

Subquestion

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{\qquad}$ in the following description.

(1) Modify the main program `Main` so that it can print the contents of array `S` whenever one state of a combination is obtained in array `S`. The following subprogram is used for printing:

○`Subprogram Dump(Integer type: S[], Integer type: N)`

Arguments: `S[]` and `N` are input arguments.

Description: The function prints the values stored in the first `N` elements of array `S` on a single line.

For printing, the part indicated by ***α*** in the main program `Main` is replaced with the following part:

```
• R ← Init(S, N, K)
■ R = 0
│  [     A     ]
■
```

(2) The function `Next` searches the received array `S` from the smallest element number, and exchanges the contents of two consecutive elements that have the values $\boxed{\text{B}}$. Next, the function `Next` invokes the function `Init` for the part of array `S`, namely, $\boxed{\text{C}}$ the two consecutive elements. For example, if the contents of the element numbers 1 to 5 of array `S` are 1, 0, 1, 0, 1 at the beginning of the execution of the function `Next`, the contents of the element numbers 1 to 5 of array `S` at the termination of the execution are $\boxed{\text{D}}$.

(3) When this program is executed and the function `Init` is invoked from the function `Next`, the value of `N` received by the function `Init` ranges $\boxed{\text{E}}$, and the value of `K` ranges from 0 to 2. Thus, in some cases, the value of `N` and `K` received by the function `Init` may not satisfy 1 ≤ `K` ≤ `N`.

(4) When the execution of the main program `Main` is complete, the contents of the element numbers 1 to 5 of array `S` will be $\boxed{\text{F}}$.

Answer group for A

a) • `Dump(S, N)`

b) • `Dump(S, N)`
   • `R ←Next(S, N)`

c) • `R ←Next(S, N)`
   • `Dump(S, N)`

Answer group for B

a) 0, 1 that are found first
b) 0, 1 that are found last
c) 1, 0 that are found first
d) 1, 0 that are found last

Answer group for C

a) after
b) after and including
c) before
d) before and including

Answer group for D

a) 0, 1, 1, 0, 1
b) 1, 0, 0, 1, 1
c) 1, 0, 1, 1, 0
d) 1, 1, 0, 0, 1

Answer group for E

a) from 0 to 2
b) from 0 to 3
c) from 1 to 3
d) from 1 to 4

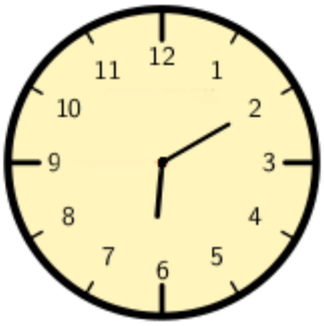
Answer group for F

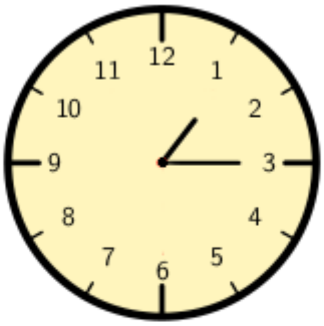
a) 0, 0, 0, 0, 0
b) 0, 0, 1, 1, 1
c) 1, 1, 1, 0, 0
d) 1, 1, 1, 1, 1

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestions 1 and 2.

Analog clocks indicate time using angles. Usually, an analog clock has three hands: the hour hand, minute hand, and second hand. Sometimes it has only two hands: the hour hand and minute hand. Every second, all three (or two) hands of the analog clock change their positions, creating different angles between them.





For instance, at 06:10, the angle between the hour hand and the minute hand is 125°, and at 01:15, the angle is 52.5°. Here, only acute, right, obtuse or straight angle between the hands are considered (namely, 0° ≤ the angle ≤ 180°).
Commonly, there are two types of time format: 12 hour and 24 hour. However, an analog clock can show time only in the 12 hour format. For example, at 06:10 and 18:10, the hour hand and minute hand of an analog clock are in the same position.

[Program Description]

(1) In this program, the analog clock is considered to have only two hands (hour and minute).

(2) The program reads a series of time data from the standard input. The input for the program is given in the following order. The first line contains the number of time data `n`, where 1 ≤ `n` ≤ 51. The next `n` lines contain two integer values `hh` (0 ≤ `hh` ≤ 23) and `mm` (0 ≤ `mm` ≤ 59), indicating the hour and minute of a particular time in the 24 hour time format. Two integer values are separated by one or more space characters.

(3) The program prints `n` lines as output. Each line contains three values: hour, minute, and the angle between the hour and minute hands. The output is sorted in ascending order of the angle. If there are time data with the same angle, then the earlier time should appear first (00:00 is the earliest time and 23:59 is the latest time)

(4) The following list shows an example of input data.

```
4
6 20
0 0
15 0
6 0
```

(5) The following list shows an example of output for the input data shown in (4) above.

```
 0  0   0.0
 6 20  70.0
15  0  90.0
 6  0 180.0
```

(6) In the program, a structure named `time` is used. This structure has three integer variables `hh`, `mm`, and `angle`, and one long integer variable `sortValue`. The variables `hh` and `mm` are used to store the hour value and minute value of the time data. The variable `angle` holds the 10 times value of the angle created by the hour and minute hands of the clock. The variable `sortValue` holds a value that is used to sort the list of time data.

(7) Two user defined functions are used.

(i) `void setSortingValue(int i)`

This function combines three values (`hh`, `mm` and `angle`) into a single value, and sets this value to the variable `sortValue` of a particular element having index `i`.

By using `sortValue` as the sort key, `t[]` can be sorted in ascending order of the angle and time.

(ii) `void sort(int n)`

This function sorts `t[]` in ascending order of `sortValue`, where `n` indicates the number of elements in `t[]`.

[Program]

```
#include <stdio.h>

void setSortingValue(int);
void sort(int);

struct time {
   int hh, mm, angle;
   long sortValue;
} t[51];

void setSortingValue(int i) {
   t[i].sortValue = 10000 * (long)t[i].angle
                  + 100 * t[i].hh + t[i].mm;
}

void sort(int n) {
   int i,j;
   struct time temp;

   for (i = 0; i < n - 1; i++) {
      for (j = i + 1; j < n; j++) {
         if (     A     ) {
            temp = t[i];
            t[i] = t[j];
            t[j] = temp;
         }
      }
   }
}
```

```
int main() {
   int n, i, hAngle, mAngle, angle;

   scanf("%d", &n);

   for (i = 0; i < n; i++) {
      scanf("%d%d", &t[i].hh, &t[i].mm);
      hAngle = (t[i].hh % 12 * 60 + t[i].mm) * 5;
      mAngle = [    B    ];
      if (hAngle > mAngle) {
         t[i].angle = hAngle - mAngle;
      } else {
         t[i].angle = mAngle - hAngle;
      }
      if (t[i].angle > 1800) {
         t[i].angle = [    C    ];
      }
      setSortingValue(i);
   }

   sort(n);

   for (i = 0; i < n; i++) {
      printf("%2d %2d %5.1f\n", t[i].hh, t[i].mm, [    D    ]);
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank [   ] in the above program.

Answer group for A

a) `t[i].sortValue > t[j-1].sortValue`
b) `t[i].sortValue < t[j].sortValue`
c) `t[i].sortValue > t[j].sortValue`
d) `t[i+1].sortValue < t[j].sortValue`

Answer group for B

a) `t[i].mm * 5`	b) `t[i].mm * 10`	c) `t[i].mm * 30`
d) `t[i].mm * 60`	e) `t[i].mm * 180`	f) `t[i].mm * 360`

Answer group for C

a) `900 + t[i].angle` b) `3600 - t[i].angle`

c) `t[i].angle - 900` d) `t[i].angle – 1800`

Answer group for D

a) `t[i].angle % 10` b) `t[i].angle % 3600`

c) `t[i].angle * 10` d) `t[i].angle * 10.0`

e) `t[i].angle / 10` f) `t[i].angle / 10.0`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank [ ] in the following description.

When the values in `t[4]` are, `t[4].hh` = 23, `t[4].mm` = 59 and `t[4].angle` = 55, the value stored in `t[4].sortValue` will be [E] after `SetSortingValue(4)` is executed.

Answer group

a) 551159 b) 552359

c) 5501159 d) 5502359

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestions 1 and 2.

[Program Description]

(1) An education division in an IT company tasks its trainee programmers to develop a program that determines the day of the week of the birth date of given employees.

(2) Figure 1 shows the execution result of this program.

```
ID: 1
Department: IT
Name: Dela Cruz, John X.
BirthDate: 7/8/1974 (Monday)

ID: 2
Department: AC
Name: Victoria, Glenda Y.
BirthDate: 4/29/1980 (Tuesday)

ID: 3
Department: HR
Name: Hilary, Brent Z.
BirthDate: 10/25/1991 (Friday)
```

Figure 1. Execution result of the program

(3) The programs are composed of four classes: `Date`, `Person`, `Employee` and `BirthDayRooster`.

(4) The class `Date` contains a method `dayOfWeek()` that uses Zeller's algorithm for determining the day of the week. The method returns a value between 0 and 6 that represents the day of the week as follows:

0 – Saturday
1 – Sunday
2 – Monday
3 – Tuesday
4 – Wednesday
5 – Thursday
6 – Friday

[Program 1]

```
public class Date {
    private int month;
    private int day;
    private int year;

    public Date(int month, int day, int year) {
        this.month=month;
        this.day=day;
        this.year=year;
    }
    public int dayOfWeek() {
        int t, mm, dd, yy, cc;
        mm = month;
        yy = year;
        if (mm < 3) {
            mm = mm + 12;
            yy = yy - 1;
        }
        cc = (int) (yy / 100);
        yy = yy % 100;
        t = day + (int) (26 * (mm + 1) / 10) + yy;
        t = t + (int) (yy / 4) + (int) (cc / 4) - 2 * cc;
        dd = t % 7;
        if (dd < 0)
            dd = dd + 7;
        return dd;
    }
    public String toString() {
        return month+"/"+day+"/"+year;
    }
}
```

[Program 2]

```
public class Person {
    private String lastName;
    private String firstName;
    private char middleInitial;
    private Date birthdate;
```

```
    public Person(String lastName, String firstName,
                  char middleInitial, Date birthdate) {
        this.lastName = lastName;
        this.firstName = firstName;
        this.middleInitial = middleInitial;
        this.birthdate = birthdate;
    }
    public void setBirthdate(Date birthdate) {
        this.birthdate=birthdate;
    }
    public Date getBirthdate() {
        return birthdate;
    }
    public String toString() {
        return "Name: " + lastName + ", " + firstName
               + " " + middleInitial + ". \nBirthDate: " + birthdate;
    }
}
```

[Program 3]

```
public class Employee [      A      ] {
    private static int counter = 1;
    private int id;
    private String department;

    public Employee(String department, String lastName,
                    String firstName, char middleInitial,
                    Date birthdate) {
        [      B      ];
        id = counter++;
        this.department = department;
    }
    public int getId() {
        return id;
    }
    public String getDepartment() {
        return department;
    }
    public String toString() {
        return "ID: " + getId() + "\nDepartment: "
               + getDepartment() + "\n" + [      C      ];
    }
}
```

[Program 4]

```
public class BirthDayRooster {
    [    D    ] = {"Saturday", "Sunday", "Monday",
                      "Tuesday", "Wednesday", "Thursday", "Friday"};

    public static void main(String[] args) {
        Employee[] emp = {new Employee("IT", "Dela Cruz", "John",
                                       'X', new Date(7, 8, 1974)),
                          new Employee("AC", "Victoria", "Glenda",
                                       'Y', new Date(4, 29, 1980)),
                          new Employee("HR", "Hilary", "Brent",
                                       'Z', new Date(10, 25, 1991))};

        for (Employee e : emp) {
            System.out.println(e + " (" + [    E    ] + ")");
            System.out.println();
        }
    }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank $\boxed{}$ in the above program.

Answer group for A

a) `extends BirthDayRooster`
b) `extends Date`
c) `extends Person`
d) `implements BirthDayRooster`
e) `implements Date`
f) `implements Person`

Answer group for B and C

a) `super()`
b) `super(lastName, firstName, middleInitial, birthdate)`
c) `super.toString()`
d) `this()`
e) `this(lastName, firstName, middleInitial, birthdate)`
f) `this.toString()`

Answer group for D

a) `private String[] day`
b) `static String[] day`
c) `String[] day`
d) `String[] static day`

Answer group for E

a) `day.e.getBirthdate().dayOfWeek()`
b) `day.emp.getBirthdate().dayOfWeek()`
c) `day[e.getBirthdate().dayOfWeek()]`
d) `day[emp.getBirthdate().dayOfWeek()]`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank $\boxed{}$ in the following description.

According to the method `dayOfWeek()` in the class `Date` shown in [Program 1], the day of the week of 01/20/1901 was $\boxed{\quad F \quad}$.

Answer group

a) Monday
b) Tuesday
c) Wednesday
d) Thursday
e) Friday
f) Saturday
g) Sunday

20th ITPEC FE Afternoon Examination, October 2015

Question No.	Subquestion No.		Correct Answers
1	1		e
	2	A	h
		B	e
		C	d
	3	D	b
2		A	c
		B	b
		C	c
		D	b
3	1	A	e
		B	d
		C	d
	2	D	c
		E	c
4		A	d
		B	c
		C	b
		D	g
		E	c
5	1	A	d
		B	a
		C	a
		D	a
		E	b
	2		d
6		A	b
		B	c
		C	c
		D	a
		E	b
		F	b
7	1	A	c
		B	d
		C	b
		D	f
	2	E	b
8	1	A	c
		B	b
		C	c
		D	b
		E	c
	2	F	g

May 2015

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.
   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is this spring Fundamental IT Engineer Examination conducted?

   Answer group
   a) April b) May c) June d) July

   Since the correct answer is " b) May ", mark your answer sheet as follows:

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated.

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc. of procedures, variables, etc.
/* text */		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 ――― process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then the process 1 is executed. If it is `false`, then the process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	`+, −, not`	High
Multiplication, division	`×, ÷, %`	↑
Addition, subtraction	`+, −`	
Relational operation	`>, <, ≥, ≤, =, ≠`	
Logical product	`and`	↓
Logical sum	`or`	Low

Note: With division of integers, integer quotient is returned as a result.
The % operator indicates a remainder operation.

Notations used for E-R diagrams

In questions that use E-R diagrams, the following notations are used unless otherwise stated.

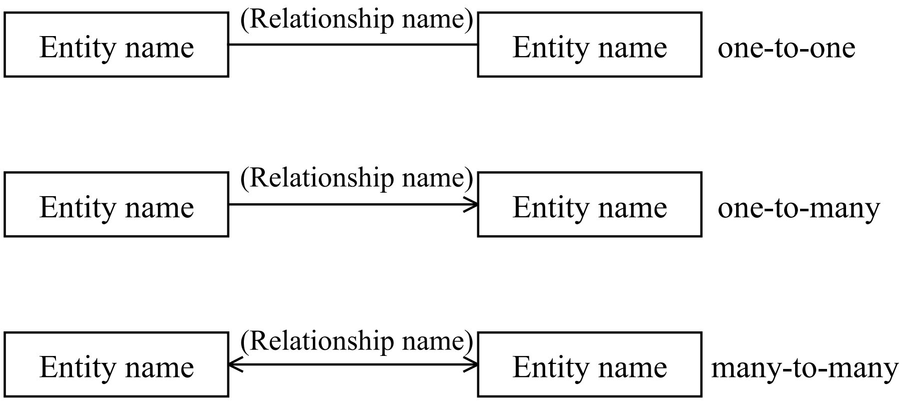


Figure Notations used for entities and relationships

1. Entities are represented by rectangles.
2. Entity names are indicated within the rectangles.
3. A relationship between entities is represented by a line.
   The relationship name is indicated at the side of the line as “(Relationship name)”.
   The relationship name can be omitted.
4. A “one-to-one” relationship is represented by a line.
   A “one-to-many” relationship is represented by a line with an arrow pointing towards the “many” side.
   A “many-to-many” relationship is represented by a line with an arrow on both ends.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description concerning memory management, and then answer Subquestions 1 and 2.

Many applications for mobile devices make use of data that is partially stored in a data buffer on a storage media of the devices. Most of the data is stored in an external server accessed through the Internet. When a user tries to access data not found in the device, the application connects to the server to retrieve the data. The new data will replace some of the data existing in the device due to memory constraints. An efficient implementation reduces the number of times data is retrieved from the server. This efficiency is influenced by factors such as the amount of a data buffer needed to store the partial data, the method of determining the data to be replaced by the new data, and the pattern of how the user accesses the data.

In the case of an application that accesses 6 different data blocks repeatedly by using a data buffer that can hold 5 data blocks, fetching data from an external server is necessary all the time.

Figure 1 illustrates how the data buffer is being replaced in this case. In Figure 1, the set of data blocks at the top represents the data being accessed by the application, and the set of data blocks at the bottom represents the data buffer and its corresponding data. When the application accesses the data block [6], it is fetched from the server and replaces [1]. When the application accesses [1] after [6], it is fetched from the server and replaces [2], and so on.

This uses a First-In-First-Out (FIFO) method. But if the data buffer can hold 6 data blocks, then there will be only 6 fetches at the start of the application, and there is no need to fetch data till the end of the application.

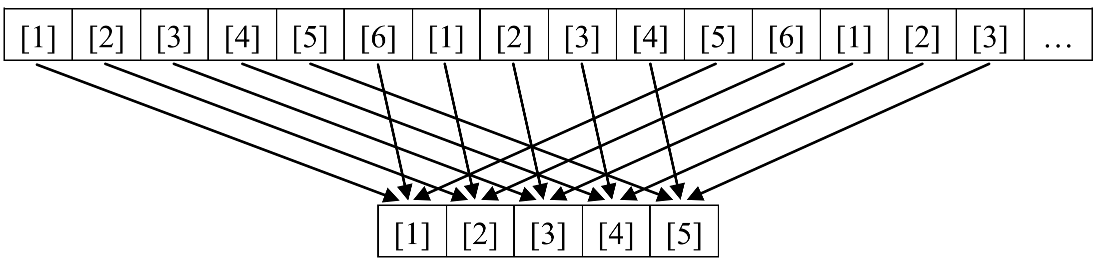


Figure 1. Accessing 6 data blocks repeatedly by using data buffer

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the following description.

Assuming that an application for a mobile device accesses 6 data blocks [1], [2], … , [6] in the following order.

									▼							
[1]	[2]	[3]	[5]	[6]	[1]	[2]	[3]	[4]	[6]	[1]	[2]	[5]	[6]	[4]	[1]	[2]

The device has a data buffer that can hold 5 data blocks. When the application starts, the data blocks [1], [2], … , [5] have already been loaded into the data buffer in this sequence.

[1]	[2]	[3]	[4]	[5]

When the FIFO method is used, at the point ▼ shown above immediately after the data block [6] is accessed, the contents of the data buffer will be $\boxed{\text{A}}$, and at the end of the series of data block accesses, the contents of the data buffer will be $\boxed{\text{B}}$ with a total of $\boxed{\text{C}}$ data block replacements for the entire duration of the application.

Answer group for A and B

a)	[1]	[2]	[3]	[6]	[4]	b)	[1]	[2]	[5]	[6]	[4]
c)	[4]	[1]	[2]	[5]	[6]	d)	[4]	[2]	[3]	[5]	[6]
e)	[5]	[6]	[1]	[2]	[4]	f)	[6]	[1]	[2]	[3]	[4]

Answer group for C

a) 3 b) 4 c) 5 d) 8 e) 9 f) 10

Subquestion 2

From the answer group below, select the correct answer to be inserted into each blank $\boxed{}$ in Table 1.

Two alternative methods are explored to replace the FIFO method.

The first alternative is called the Most Frequently Used (MFU) method. In the MFU method, the system counts how many times each data block was used while it remains in the data buffer. The data block with the lowest count is the data block that is replaced when a new data block needs space. If multiple data blocks have the same count, then the oldest data block is replaced according to the FIFO method.

The second alternative is called the Least Recently Used (LRU) method. In the LRU method, the system keeps track of which data block was used when while it remains in the data buffer. The least recently used data block is the data block that is replaced when a new data block needs space.

Table 1 shows the analysis of the two methods for different program behaviors.
Here, the data buffer can hold only 5 data blocks.

Table 1. Analysis of the two methods

Program behavior	MFU method	LRU method
A program accesses different data blocks randomly throughout its use.	Since each new access will require to bring in new data block, there will be no improvement.	Since each new access will require to bring in new data block, there will be no improvement.
A program accesses 8 data blocks with the following cyclic pattern: First 3 data blocks → 2 data blocks out of the remaining 5 → first 3 data blocks → 2 data blocks out of the remaining 5 → … .	D	D
A program first accesses 5 data blocks repeatedly (10 times), then accesses the other 3 data blocks repeatedly (10 times).	E	F

Answer group

a) For the first 5 data blocks, once they are loaded into the data buffer, no replacement will take place during their repeated use. For the next 3 data blocks, replacement will always take place for each succeeding data block access.

b) For the first 5 data blocks, once they are loaded into the data buffer, no replacement will take place during their repeated use. Similarly, for the next 3 data blocks, once they are loaded, no replacement will take place during their repeated use.

c) No data blocks out of 8 will stay in the data buffer permanently during the program execution. All 8 data blocks will be subject to the replacement.

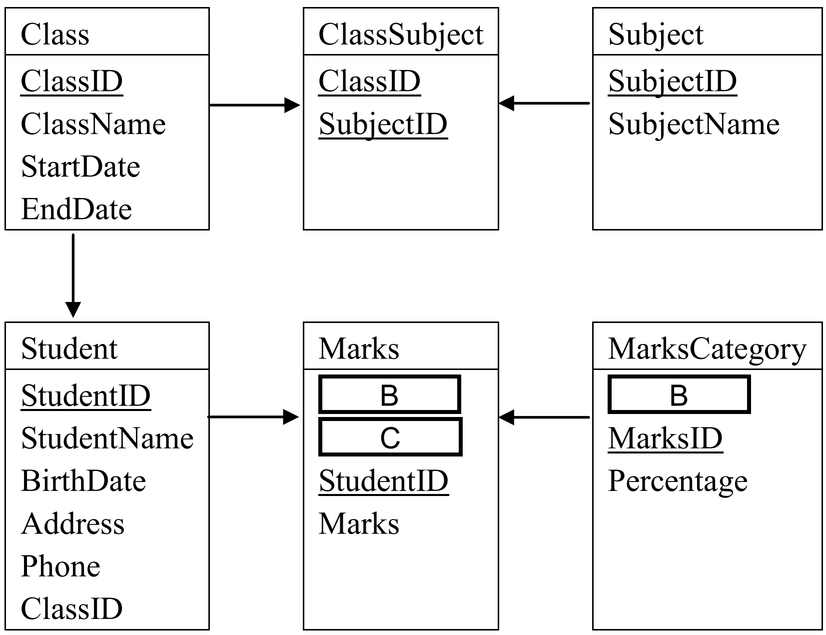
d) The 3 data blocks out of 8 will stay in the data buffer permanently during the program execution. Other 5 data blocks will be subject to the replacement.

Q2. Read the following description concerning a database for student management, and then answer Subquestions 1 and 2.

An IT faculty of a university is going to develop the student management system. Ms. *T* is in charge of getting requirements for this system. She summarizes main requirements:

1) Each student can be enrolled in only one class. Information about the student are ID, name, birth date, address, and phone number. Information about the class are ID, name, start date, and end date.
2) Some different subjects will be taught in one class.
3) Each subject has mark items such as exercises, tests, and final exam, to evaluate the students' competence for the subject. The responsible lecturer will decide the mark items and their contribution percentage. For example, the total marks of the subject CNET will be evaluated by the mark items Lab, Test, Project, Practical exam, and Final exam with percentages 15%, 15%, 20%, 20%, and 30% respectively.

Based on the above requirements, Ms. *T* is designing the database for the system. Figure 1 shows an E-R diagram of the database she is now creating. This E-R diagram is incomplete, and "one-to-many" relationship from $\boxed{\text{A}}$ to MarksCategory is missing.



(Note) A solid underline _____ indicates an attribute of a primary key.

Figure 1. E-R Diagram of the database (incomplete)

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the above description and Figure 1.

Answer group for A

a) Class
b) ClassSubject
c) Student
d) Subject

Answer group for B and C

a) <u>ClassID</u>
b) <u>MarksID</u>
c) <u>StudentID</u>
d) <u>SubjectID</u>

Subquestion 2

From the answer group below, select the correct answer to be inserted into each blank $\boxed{}$ in the following SQL statement.

At the end of a semester, the responsible lecturer outputs the list of all students with total marks for the specified class and subject as shown below, based on the data in the database.

```
Student ID  Student Name  Total Marks
   S014       B. Eeeee        92
   S015       B. Fffff        65
   S035       J. Ccccc        89
   ...        ...             ...
```

Total marks of each student is sum of individual marks of mark items multiplied with their contribution percentages. For example, when the total marks of the subject CNET will be evaluated by the mark items Lab, Test, Project, Practical exam, and Final exam with percentages 15%, 15%, 20%, 20%, and 30% respectively, the total marks will be:

$$\text{Total marks} = (\text{Lab marks} \times 15 + \text{Test marks} \times 15 + \text{Project marks} \times 20 + \text{Practical exam marks} \times 20 + \text{Final exam marks} \times 30) \div 100$$

Here, ClassID and SubjectID are given by the host variables `:ClassID` and `:SubjectID` respectively.

```
SELECT    Student.StudentID, StudentName,
          [    D    ] AS TotalMarks
FROM      Class, Student, Marks, MarksCategory
WHERE     Class.ClassID = :ClassID
  AND     Marks.SubjectID = :SubjectID
  AND     Class.ClassID = Student.ClassID
  AND     [    E    ]
GROUP BY  [    F    ]
```

Answer group

a) `AVG(Marks * Percentage)`

b) `Student.StudentID = Marks.StudentID`
   `AND Marks.SubjectID = MarksCategory.SubjectID`

c) `Student.StudentID = Marks.StudentID`
   `AND Marks.SubjectID = MarksCategory.SubjectID`
   `AND Marks.MarksID = MarksCategory.MarksID`

d) `Student.StudentID, StudentName`

e) `StudentID`

f) `StudentID, StudentName`

g) `SUM(Marks * Percentage) / 100`

Q3. Read the following description concerning network trouble-shooting, and then answer Subquestions 1 and 2.

Company *U* has setup its IP network which mainly provides Intranet application services, network-based office automation, and the Internet access services to its employees. The network can be accessed via wired Ethernet connections within the company covering two physical rooms. In the network, Internet access router and 3 Layer-2 switches (SW1, SW2 and SW3) are installed. Figure 1 shows the configuration of the network in Company *U*.

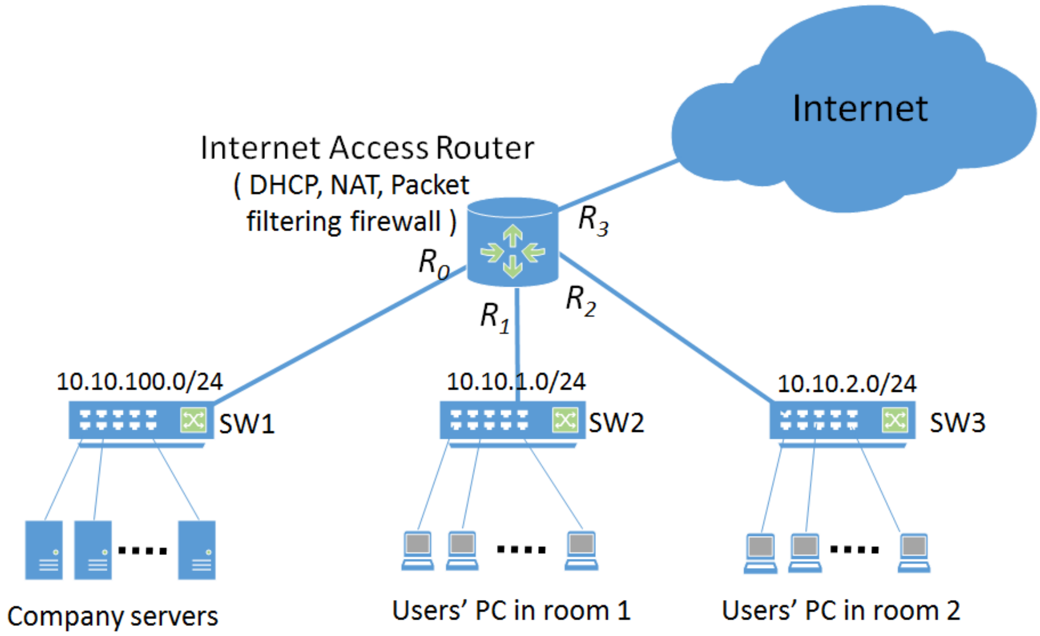


Figure 1. Configuration of the network in Company *U*

Table 1. OSI reference model with examples of possible problems

Layer	Examples of possible problems
7 Application	• DSN misconfiguration • Server problem • Application program problem
6 Presentation	
5 Session	
4 Transport	• TCP or UDP port blocking
3 Network	• IP address misconfiguration • IP routing table incomplete • Default router problem
2 Data link	• MAC address blocking • MAC level authentication
1 Physical	• Cable unplug • Cable defect • Network port malfunction

In Company *U*, Mr. *M* has a responsibility to take care of any technical problems relevant to the network services. Based on his 3-year experiences, he has developed a procedure for network trouble-shooting which can cover 90% of daily network problems. The procedure uses bottom-up approach to identify the causes of the problems according with OSI reference model as shown in Table 1.

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank [ ] in the following description.

One day, an organization change was performed. A large number of employees changed their workplaces from room 1 to room 2, or from room 2 to room 1. After the change, some employees made complaints about the network connection. When being informed about the problems, Mr. *M* took the first step to determine whether the problems can be categorized as Layer 1 or 2 problems or not, by checking the information in the Layer-2 switches used by those employees.

Table 2 shows the list of users who could not use the network, and Table 3 shows the snapshot of the switch usage data taken by Mr. *M*.

Table 2. List of users who could not use the network

User name	Switch/Port ID	User PC's MAC address
User *A*	SW2 Port09	AA:AA:AA:AA:AA:AA
User *B*	SW2 Port13	BB:BB:BB:BB:BB:BB
User *C*	SW3 Port01	CC:CC:CC:CC:CC:CC
User *D*	SW3 Port12	DD:DD:DD:DD:DD:DD
User *E*	SW3 Port21	EE:EE:EE:EE:EE:EE

Table 3. Snapshot of the switch usage data

Switch/Port ID	Link status	Frame Send/Receive CRC error	Recognized MAC	MAC filter setting
SW2 Port09	Down	0%	Not Detected	Allow: All
SW2 Port13	Up	0%	BB:BB:BB:BB:BB:BB	Allow: All
SW3 Port01	Up	0%	CC:CC:CC:CC:CC:CC	Allow: DD:DD:DD:DD:DD:DD
SW3 Port12	Up	85%	DD:DD:DD:DD:DD:DD	Allow: DD:DD:DD:DD:DD:DD
SW3 Port21	Up	0%	EE:EE:EE:EE:EE:EE	Allow: All

Judging from Table 2 and Table 3, Mr. *M* found out that [A] had cabling or switch port problems. He solved these problems by taking the steps such as checking the cable plugging status, replacing the network cable, and changing the plugging switch port.
Also, Mr. *M* found out that [B] had MAC configuration problem. He solved this problem by changing the MAC filter setting.

Answer group

a) User *A*, User *B* and User *E*
b) User *A* and User *C*
c) User *A* and User *D*
d) User *C*
e) User *C* and User *D*
f) User *D*

Subquestion 2

From the answer groups below, select the correct answer to be inserted into each blank [ ] in the following description.

Mr. *M* was informed that User *F*'s PC could not send IP packets to both Intranet and the Internet. By checking the IP address configuration in Table 4 and the IP routing table in Table 5, Mr. *M* found out that this was a Layer 3 problem concerning the IP routing table. Accordingly, he added the missing routing record [C] to the IP routing table in Table 5, and then User *F*'s PC was able to access both Intranet and the Internet.
Here, the IP addresses of interfaces R_0, R_1, R_2 and R_3 at Internet access router in Figure 1 are configured as 10.10.100.254, 10.10.1.254, 10.10.2.254 and 202.170.10.1, respectively.

Table 4. IP address configuration of User *F*'s PC

User name	User PC's IP address	Netmask	Switch/Port ID
User *F*	10.10.2.55	255.255.255.0	SW3 Port19

Table 5. IP routing table of User *F*'s PC

Destination	Netmask	Gateway
10.10.2.0	255.255.255.0	(not specified)

After clearing Layer 1-3 problems, Mr. *M* set about solving the remaining problem related to Web site accesses from a Web browser. In order to identify the cause, he tried to access the Web site "http://www.itpec.org", whose IP address is 64.22.66.88, from the user PC's Web browser. As the test result showed that [D], he was confident that the cause of the problem was DNS misconfiguration. Then he changed the related settings, and the problem was solved.

Answer group for C

	Destination	Netmask	Gateway
a)	0.0.0.0	0.0.0.0	10.10.1.254
b)	0.0.0.0	0.0.0.0	10.10.2.254
c)	0.0.0.0	0.0.0.0	202.170.10.1
d)	10.0.0.0	255.0.0.0	10.10.1.254
e)	10.10.0.0	255.255.0.0	10.10.2.254
f)	10.10.2.0	255.255.255.0	202.170.10.1

Answer group for D

a) http://www.itpec.org was accessible and http://64.22.66.88 was accessible
b) http://www.itpec.org was accessible and http://64.22.66.88 was not accessible
c) http://www.itpec.org was not accessible and http://64.22.66.88 was accessible
d) http://www.itpec.org was not accessible and http://64.22.66.88 was not accessible

Q4. Read the following description concerning Web site security, and then answer Subquestions 1 through 3.

A mid-sized company has developed a Web application for online sales reporting system to enable its sales force to process sales data from remote sites. The application transfers certain secured sales data through the Internet, and that is a big concern for the company. Therefore, the company has established an SSL (Secure Sockets Layer) enabled Web site for this application.

Processing information securely through the Internet means that the information need to be transmitted between the sender and the receiver in a manner that makes it difficult for other people to intercept and read. SSL takes care of this, and it works through a combination of programs and encryption/decryption routines that exist on both the application on the Web site and the browser program on the user computer.

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank in Figure 1.

In HTTPS communication, the handshake is the most complicated phase in the process. The handshake synchronizes the client and the server up with the encryption methods and keys that will be used for the subsequent communication.

The HTTPS handshake consists of multiple steps as shown in Figure 1.

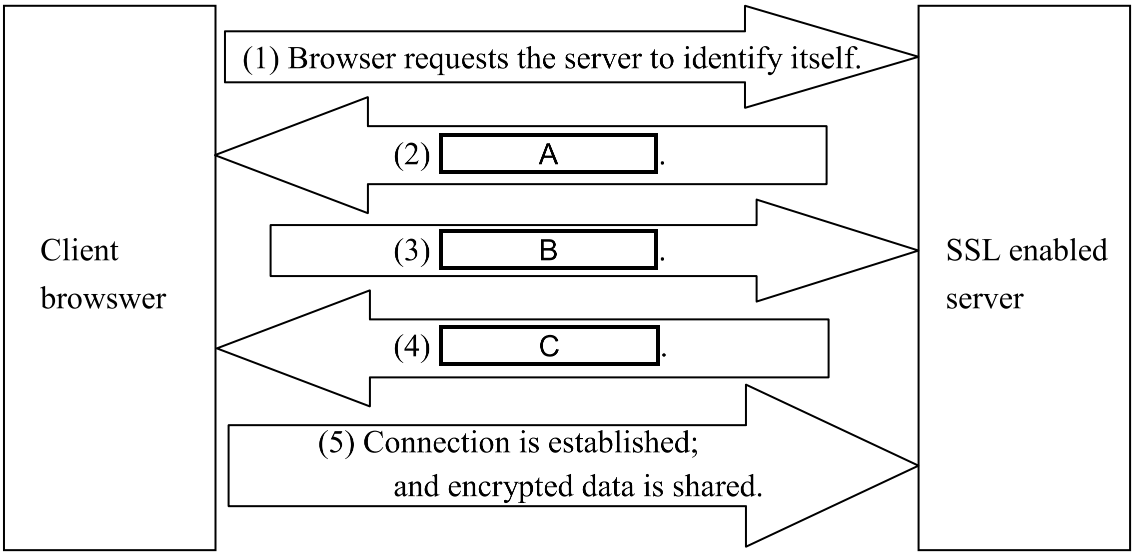


Figure 1. Flow of HTTPS handshake

Answer group

a) Browser sends a copy of the SSL certificate to the server
b) Browser verifies the certificate and sends an acknowledgment message to the server
c) Server sends a copy of the SSL certificate to the browser
d) Server sends a digitally signed acknowledgement to the browser

Subquestion 2

From the answer group below, select the correct answer to be inserted into each blank [ ] in Figure 2.

The HTTPS protocol has several commands such as GET, POST, PUT and DELETE. The application uses POST command to transfer secured sales data from remote sites. The software developer wanted to test if SSL is properly configured or not, so he used a network monitoring tool and made a POST request to the server https://www.example.com (IP address: 93.184.216.119) from his PC (IP address: 192.168.1.103).
In the monitoring tool, he found a "Server Hello" packet header as shown in Figure 2.
Here, the port number for https service is 443.

```
Frame 43: 1414 bytes on wire (11312 bits), 1414 bytes captured (11312 bits) on interface 0
Ethernet II, Src: Tp-LinkT_d0:d2:44 (74:ea:3a:d0:d2:44), Dst: Azurewav_e7:10:41 (48:5d:60:e7:10:41)
Internet Protocol Version 4, Src: [    D    ], Dst: [    E    ]
Transmission Control Protocol, Src Port: https (443), Dst Port: 3par-evts (5781), Seq: 1, Ack: 230, Len: 13
    Source port: https (443)
    Destination port: 3par-evts (5781)
    [Stream index: 6]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1361    (relative sequence number)]
    Acknowledgment number: 230    (relative ack number)
    Header length: 20 bytes
```

Figure 2. "Server Hello" packet header

Answer group

a) 93.184.216.119
b) 192.168.1.103

Subquestion 3

From the answer group below, select the correct answer to be inserted into each blank ______ in the following description.

Once SSL was configured properly and tested successfully as well, the software developer started a browser on his PC and tried to browse the Web pages. He found that the system was working smoothly with non-secured Web pages. When he tried to browse a secured page at https://www.example.com/order, he got the security message as shown in Figure 3.

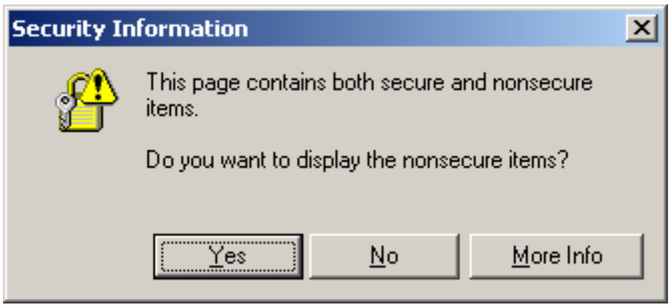


Figure 3. Security message

Source code

```
      <!DOCTYPE html>
      <html>
      <head>
      <title>Secured Page</title>
(1) → <link rel="stylesheet" type="text/css"
            href="http://www.example.com/themes/main.css">
(2) → <link rel="stylesheet" type="text/css"
            href="http://www.example.com/themes/header.css">
(3) → <script type="text/javascript"
              src="https://www.example.com/scripts/jquery.js"></script>
      </head>
      <body>
      <h1>Order Form</h1>
      <div>
(4) → <img src="http://www.example.com/images/shirt.jpg">
      <h4>Polo T-Shitr</h4>
      </div>
      <form action="/order/create" method="post">  Quantity:
      <input type="text" name="quantity" value="1">
      <input type="submit" value="Buy">
      </form>
      </body>
      </html>
```

Figure 4. HTML source code of the requested Web page

When the software developer opened the source file, he found the HTML (HyperText Markup Language) source code of the requested Web page, as shown in Figure 4. HTML is used most commonly for publishing Web pages.

In HTML, the contents are encoded using <tags> for annotating and formatting. Different tags are used for different purposes. A few of them are used to mention external resources such as images, link and Java scripts.

The tag `<link rel="…" …>` is used for making a relational link to an external resource of type style sheets.

The tag `<script src="…" …>` is used for embedding Java script files.

The tag `<img src="…" …>` is used for embedding images to the page.

Here, `rel="…"` and `src="…"` contain path information for a physical file. The path can be either an absolute (full) path or a relative path.

In the HTML source code in Figure 4, four tags (marked as (1), (2), (3) and (4) with "→") refer to the external resources. Out of these tags, the tags that caused the security message as shown in Figure 3 were $\boxed{\quad F \quad}$.

Answer group

a) (1), (2) and (3)

b) (1), (2), (3) and (4)

c) (1), (2) and (4)

d) (1), (3) and (4)

Q5. Read the following description concerning program development and testing, and then answer Subquestion.

Ministry of Labor (MOL) maintains a statistic file that contains the information about population and employment status by state.
A staff of MOL creates a program which reads the statistic file and outputs a statistic report.

[Program description]

(1) Each record of the statistic file has 4 fields. Those are, from left to right, the name of the state, population, adult population, employed and unemployed. The following is an example of the contents of the statistic file:

State	Population	Adult Population	Employed	Unemployed
East	1240000	1040000	900000	100000
North	2520000	2080000	1760000	240000
South	1320000	1060000	860000	140000
West	2600000	2120000	1680000	320000

(2) For each record, the program calculates and outputs the labor force rate and unemployment rate, by using the following formulas:

Labor force rate (%) = 100 × (Employed + Unemployed) ÷ Adult Population
Unemployment rate (%) = 100 × Unemployed ÷ (Employed + Unemployed)

(3) When all the records are processed, the program outputs the total of population, adult population, employed and unemployed, and the nationwide values of labor force rate and unemployment rate. After that, the program outputs the highest unemployment rate with the name of that state, and the lowest unemployment rate with the name of that state.

(4) The following is an example of the statistic report:

```
State         Pop  AdultPop       Emp     Unemp  Labor%  Unemp%
------  --------  --------  --------  --------  ------  ------
East     1240000   1040000    900000    100000    96.2    10.0
North    2520000   2080000   1760000    240000    96.2    12.0
South    1320000   1060000    860000    140000    94.3    14.0
West     2600000   2120000   1680000    320000    94.3    16.0
------  --------  --------  --------  --------  ------  ------
TOTAL    7680000   6300000   5200000    800000    95.2    13.3

Highest Unemp%: 16.0, State: West
Lowest  Unemp%: 10.0, State: East
```

[Program]

```
   ○ character type: State, StateHigh ← "????", StateLow ← "????"
   ○ integer type:   Pop, AdultPop, Emp, Unemp
   ○ integer type:   TotalPop ← 0, TotalAdultPop ← 0,
                     TotalEmp ← 0, TotalUnemp ← 0
   ○ float type:     LaborRate, UnempRate
X →○ float type:     UnempRateHigh ← 0.0, UnempRateLow ← 100.0

   • OpenFile("statistic")     /* Open the statistic file */
   • Put("State       Pop  AdultPop       Emp     Unemp  Labor%  Unemp%")
   • Put("------  --------  --------  --------  --------  ------  ------")
   • GetRecord(State, Pop, AdultPop, Emp, Unemp)
   ■ Not end-of-file     /* Loop until the statistic file reaches the end-of-file */
   │ • LaborRate ← 100.0 × (Emp + Unemp) ÷ AdultPop
   │ • UnempRate ← 100.0 × Unemp ÷ (Emp + Unemp)
Y →│ ▲ UnempRate > UnempRateHigh
   │ │ • StateHigh ← State
   │ │ • UnempRateHigh ← UnempRate
   │ ▼
Z →│ ▲ UnempRate < UnempRateLow
   │ │ • StateLow ← State
   │ │ • UnempRateLow ← UnempRate
   │ ▼
   │ • TotalPop ← TotalPop + Pop
   │ • TotalAdultPop ← TotalAdultPop + AdultPop
   │ • TotalEmp ← TotalEmp + Emp
   │ • TotalUnemp ← TotalUnemp + Unemp
   │ • Put(State, Pop, AdultPop, Emp, Unemp, LaborRate, UnempRate)
   │ • GetRecord(State, Pop, AdultPop, Emp, Unemp)
   ■
   • Put("------  --------  --------  --------  --------  ------  ------")
   • LaborRate ← 100.0 × (TotalEmp + TotalUnemp) ÷ TotalAdultPop
   • UnempRate ← 100.0 × TotalUnemp ÷ (TotalEmp + TotalUnemp)
   • Put("TOTAL", TotalPop, TotalAdultPop,
                  TotalEmp, TotalUnemp, LaborRate, UnempRate)
   • Put()     /* Print a blank line */
   • Put("Highest Unemp%: ", UnempRateHigh, ", State: ", StateHigh)
   • Put("Lowest  Unemp%: ", UnempRateLow,  ", State: ", StateLow )
   • CloseFile("statistic")     /* Close the statistic file */
```

Note: The function `GetRecord(`v_1, v_2, …`)` reads the next record from the statistic file, and stores the values into variables v_1, v_2, … .
The function `Put(`p_1, p_2, …`)` prints variables/constants p_1, p_2, … in one line.

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank [   ] in the following description.

Note that the shaded parts [   ] are not shown.

The program development is now in test phase, and testing is performed by using various test cases. Four test cases and their test results are shown below:

[Test case 1]

This is the case where two states have the same unemployment rate for both the highest and lowest rates. In this case, the program outputs the following report.

```
State         Pop  AdultPop       Emp     Unemp  Labor%  Unemp%
------  --------  --------  --------  --------  ------  ------
East     1200000   1040000    900000    100000    96.2    10.0
North    1200000   1050000    900000    100000    95.2    10.0
South    1200000   1060000    850000    150000    94.3    15.0
West     1200000   1070000    850000    150000    93.5    15.0
------  --------  --------  --------  --------  ------  ------
TOTAL    4800000   4220000   3500000    500000    94.8    12.5

Highest Unemp%: 15.0, State: [   A   ]
Lowest  Unemp%: 10.0, State: [       ]
```

[Test case 2]

This is the particular case where all the states have 0% unemployment rate. In this case, expected output cannot be obtained. According to the program logic, the states shown on the last two lines are as follows:

```
State         Pop  AdultPop       Emp     Unemp  Labor%  Unemp%
------  --------  --------  --------  --------  ------  ------
East     1200000   1040000    900000         0    86.5     0.0
North    1200000   1050000    900000         0    85.7     0.0
South    1200000   1060000    850000         0    80.2     0.0
West     1200000   1070000    850000         0    79.4     0.0
------  --------  --------  --------  --------  ------  ------
TOTAL    4800000   4220000   3500000         0    82.9     0.0

Highest Unemp%:  0.0, State: [   B   ]
Lowest  Unemp%:  0.0, State: [       ]
```

[Test case 3]

The problem in test case 2 can be resolved by changing the operator “>” to “≥” on line Y, and changing the operator “<” to “≤” on line Z.

After this change, another particular case where all the states have 100% unemployment rate is tested. The problem is resolved, and the program outputs the following report.

```
State        Pop  AdultPop       Emp     Unemp  Labor%  Unemp%
------  --------  --------  --------  --------  ------  ------
East     1200000   1040000         0    900000    86.5   100.0
North    1200000   1050000         0    900000    85.7   100.0
South    1200000   1060000         0    850000    80.2   100.0
West     1200000   1070000         0    850000    79.4   100.0
------  --------  --------  --------  --------  ------  ------
TOTAL    4800000   4220000         0   3500000    82.9   100.0

Highest Unemp%:100.0, State: [        ]
Lowest  Unemp%:100.0, State: [   C    ]
```

[Test case 4]

After the change mentioned in test case 3, test case 1 is tested again. This time, because the program logic is changed, the program outputs the following report.

```
State        Pop  AdultPop       Emp     Unemp  Labor%  Unemp%
------  --------  --------  --------  --------  ------  ------
East     1200000   1040000    900000    100000    96.2    10.0
North    1200000   1050000    900000    100000    95.2    10.0
South    1200000   1060000    850000    150000    94.3    15.0
West     1200000   1070000    850000    150000    93.5    15.0
------  --------  --------  --------  --------  ------  ------
TOTAL    4800000   4220000   3500000    500000    94.8    12.5

Highest Unemp%: 15.0, State: [        ]
Lowest  Unemp%: 10.0, State: [   D    ]
```

Incidentally, the problem mentioned above can be resolved in a different way by replacing the initial values on line X by, for example, the following values:

```
○ float type:   UnempRateHigh ← [  E  ], UnempRateLow ← [  F  ]
```

Answer group for A through D

a) `????` b) `East` c) `North` d) `South` e) `West`

Answer group for E and F

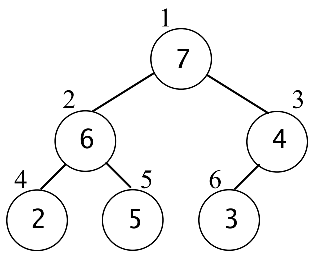
a) `-0.1` b) `0.1` c) `99.9` d) `100.1`

Q6. Read the following description of a heap sort program and the program itself, and then answer Subquestions 1 through 3.

"Heap" is a kind of tree-based data structure. A heap can be defined as a binary tree, in which for each node, the values of its children are less than or equal to the node value (this is called max heap) or greater than or equal to the node value (min heap).
In a max heap, for each node except the root node, the value of the node is less than or equal to the value of its parent. Hence, in a max heap, the root contains the maximum value.

[Program Description]

(1) The program `HeapSort` sorts a given array `A[]` with `n` elements by adopting the heap sort algorithm. Heap sort is a comparison-based sorting algorithm. Figure 1 shows an example of a max heap and its array representation. The node index *i* is associated with `A[`*i*`]`. Here, the index of array `A[]` starts at 1.



Number of elements `n`: 6

(index)	1	2	3	4	5	6
Values in `A[]`:	7	6	4	2	5	3

Figure 1. Example of a max heap and its array representation (heap size = 6)

(2) For any arbitrary node index *i*, the index of its left child is $2 \times i$, the index of its right child is $2 \times i + 1$, and the index of its parent is (integer quotient of) $i \div 2$.

(3) The program `BuildHeap` builds a max heap on the input array `A[]`.

(4) Sorting operation proceeds as follows:

(i) Build the max heap with `n` elements `A[1]`, `A[2]`, … , `A[n]`. Then, the maximum value among them is obtained in the root `A[1]`. So, exchange the value in `A[1]` with the value in `A[n]`. Thus, the maximum value is obtained in `A[n]`.

(ii) Discard the sorted element `A[n]` from the heap. In the heap with `n-1` elements, any parent node except the root node keeps the heap property, but the new root node may violate the heap property as it has been exchanged with the last node.

(iii) Build the max heap with `n-1` elements `A[1]`, `A[2]`, … , `A[n-1]`. Then, the maximum value among them is obtained in `A[1]`. So, exchange the value in `A[1]` with the value in `A[n-1]`. Thus, the next-maximum value is obtained in `A[n-1]`.

(iv) In this way, execute the same process repeatedly for the heap size `n-2` down to the heap size 2. Eventually, `A[]` will contain `n` sorted elements in ascending order.

[Program]

```
○ program: HeapSort(integer type: A[], integer type: n)
  ○ integer type: i, temp

  • BuildHeap(A, n)
  • i ← n
  ■ i ≥ 2
  │ • temp ← A[1]
  │ • A[1] ← A[i]
  │ • A[i] ← temp
  │ • i ← i - 1
  │ • Heapify(A, 1, i)
  ■

○ program: BuildHeap(integer type: A[], integer type: n)
  ○ integer type: i

  ■ i: n ÷ 2, i ≥ 1, -1
  │ • Heapify(A, i, n)
  ■

○ program: Heapify(integer type: A[],
                     integer type: i, integer type: heapsize)
  ○ integer type: lChild, rChild, largest, temp
                /* lChild: left child, rChild: right child */

  • lChild ← [    A    ]
  • rChild ← [    B    ]
  ▲ lChild ≤ heapsize and A[lChild] > A[i]
X→│ • largest ← lChild
  ├──
  │ • largest ← i
  ▼
  ▲ rChild ≤ heapsize and A[rChild] > A[largest]
Y→│ • largest ← rChild
  ▼
  ▲ largest ≠ i
  │ • temp ← A[i]
  │ • A[i] ← A[largest]
  │ • A[largest] ← temp
  │ • Heapify(A, largest, heapsize)
  ▼
```

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank [   ] in the above program.

Answer group

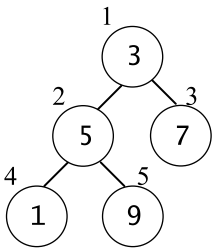
a) `2 × i` b) `2 × i + 1`

c) `2 × i + 2` d) `i ÷ 2`

Subquestion 2

From the answer groups below, select the correct answer to be inserted into each blank [   ] in the following description and Figure 4.

Consider the case where the unsorted data shown in Figure 2 is passed to `HeapSort`.



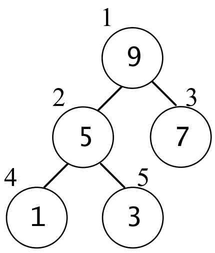
Number of elements `n`: 5

(index)	1	2	3	4	5
values in `A[]`:	3	5	7	1	9

Figure 2. Unsorted data passed to `HeapSort` (heap size = 5)

First, `HeapSort` executes `BuildHeap(A, n)`. When returned from `BuildHeap(A, n)`, the max heap is built in `A[]`, and the root `A[1]` contains the maximum value 9, as shown in Figure 3.

During this process, the program `Heapify` is called several times, and `largest ← lChild` on line [X] is executed [C] time(s), and `largest ← rChild` on line [Y] is executed [D] time(s).



Number of elements `n`: 5

(index)	1	2	3	4	5
Values in `A[]`:	9	5	7	1	3

Figure 3. The max heap is built in `A[]` (heap size = 5)

Next, `HeapSort` executes the following 3 steps repeatedly in the loop, while `i ≥ 2`.

Step 1: Exchange the values of `A[1]` and `A[i]`, and `A[i]` becomes the sorted element.

Step 2: Decrement the heap size `i` by 1.

Step 3: Execute `Heapify(A, 1, i)` to set the next maximum value in root `A[1]`.

Figure 4 shows the contents of `i` and `A[]` at some checkpoints in the loop. This shows how the sorting operation proceeds. Here, shaded parts ▭ indicate the sorted elements.

Checkpoint	Value `i`	Values in `A[]` 1	2	3	4	5
start of the loop (as in Figure 3)	5	9	5	7	1	3
after step 1 (1st time)	5	3	5	7	1	9 (shaded)
after step 3 (1st time)	4	E				9 (shaded)
after step 1 (2nd time)	4	1	5	3	7 (shaded)	9 (shaded)
after step 3 (2nd time)	3	F			7 (shaded)	9 (shaded)
after step 1 (3rd time)	3	3	1	5 (shaded)	7 (shaded)	9 (shaded)
after step 3 (3rd time)	2	3	1	5 (shaded)	7 (shaded)	9 (shaded)
after step 1 (4th time)	2	1	3 (shaded)	5 (shaded)	7 (shaded)	9 (shaded)
after step 3 (4th time)	1	1	3 (shaded)	5 (shaded)	7 (shaded)	9 (shaded)

Figure 4. Contents of `i` and `A[]` at the checkpoints in the loop

Answer group for C and D

a) 0   b) 1   c) 2   d) 3   e) 4

Answer group for E

a) | 1 | 5 | 7 | 3 |

b) | 1 | 7 | 5 | 3 |

c) | 3 | 1 | 5 | 7 |

d) | 3 | 5 | 1 | 7 |

e) | 7 | 3 | 5 | 1 |

f) | 7 | 5 | 3 | 1 |

Answer group for F

a) | 1 | 3 | 5 |

b) | 1 | 5 | 3 |

c) | 3 | 1 | 5 |

d) | 3 | 5 | 1 |

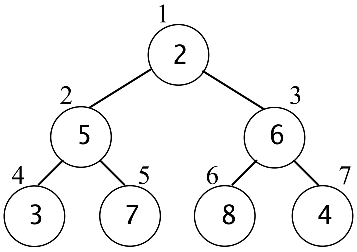
e) | 5 | 1 | 3 |

f) | 5 | 3 | 1 |

Subquestion 3

From the answer group below, select the correct answer to be inserted into the blank $\boxed{}$ in the following description.

Consider another case where the unsorted data shown in Figure 5 is passed to `HeapSort`.



Number of elements `n`: 7

(index)	1	2	3	4	5	6	7
Values in `A[]`:	2	5	6	3	7	8	4

Figure 5. Unsorted data passed to `HeapSort` (heap size = 7)

`HeapSort` first executes `BuildHeap(A, n)`. When returned from `BuildHeap(A, n)`, the max heap is built in `A[]`, and the values in `A[]` are as follows:

(index)	1	2	3	4	5	6	7
Values in `A[]`:	G						

Answer group

a)	8	5	7	2	3	4	6
b)	8	5	7	2	3	6	4
c)	8	7	6	3	5	2	4
d)	8	7	6	3	5	4	2
e)	8	7	6	5	3	2	4
f)	8	7	6	5	3	4	2

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestion.

This program is a simple text editor which is able to perform editing operations on a line of text.

[Program Description]

(1) The editor first gets a line of text to edit. Then, the editor repeatedly accepts editor commands "`D`" (delete), "`F`" (find), or "`I`" (insert), until it receives "`Q`" (quit) command.

(2) The editor command "`D`" deletes a specified character string from the text, "`F`" finds a specified character string in the text, and "`I`" inserts a specified character string into the text at a specified location.

(3) The following 5 functions are used in the program.

`char * deleted(char *source, int index, int n)`

The function deletes `n` characters starting from `source[index]`, and returns the resulted `source`. If `source` is too short for full deletion, characters from the position `index` to the end of the text are deleted.

`char * do_edit(char *source, char command)`

The function performs the editing operation (delete, find, or insert) according to `command` ("`D`", "`F`", or "`I`"), and returns the resulted `source`.

`char get_command(void)`

The function gets a character representing an editor command, and returns the command as the uppercase character. The succeeding input characters are ignored.

`char * insert(char *source, const char *to_insert, int index)`

The function inserts the character string `to_insert` to the position `index`, and returns the resulted `source`. If `source` is too short and the position `index` does not exist, `to_insert` is added after the end of the text.

`int pos(const char *source, const char *to_find)`

The function returns the position of the first occurrence of `to_find` in `source`, or the value `NOT_FOUND` if `to_find` is not found in `source`.

(4) Assuming that the length of any input character string and the length of `source` after editing operations range between 1 and 99. Also, it is assumed to be no errors in input character strings and arguments. The editor does not check for overflows.

(5) The following is a sample run of the text editor program. For each input character string, to show whether it contains space characters or not, an underline ("_") is used instead of a space (" ").

```
Enter the source string:
> Internet_use_is_growing_rapidly.
Enter D(Delete), I(Insert), F(Find), or Q(Quit)> d
String to delete> _growing
New source: Internet_use_is_rapidly.

Enter D(Delete), I(Insert), F(Find), or Q(Quit)> f
String to find> .
'.' found at position 23
New source: Internet_use_is_rapidly.

Enter D(Delete), I(Insert), F(Find), or Q(Quit)> i
String to insert> _expanding
Position of insertion> 23
New source: Internet_use_is_rapidly_expanding.

Enter D(Delete), I(Insert), F(Find), or Q(Quit)> q
String after editing: Internet_use_is_rapidly_expanding.
```

(6) The following library functions are used in the program.

`char * strcat(char *str1, const char *str2)`

Appends the string pointed to by `str2` to the end of the string pointed to by `str1`.

`int strcmp(const char *str1, const char *str2)`

Compares the string pointed to by `str1` with the string pointed to by `str2`.

`char * strcpy(char *s1, const char *s2)`

Copies the string pointed to by `str2` to the string pointed to by `str1`.

`size_t strlen(const char *str)`

Computes the length of the string `str`, excluding the terminating null character.

`char * strncpy(char *s1, const char *s2, size_t n)`

Copies up to `n` characters from the string pointed to by `str2` to the string pointed to by `str1`.

`int toupper(int c)`

If `c` is an alphabetical lowercase character, converts it to the uppercase character.

[Program]

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_LEN 100
#define NOT_FOUND -1

char * deleted(char *source, int index, int n);
char * do_edit(char *source, char command);
char get_command(void);
char * insert(char *source, const char *to_insert, int index);
int pos(const char *source, const char *to_find);

int main(void) {
   char source[MAX_LEN], command;

   source[0] = '\0';
   printf("Enter the source string: \n> ");
   fgets(source, sizeof(source), stdin);
   source[strlen(source)-1] = '\0';
   for (command = get_command();
        command != 'Q'; command = get_command()) {
      do_edit(source, command);
      printf("New source: %s\n\n", source);
   }
   printf("String after editing: %s\n", source);
   return (0);
}

char * deleted(char *source, int index, int n) {
   char rest_str[MAX_LEN];

   if (strlen(source) <= index + n) {
      source[index] = '\0';
   }
   else {
      strcpy(rest_str, &source[index + n]);
      [      A      ];
   }
   return (source);
}
```

```
char * do_edit(char *source, char command) {
   char str[MAX_LEN], buf[4];
   int index;

   switch (command) {
      case 'D':
         printf("String to delete> ");
         fgets(str, sizeof(str), stdin);
         str[strlen(str)-1] = '\0';
         index = pos(source, str);
         if (index == NOT_FOUND)
            printf("'%s' not found\n", str);
         else
            deleted(source, index, strlen(str));
         break;
      case 'I':
         printf("String to insert> ");
         fgets(str, sizeof(str), stdin);
         str[strlen(str)-1] = '\0';
         printf("Position of insertion> ");
         fgets(buf, sizeof(buf), stdin);
         sscanf(buf, "%d", &index);
         [      B      ];
         break;
      case 'F':
         printf("String to find> ");
         fgets(str, sizeof(str), stdin);
         str[strlen(str)-1] = '\0';
         index = [      C      ];
         if (index == NOT_FOUND)
            printf("'%s' not found\n", str);
         else
            printf("'%s' found at position %d\n", str, index);
         break;
      default:
         printf("Invalid edit command '%c' \n", command);
   }
   return (source);
}
```

```
char get_command(void) {
   char command, buf[100];

   printf("Enter D(Delete), I(Insert), F(Find), or Q(Quit)> ");
   fgets(buf, sizeof(buf), stdin);
   buf[strlen(buf)-1] = '\0';
   sscanf(buf, "%c", &command);
   return (toupper(command));
}

char * insert(char *source, const char *to_insert, int index) {
   char rest_str[MAX_LEN];

   if (strlen(source) <= index) {
      strcat(source, to_insert);
   }
   else {
      [      D      ];
      strcpy(&source[index], to_insert);
      strcat(source, rest_str);
   }
   return (source);
}

int pos(const char *source, const char *to_find) {
   int i = 0, find_len, found = 0, position;
   char substring[MAX_LEN];

   [      E      ];
   while (!found && i <= (int)strlen(source) - find_len) {
      strncpy(substring, &source[i], find_len);
      [      F      ];
      if (strcmp(substring, to_find) == 0)
         found = 1;
      else
         ++i;
   }
   if (found)
      position = i;
   else
      position = NOT_FOUND;
   return (position);
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the above program.

Answer group for A and D

a) `strcpy(&source[index], rest_str)`

b) `strcpy(rest_str, &source[index])`

c) `strcpy(rest_str, source[index])`

d) `strcpy(source[index], rest_str)`

Answer group for B

a) `insert(source, str, index)`
b) `insert(str, source, index)`

c) `strcpy(source, str)`
d) `strcpy(str, source)`

Answer group for C

a) `pos(*source, *str)`
b) `pos(*str, *source)`

c) `pos(source, str)`
d) `pos(str, source)`

Answer group for E

a) `find_len = (int)strlen(source)`
b) `find_len = (int)strlen(to_find)`

c) `position = (int)strlen(source)`
d) `position = (int)strlen(to_find)`

Answer group for F

a) `source[find_len] = '\0'`
b) `source[find_len] = '\n'`

c) `substring[find_len] = '\0'`
d) `substring[find_len] = '\n'`

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestion.

[Program Description]

Rock-paper-scissors is a hand game played by two (or more) people, where each player simultaneously forms one of three shapes ("rock", "scissors" and "paper") with his/her outstretched hand. The rock beats scissors, the scissors beats paper, and the paper beats rock. If both players throw the same shape, the game is tied.

This game is implemented as two-player game; single player vs. computer. Computer chooses the shape randomly. A winner gets one point. After playing three times, the program ends automatically.

The program is composed of the following interface and classes:

`GameLoop`	The interface defining the actions to start game, play game, and end game.
`Player`	The class describing the player. It saves the name and score of the player. It can sort the players by their scores in descending order.
`Game`	The main class describing game playing options. It implements `GameLoop`. It can have as many as players (`List<Player> players`).
`GamePlayer`	Inherited from the class `Player`. It holds the player's chosen shape (`choice`). If the chosen shape is rock, the value of `choice` is 0. If the chosen shape is paper, the value of `choice` is 1. If the chosen shape is scissors, the value of `choice` is 2.
`GameLogic`	The class describing the game logic. The method `checkWinner()` determines the player who beats.
`RoPaScGame`	Inherited from the class `Game`. It implements the "RockPaperScissors" game.
`TestGame`	The class for testing the game.

The program uses the following java classes from `java.util` and `java.io` packages.

`ArrayList`	Resizable-array implementation of the `List` interface. Implements all optional list operations, and permits all elements, including `null`. In addition to implementing the `List` interface, this class provides methods to manipulate the size of the array that is used internally to store the list.
`Collections`	The class consists exclusively of static methods that operate on or return collections. It contains polymorphic algorithms that operate on collections, "wrappers", which return a new collection backed by a specified collection, and a few other odds and ends.

`List` An ordered collection (also known as a sequence). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.

`BufferedReader` It reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines. The method `readLine()` reads a line of text.

`InputStreamReader` An `InputStreamReader` is a bridge from byte streams to character streams. Each invocation of one of an `InputStreamReader`'s method `read()` may cause one or more bytes to be read from the underlying byte-input stream. To enable the efficient conversion of bytes to characters, more bytes may be read ahead from the underlying stream than are necessary to satisfy the current read operation. For top efficiency, consider wrapping an `InputStreamReader` within a `BufferedReader`.

When the program is executed, the following result, for example, will be printed out.

```
Input r (rock), p (paper) or s (scissors)
_________________________________________
Input your choice
r
Winner is Player
_________________________
Input your choice
p
Winner is Computer
_________________________
Input your choice
s
Winner is Player
_________________________
Player vs. Computer
Player's score :        2
Computer's score :      1
Player won.
_________________________
```

[Program 1]

```
public interface GameLoop {
   public void startGame();
   public void playGame();
   public void endGame();
}
```

[Program 2]

```
// Object list of this class can be sorted by score.
public class Player implements [    A    ] {
   private String name;
   private int score;
   public Player(String name) {
      setName(name);
      score = 0;
   }
   public void setName(String name) {
      this.name = name;
   }
   public int getScore() {
      return score;
   }
   public void setScore(int score) {
      this.score = score;
   }
   // This is overridden method, that used to sort players by their scores in descending order.
   public int compareTo(Player p) {
      return p.getScore() - this.score;
   }
   public String toString() {
      return name;
   }
}
```

[Program 3]

```
import java.util.ArrayList;
import java.util.List;
[    B    ] class Game implements GameLoop {
   protected List<Player> players;
   public Game() {
      players = new ArrayList<Player>();
   }
   public void play() {
      startGame();
   }
}
```

[Program 4]

```
public class GamePlayer extends Player {
    private int choice;
    public GamePlayer(String name) {
        super(name);
    }
    public void incScore() {
        setScore(getScore() + 1);
    }
    public int getChoice() {
        return choice;
    }
    public void setChoice(int choice) {
        this.choice = choice;
    }
}
```

[Program 5]

```
public class GameLogic {
    // If game is tied, this method return null value,
    // otherwise returns the information about the player who beats.
    [      C      ] GamePlayer checkWinner(GamePlayer player1,
                                           GamePlayer player2) {
        GamePlayer winner;
        if (player1.getChoice() == player2.getChoice())
            return null;
        else {
            if (player1.getChoice() < player2.getChoice()) {
                if ([      D      ] == 1)
                    winner = player2;
                else
                    winner = player1;
            }
            else {
                if ([      D      ] == 1)
                    winner = player1;
                else
                    winner = player2;
            }
            winner.incScore();
            return winner;
        }
    }
}
```

[Program 6]

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Collections;
public class RoPaScGame extends Game {
   // Read data from keyboard input stream.
   public BufferedReader reader;
   private final int PLAYCOUNT = 3;
   public RoPaScGame() {
      super();
      InputStreamReader input = new InputStreamReader(System.in);
      reader = new BufferedReader(input);
   }
   public void initGame() {
      players.clear();
      players.add(new GamePlayer("Computer"));
      players.add(new GamePlayer("Player"));
   }
   public void startGame() {
      initGame();
      System.out.println("Input r (rock), p (paper) or s (scissors)");
      for (int count = 0; count < PLAYCOUNT; count++) {
         playGame();
      }
      endGame();
   }
   public void playGame() {
      GamePlayer computer = (GamePlayer) players.get(0);
      GamePlayer player = (GamePlayer) players.get(1);
      computer.setChoice((int)(Math.random() * 3));
      boolean isCorrect = false;
      System.out.println("_________________________");
      while (!isCorrect) {
         System.out.println("Input your choice");
         char choice = readInput();
         switch (choice) {
            case 'r': player.setChoice(0);
                      isCorrect = true;
                      break;
            case 'p': player.setChoice(1);
                      isCorrect = true;
                      break;
            case 's': player.setChoice(2);
                      isCorrect = true;
                      break;
```

```
            default:  System.out.println("Wrong shape");
                      break;
         }
      }
      GamePlayer winner = GameLogic.checkWinner(player,computer);
      if (winner != null)
         System.out.println("Winner is " + winner);
      else
         System.out.println("The game is tied");
   }
   public void endGame() {
      [     E     ].sort(players);
      Player p1 = players.get(0);
      Player p2 = players.get(1);
      System.out.println("_________________________");
      System.out.println(p1 + " vs. " + p2);
      System.out.println(p1 + "'s score :\t" + p1.getScore());
      System.out.println(p2 + "'s score :\t" + p2.getScore());
      if (p1.getScore() != p2.getScore())
         System.out.println(p1 + " won.");
      else
         System.out.println("End in a tie.");
      System.out.println("_________________________");
   }
   public char readInput() {
      String result = null;
      try {
         result = reader.readLine();
      }
      catch (Exception e) {
      }
      return result.charAt(0);
   }
}
```

[Program 7]

```
public class TestGame {
   public static void main(String[] args) {
      Game game = new RoPaScGame();
      game.play();
   }
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank in the above programs.

Answer group for A and E

a) ArrayList
b) Collection<Player>
c) Collections
d) Comparable<Player>
e) Comparables
f) Comparator
g) Comparator<Player>
h) List

Answer group for B and C

a) private
b) private abstract
c) private final
d) private static
e) public
f) public abstract
g) public final
h) public static

Answer group for D

a) Math.abs(player1.getChoice() - player2.getChoice())
b) Math.min(player1.getChoice(), player2.getChoice())
c) player1.getChoice()
d) player1.getChoice() - player2.getChoice()
e) player2.getChoice()
f) Player2.getChoice() - player1.getChoice()

19th ITPEC FE Afternoon Examination, May 2015

Question No	Subquestion No		Correct Answer
1	1	A	f
		B	e
		C	e
	2	D	d
		E	a
		F	b
2	1	A	d
		B	d
		C	b
	2	D	g
		E	c
		F	d
3	1	A	c
		B	d
	2	C	b
		D	c
4	1	A	c
		B	b
		C	d
	2	D	a
		E	b
	3	F	c
5		A	d
		B	a
		C	e
		D	c
		E	a
		F	d
6	1	A	a
		B	b
	2	C	b
		D	c
		E	f
		F	e
	3	G	c
7		A	a
		B	a
		C	c
		D	b
		E	b
		F	c
8		A	d
		B	f
		C	h
		D	a
		E	c

[Note] Answers for
A and B can be exchanged.

October 2014

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a 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 answer 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) **Question Selection**
   For **Q7** and **Q8**, mark the S of the question you select to answer in the "Selection Column" on your answer sheet.
   (4) **Answers**
   Mark your answers as shown in the following sample question.

   [Sample Question]
   In which month is the autumn Fundamental IT Engineer Examination conducted?

   Answer group
   a) September   b) October   c) November   d) December

   Since the correct answer is " b) October ", mark your answer sheet as follows:

   [Sample Answer]

Sample	a	●	c	d	e	f	g	h	i	j

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

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated.

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc. of procedures, variables, etc.
/* text */		Describes comments in the text.
Process	• variable ← expression	Assigns the value of the expression to the variable.
	• procedure(argument, ...)	Calls the procedure and passes / receives the argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	▲ conditional expression process 1 ――― process 2 ▼	Indicates a two-way selection process. If the conditional expression is `true`, then the process 1 is executed. If it is `false`, then the process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	+, −, not	High
Multiplication, division	×, ÷, %	↑
Addition, subtraction	+, −	
Relational operation	>, <, ≥, ≤, =, ≠	
Logical product	and	↓
Logical sum	or	Low

Note: With division of integers, integer quotient is returned as a result.
The % operator indicates a remainder operation.

Notations used for E-R diagrams

In questions that use E-R diagrams, the following notations are used unless otherwise stated.

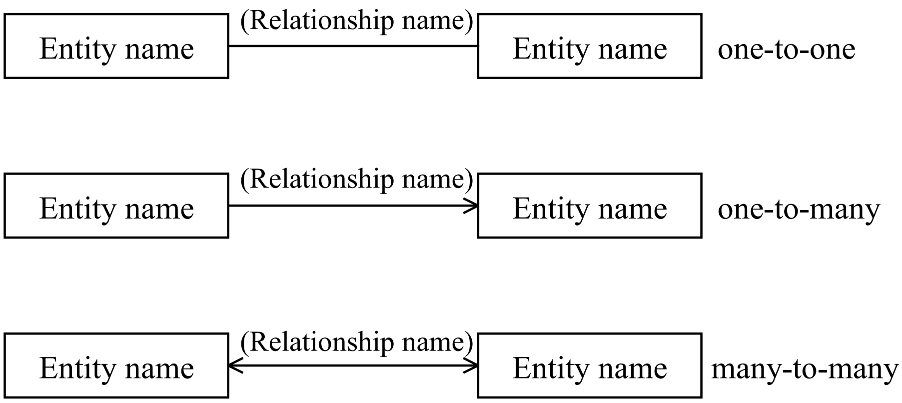


Figure Notations used for entities and relationships

1. Entities are represented by rectangles.
2. Entity names are indicated within the rectangles.
3. A relationship between entities is represented by a line.
   The relationship name is indicated at the side of the line as “(Relationship name)”.
   The relationship name can be omitted.
4. A “one-to-one” relationship is represented by a line.
   A “one-to-many” relationship is represented by a line with an arrow pointing towards the “many” side.
   A “many-to-many” relationship is represented by a line with an arrow on both ends.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description concerning electronic circuits, and then answer Subquestions 1 and 2.

Flip-flop circuits are used for designing computer hardware such as CPU, memory, counter etc. J-K flip-flop is one of the most commonly used such type of flip-flop circuits.
Figure 1 shows the block diagram of J-K flip-flop, and Table 1 shows the characteristic table of J-K flip-flop.

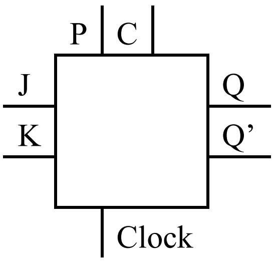


Figure 1. Block diagram of J-K flip-flop

Table 1. Characteristic table of J-K flip-flop

Inputs					Output
P (Preset)	C (Clear)	J	K	Clock	Q
1	1	0	0	-	No change
1	1	0	1	-	Reset
1	1	1	0	-	Set
1	1	1	1	-	Toggle
1	0	any	any	-	0
0	1	any	any	-	1

The outputs Q and Q' varies according to the values of J and K. Q and Q' are always inverse of each other. Toggle means the value is inversed, that is, if the value of Q is 0 then after toggle it becomes 1, and vice versa. Reset and Set means the value becomes 0 and 1 respectively.
Clock supplies the values 0 and 1 alternately and cyclically. When Clock rises (goes 0 to 1), the J-K flip-flop obtains the values of J and K. When Clock falls (goes 1 to 0), the J-K flip-flop outputs Q and Q' according to the last-obtained values of J and K.
If P=1 and C=0, or if P=0 and C=1, the value of Q is set as shown in Table 1, regardless of the values of J and K.

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank [   ] in the following description.

(1) Assuming that, at the last clock rise, the input values were P=1, C=1, J=0 and K=1. Then, at the subsequent clock fall, the output values will be [A].

(2) Figure 2 shows an example of the input and output wave forms of the J-K flip-flop. In Figure 2, the wave form in shaded part (1) is [B], and the wave form in shaded part (2) can be either [C] or [D].
Here, P and C will have the values P=1 and C=1 throughout the given clock cycles.

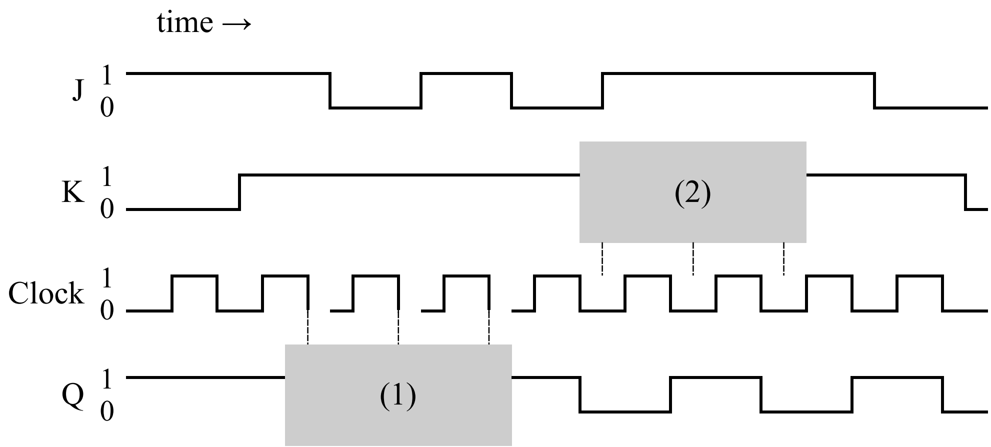
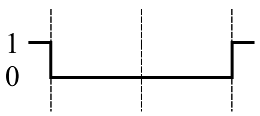


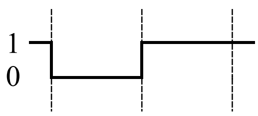
Figure 2. Example of input and output wave forms of J-K flip-flop

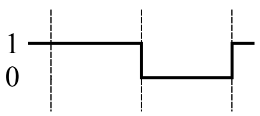
Answer group for A

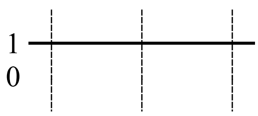
a) Q = 0 and Q' = 0
b) Q = 0 and Q' = 1
c) Q = 1 and Q' = 0
d) Q = 1 and Q' = 1

Answer group for B through D

a) 

b) 

c) 

d) 

Subquestion 2

From the answer groups below, select the correct answer to be inserted into the blank $\boxed{\quad\quad}$ in Table 2.

In designing electronic devices, J-K flip-flops can be used to realize a counter.
Figure 3 shows a 4-bit counter that uses 4 J-K flip-flops. All the inputs of P, C, J and K are 1s. On Clock input line on FF1, the cyclic clock pulse is supplied. However, on each Clock input line on FF2, FF3 and FF4, instead of the cyclic clock pulse, the output from the previous J-K flip-flop is supplied.

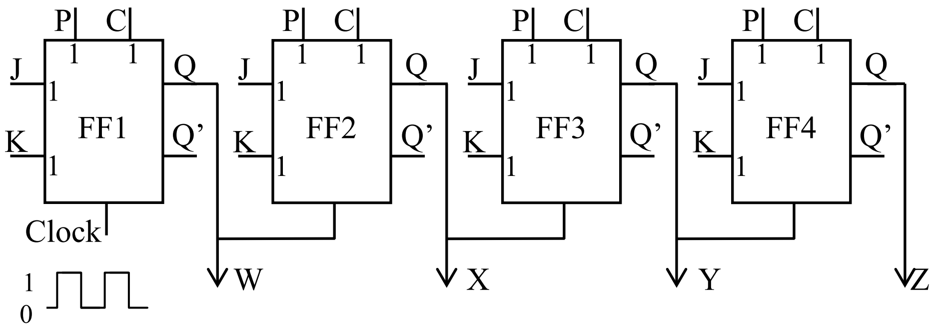


Figure 3. 4-bit counter that uses 4 J-K flip-flops

Assuming that, at the clock cycle *c*, the initial values of W, X, Y and Z are 1, 0, 0 and 0 respectively. Table 2 shows how the output values change on and after the clock cycle *c*.

Table 2. Output values on and after the clock cycle *c*

Clock Cycle	W	X	Y	Z
c	1	0	0	0
$c+1$	0	1	0	0
$c+2$				
$c+3$	E			

(Note) Shaded part is not shown

Answer group

a)	0	0	0	1	b)	0	0	1	0
c)	0	1	0	1	d)	1	0	1	0

Q2. Read the following description concerning a virtual memory system, and then answer Subquestion. In this question, "…h" denotes a hexadecimal value. For example, 100h denotes a hexadecimal value 100 (decimal value 256).

The basic idea behind virtual memory is to run a program whose size may exceed the physical memory size available for it. The operating system keeps those parts of the program currently in use in the physical memory, and the rest on the disk, by swapping pieces of the program between the physical memory and the disk as needed.

In a typical virtual memory system, a virtual address is split into a virtual page number (high-order bits) and a byte offset (low-order bits). Figure 1 shows an example of a simple virtual memory system. In this example, a 16-bit virtual address and a 4k-byte page size is used. Therefore, the virtual addresses range from 0 to [A], and the high-order 4 bits could specify one of the [B] virtual pages, and the low-order 12 bits would then specify the byte offset (0 to FFFh) within the virtual page.

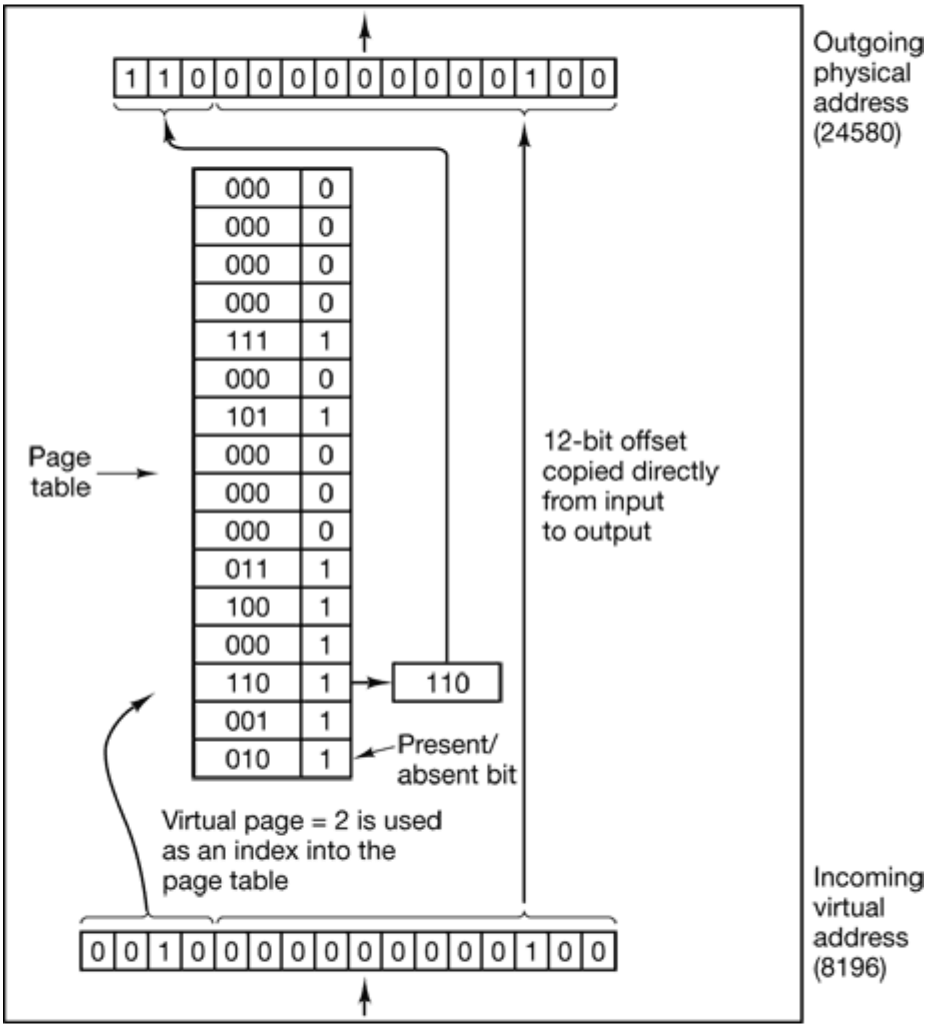


Figure 1. Example of a simple virtual memory system

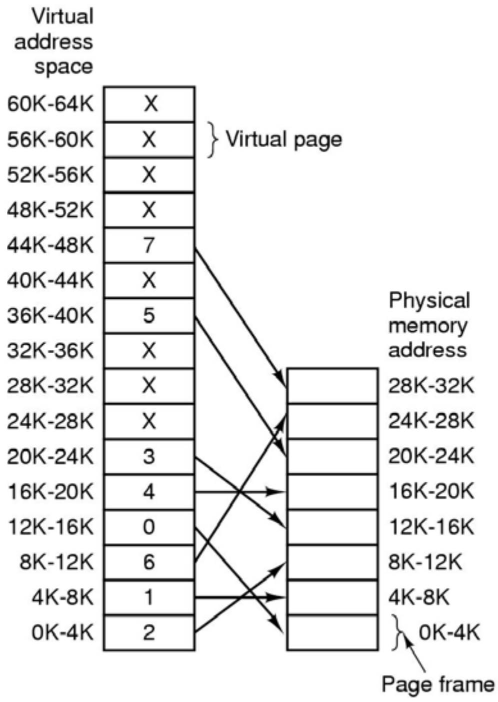


Figure 2. Snapshot of the page table

The relation between virtual addresses and physical addresses is given by the page table. Assuming a case in which program *A* is running on the virtual memory system shown in Figure 1. The size of program *A* is C000h bytes. A snapshot of the page table during the program execution is shown in Figure 2. At this point, 8 virtual pages are loaded into the physical memory, and the virtual address B300h corresponds to the physical address $\boxed{\quad C \quad}$.

There are two major page replacement algorithms used for paging:

(1) FIFO (First In First Out)

When a page fault occurs, page-out the page whose page-in time was the oldest, and page-in the required page.

(2) LRU (Least Recently Used)

When a page fault occurs, page-out the page that has been unused for the longest time, and page-in the required page.

Consider the case of program *A* again.

(1) Before the program starts, no virtual pages are loaded into the physical memory.

(2) After the program starts, the virtual pages are accessed in the following sequence:

$$3 \rightarrow 1 \rightarrow 0 \rightarrow 5 \rightarrow 4 \rightarrow 9 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 11 \rightarrow 6$$

(▲)

Figure 2 shows the snapshot of the page table at the point (▲) in the above sequence.

(3) After the point (▲), when the virtual page 6 is accessed, a page fault occurs. If the virtual memory system uses FIFO algorithm, the page $\boxed{\quad D \quad}$ is paged-out, and the virtual page 6 is paged-in. If the virtual memory system uses LRU algorithm, the page $\boxed{\quad E \quad}$ is paged-out, and the virtual page 6 is paged-in.

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{\qquad}$ in the above description.

Answer group for A

a) FFFh	b) 1FFFh	c) 7FFFh	d) FFFFh

Answer group for B

a) 8	b) 16	c) 128	d) 256

Answer group for C

a) 0300h	b) 1300h	c) 2300h	d) 3300h
e) 4300h	f) 5300h	g) 6300h	h) 7300h

Answer group for D and E

a) 0	b) 1	c) 2	d) 3
e) 4	f) 5	g) 9	h) 11

Q3. Read the following description concerning a relational database, and then answer Subquestions 1 and 2.

An IT faculty of a university is going to develop the system to manage all materials of different subjects. Ms. A is in charge of getting requirements for this system. She identifies some main requirements:

(1) The system records each subject with the subject name, text description of objectives, list of text books and lecture slides, and the list of contents. Each subject has many exercises to evaluate the students' competence.

(2) The system records information about each lecturer, such as the name, home address, home phone number, mobile phone number, e-mail address.

(3) Each text book and lecture slide has the title, author name, published year, the summary of content, and file name and its path where the content is stored.

(4) The system records information about each exercise, such as the lecturer who created it, created date, time for doing (in hours, days, or weeks), type of exercise (individual homework, individual test, team assignment, team project), list of questions and answers. Each exercise corresponds to one subject.

(5) Each subject has different courses. The system records information about each course, such as the start date, end date, the number of students who attended the course, and the number of passed students.

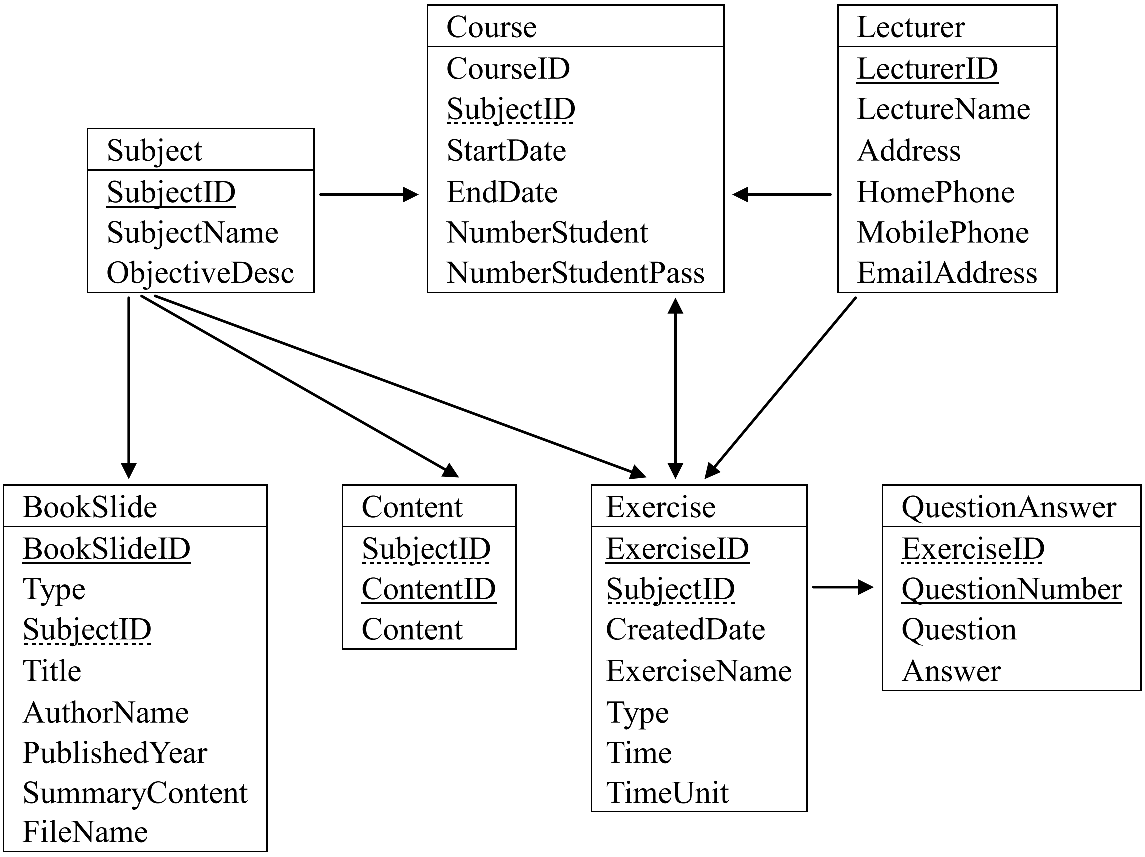
(6) During the course, the main lecturer can make the new exercises or use the old ones. The history of exercises used for each course is recorded. In order to evaluate the difficulty level of each exercise, the average grade of students of each course are also recorded.

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank [ ] in the following description.

Based on the above requirements, Ms. A is planning to create a database for the system. She first created the E-R Diagram of the database (incomplete) as shown in Figure 1.

To complete the E-R Diagram, the attribute LecturerID should be added to the entity [A] and entity [B], as a foreign key.



(Note) A solid underline ______ indicates an attribute of a primary key.
A dashed underline .... indicates an attribute of a foreign key.

Figure 1. The E-R Diagram of the database (incomplete)

Answer group

a) BookSlide
b) Content
c) Course
d) Exercise
e) QuestionAnswer
f) Subject

Subquestion 2

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the following SQL statement.

Based on the E-R Diagram completed in Subquestion 1, a database is created for the material management system.

Recently, the lecturers requested of Ms. A to provide an SQL statement that lists all exercises used for a specified subject, as shown below. The list includes the exercise name, average grades of students, and use count (the number of courses which use that exercise).

Exercise Name	Average Grades	Use Count
Class Diagram	65	3
E-R Diagram	79	2
SQL coding	84	5

Ms. A first creates a new table CourseExercise that contains the average grades of students by course and exercise, as shown below.

CourseExercise
CourseID ExerciseID AverageGrade

Ms. A then creates the following SQL statement. Here, SubjectID of the specified subject is given by the host variable `:SubjectID`.

```
SELECT Exercise.ExerciseName,
       AVG([    C    ]) AS AverageGrades,
       COUNT(*) AS UseCount
FROM Exercise, CourseExercise
     WHERE [    D    ]
     AND [    E    ]
GROUP BY Exercise.ExerciseID, ExerciseName
```

Answer group for C

a) `Course.CourseID`
b) `Course.SubjectID`
c) `CourseExercise.AverageGrade`
d) `CourseExercise.CourseID`
e) `Exercise.ExerciseID`
f) `Exercise.SubjectID`

Answer group for D and E

a) `Course.SubjectID = Exercise.SubjectID`
b) `CourseExercise.ExerciseID = Exercise.ExerciseID`
c) `CourseExercise.CourseID = Course.CourseID`
d) `Exercise.SubjectID = :SubjectID`
e) `Subject.SubjectID = :SubjectID`

Q4. Read the following description concerning a network for a small office, and then answer Subquestion.

A simple IP network using private IP addresses (192.168.1.0/24) is setup for a small office as shown in Figure 1. Some computers in the office IP network access the Internet through an ADSL router with Network Address Translation (NAT) function and Packet Filtering Firewall enabled. The IP Routing table, NAT table, and Packet Filtering Firewall table of ADSL router are correctly configured according to the setup guideline provided by the Internet Service Provider (ISP).

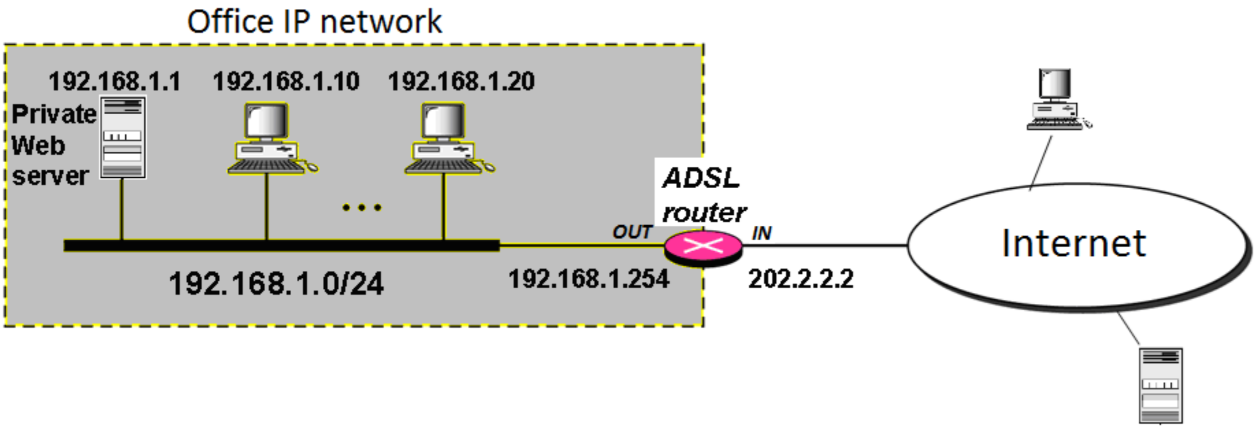


Figure 1. Configuration of office IP network

The network settings are checked according to the following three steps.

The first step is to verify the routing table entry of each computer in the office IP network. Table 1 shows a part of the routing table entries. In Table 1, the entry "Local access" is for direct communication within the office IP network, and the entry "Internet access" is for communication with computers on the Internet.

Table 1. Routing table entries of each computer

	Destination Address	Netmask	Gateway	…
Local access	192.168.1.0	A		…
Internet access	0.0.0.0	0.0.0.0	B	…

(Note) Shaded part is not shown

The second step is to monitor the Internet usage whether it complies with the office Internet usage policy or not. The information in the NAT table of ADSL router can be used to simply track current Internet usage. A snapshot of the NAT table of ADSL router is shown in Table 2.

Table 2. Information in the NAT table of ADSL router

Private address	Private port	External address	External port	NAT port	Protocol
192.168.1.12	12345	205.5.5.1	53	1003	UDP
192.168.1.12	62345	209.85.231.104	80	1005	TCP
192.168.1.1	80	60.1.1.13	31245	1006	TCP
192.168.1.10	21392	205.5.5.1	53	1004	UDP
192.168.1.10	20001	203.151.20.61	80	1005	UDP
192.168.1.20	12345	210.12.1.1	25	1007	TCP

From Table 2, it is found that the number of computers that is accessing external HTTP Web servers is C, because the default protocol and port number of HTTP Web server are TCP and port 80 respectively.
In case of the access to the external server 203.151.20.61 from the computer 192.168.1.10, the source port, destination port, source IP, and destination IP seen by the external server are 1005, 80, D, and E respectively.

The last step is to set firewall rules in ADSL router to block or allow Internet usage traffic according to the office Internet usage policy. Table 3 shows the structure of firewall table of ADSL router.

Table 3. Structure of firewall table of ADSL router

Arrive on Interface	Source		Destination		Protocol	Action
	IP Address	Port	IP Address	Port		

- "Arrive on Interface" field should specify either "IN" or "OUT". "IN" is the interface connecting to the Internet, and "OUT" is the interface connecting to the office IP network.
- "Protocol" field should specify the transport (or other) protocol the rule is applied to (if relevant).
- "Action" field should specify either "Drop" or "Accept".
- For any field in the table, "*" can be used to indicate "any value". For example, if the rule is independent of the protocol, specify "*" in Protocol field.

The last record in Table 2 reveals that the direct access to external email server running at port 25 is currently opened. Thus any virus-infected program within the office IP network can easily send spam mail through this security hole. Since all email accesses in the office are done via Web-based email services using HTTP port 80, it is not necessary to open the direct email server access at port 25. To block such outbound traffic from any computer within the office IP network, a packet filtering rule should be added to the firewall table. The rule should contain the field values $\boxed{\quad F \quad}$ (each value corresponds to each field in Table 3, from left to right).

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{\qquad}$ in the above description and Table 1.

Answer group for A, B, D and E

a) 0.0.0.0 b) 192.168.1.0 c) 192.168.1.254 d) 202.2.2.2
e) 203.151.20.61 f) 255.255.255.0 g) 255.255.255.192

Answer group for C

a) 0 b) 1 c) 2 d) 3

Answer group for F

a) IN, *, *, *, 25, TCP, Accept
b) IN, *, *, *, 25, TCP, Drop
c) IN, *, 25, *, *, TCP, Accept
d) IN, *, 25, *, *, TCP, Drop
e) OUT, *, *, *, 25, TCP, Accept
f) OUT, *, *, *, 25, TCP, Drop
g) OUT, *, 25, *, *, TCP, Accept
h) OUT, *, 25, *, *, TCP, Drop

Q5. Read the following description concerning wireless network security, and then answer Subquestion.

Company *X* is an e-commerce company selling various products online. With only 10 employees, the company is relatively small. Its e-commerce system consists of a Web server and a mail server. Recently, the company was renovated and its servers were outsourced to an external entity to reduce the hardware and infrastructure cost, leaving only client computers in the network.

After the renovation, the employees requested that they want to use their own portable computers and smart devices at work. Mr. *A*, the owner of Company *X*, agreed with the employees and installed a wireless access point (hereinafter, AP) in the company instead of upgrading the client computers. The current network configuration of Company *X* is shown in Figure 1.

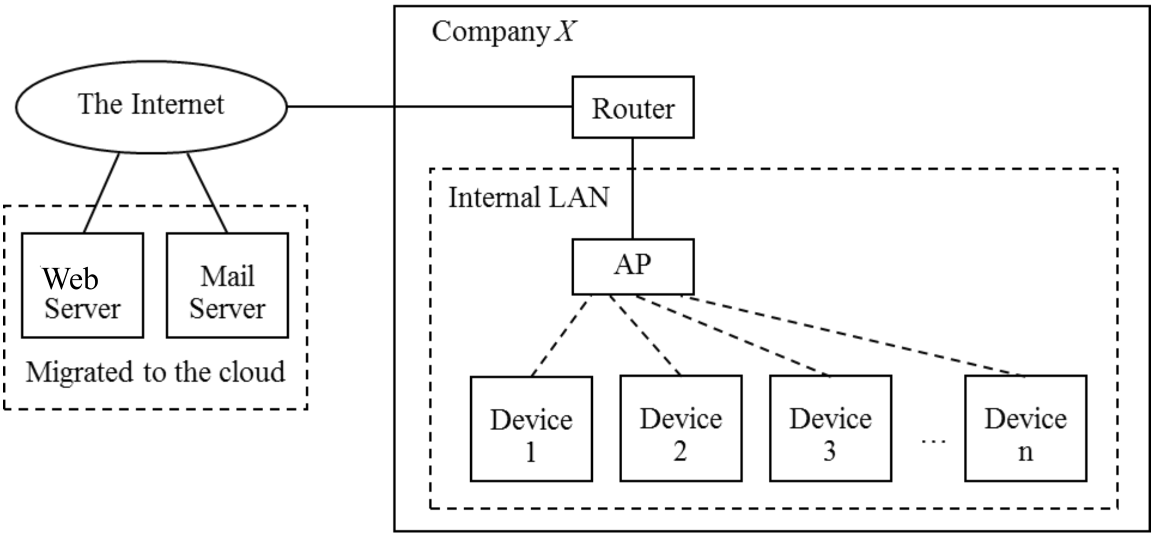


Figure 1. The current network configuration of Company *X*

Initially, [A] filtering was deployed as the primary security measure. It is an access control mechanism available to allow only pre-registered devices to connect to the network. Unlike an IP address where the user can manually change easily, [A] is embedded in the network interface card makes it more difficult to change without proper utilities.

Furthermore, Mr. *A* decided to hide the name of the network from the public. Normally, an AP sends beacon frames to provide information for other devices to join the network. This process is called beaconing.

An important information necessary to initiate the connection is $\boxed{\text{B}}$, which represents the network name. If the $\boxed{\text{B}}$ beaconing is turned off, the users are required to enter the network name manually and will be able to connect to the network only if it is matched with the one previously entered on the AP.

Since these mechanisms do not provide encryption, it is possible that the communication may be intercepted and interpreted easily by attackers. To maintain confidentiality, Mr. *A* decided to enable $\boxed{\text{C}}$ on the Web server to secure the HTTP connections between the Web server and the browsers. When used in conjunction with plain HTTP, the secure version of the protocol called HTTPS, which uses port 443 instead of port 80, is available.

Despite the implemented security measures, it turned out that unauthorized users have gained access to the network and use the network as a free Wi-Fi. Although the network name is not broadcasted by the AP and is invisible to the public, it is revealed whenever a legitimate user trying to associate with the AP. Figure 2 shows parts of the information captured from the wireless networks near Company *X*.

AP	Encryption	Network Name
`xx:xx:xx:C3:5F:AC`	`WPA`	
`xx:xx:xx:2C:46:80`	`Open`	
`xx:xx:xx:1B:83:60`	`Open`	`Public`
`xx:xx:xx:8E:1D:50`	`WEP`	`Warehouse`

AP	Client Station	Probe
`(not associated)`	`xx:xx:xx:B2:62:3C`	`CAFE`
`(not associated)`	`xx:xx:xx:1C:BB:79`	`WLAN`
`xx:xx:xx:C3:5F:AC`	`xx:xx:xx:E1:19:A1`	`Office`
`xx:xx:xx:2C:46:80`	`xx:xx:xx:F0:37:DB`	
`xx:xx:xx:2C:46:80`	`xx:xx:xx:A4:26:F0`	`WLAN`
`xx:xx:xx:8E:1D:50`	`xx:xx:xx:DD:8F:44`	

Figure 2. Parts of the information captured near Company *X*

As shown in Figure 2, there are 4 APs available near Company *X*. Two of them have hidden network names, and those names are revealed as “Office” and “WLAN” with the $\boxed{\text{A}}$ of `xx:xx:xx:C3:5F:AC` and `xx:xx:xx:2C:46:80` respectively.
Since the wireless network of Company *X* has no encryption enabled, it is obvious that the AP with the network name $\boxed{\text{D}}$ belongs to Company *X*.

Furthermore, from Figure 2, it is most likely that an unauthorized user will be able to connect to Company *X*'s network later on, by inappropriately changing his/her computer's [A] to [E], which is actually allowed to connect to the AP.

Mr. *A* realized that encryption should be enabled in order to prevent unauthorized access. Among the APs in Figure 2, the network name of the most secure AP is [F]. Mr. *A* is now studying the encryption method WPA2, that is better than the ones used in Figure 2, and he may choose it at a later time.

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank [ ] in the above description.

Answer group for A through C

a) DHCP
b) MAC address
c) Passphrase
d) SNMP
e) SSID
f) SSH
g) SSL

Answer group for D and F

a) CAFE
b) Office
c) Public
d) Warehouse
e) WLAN

Answer group for E

a) `xx:xx:xx:1B:83:60` or `xx:xx:xx:2C:46:80`
b) `xx:xx:xx:1C:BB:79`
c) `xx:xx:xx:2C:46:80`
d) `xx:xx:xx:A4:26:F0` or `xx:xx:xx:F0:37:DB`
e) `xx:xx:xx:C3:5F:AC`
f) `xx:xx:xx:E1:19:A1`

Q6. Read the following description of a merge sort algorithm and the flowcharts themselves, and then answer Subquestion.

[Program Description]

Conceptually, a merge sort works as follows:

(1) If the list is of length 0 or 1, then it is already sorted.
(2) Otherwise, divide the unsorted list into two sub-arrays of about half the size.
(3) Sort each sub-array recursively by re-applying merge sort.
(4) Merge the two sub-arrays back into one sorted list.

Merge sort incorporates two main ideas to improve its runtime:

(1) A small list will take fewer steps to sort than a large list.
(2) Fewer steps are required to construct a sorted list from two sorted lists than two unsorted lists. For example, it is enough to traverse each list only once if they're already sorted (see the merge function below for an example implementation).

Figure 1 shows an example of merge sort to sort a list of integers contained in an array. It shows how the elements [38, 27, 43, 3, 9, 82, 10] are sorted in ascending order to get the result [3, 9, 10, 27, 38, 43, 82] using merge sort algorithm.

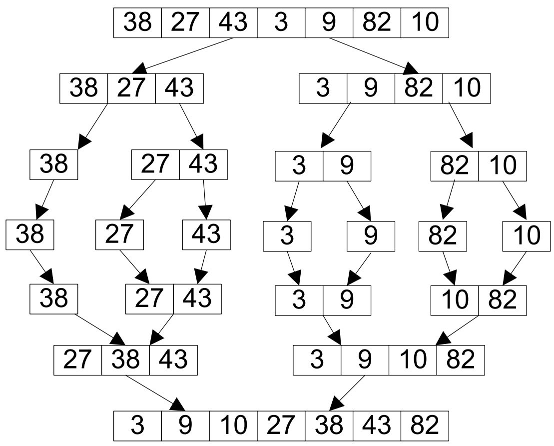


Figure 1. Merge sort example

Generally, when an array A with n elements (ranging from $A[1]$ to $A[n]$) is to be sorted, apply merge sort to 2 sub-arrays (A[l]: l = 1, 2, ..., c) and (A[r]: r = c+1, c+2, ..., n), where c is the integer part of $n \div 2$. When the 2 sub-arrays are returned, they will have been sorted. They can now be merged together to form a sorted array.

Figure 2 shows the flowchart of the function Merge_sort(m). The function sorts the data in array m in ascending order.

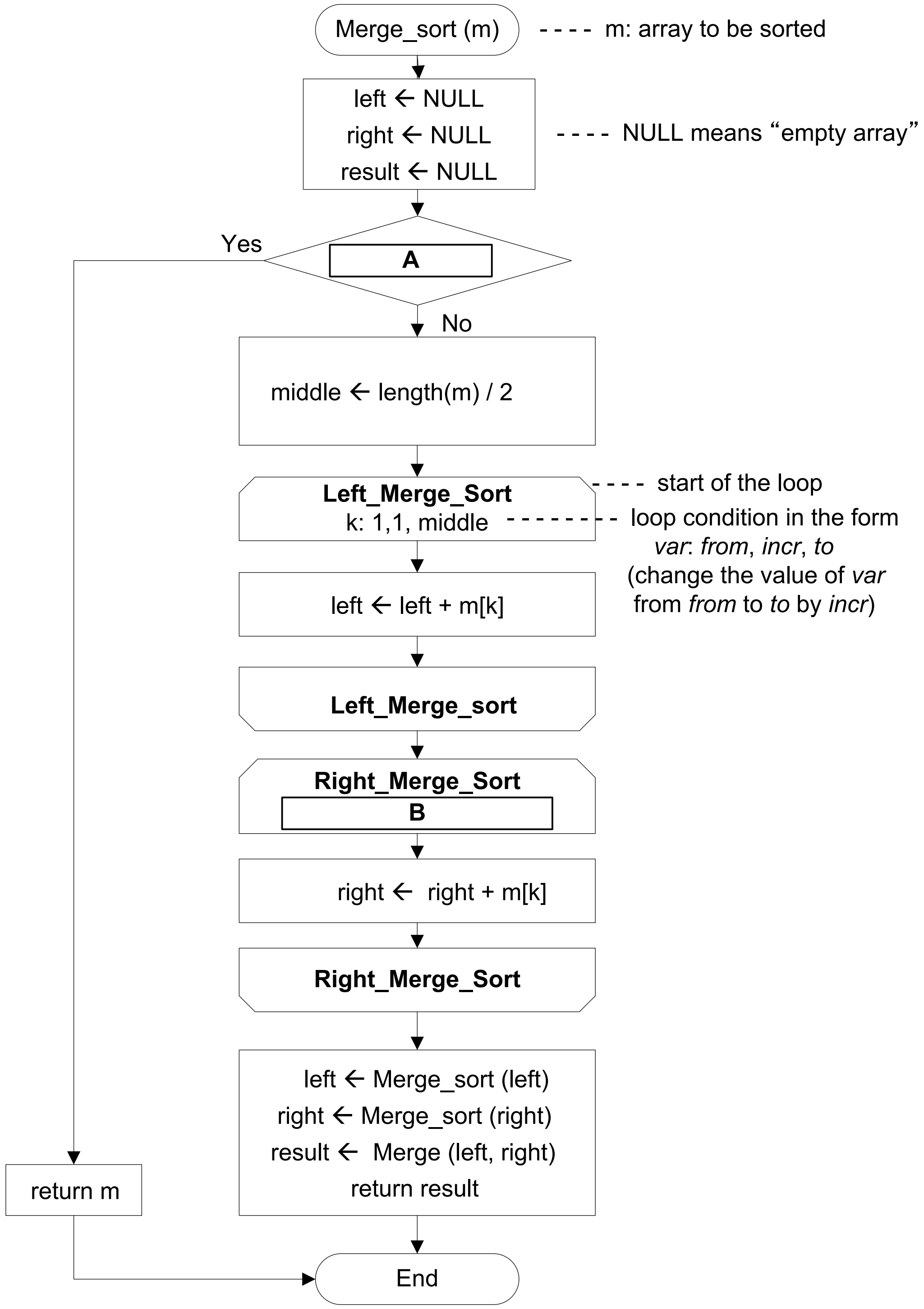
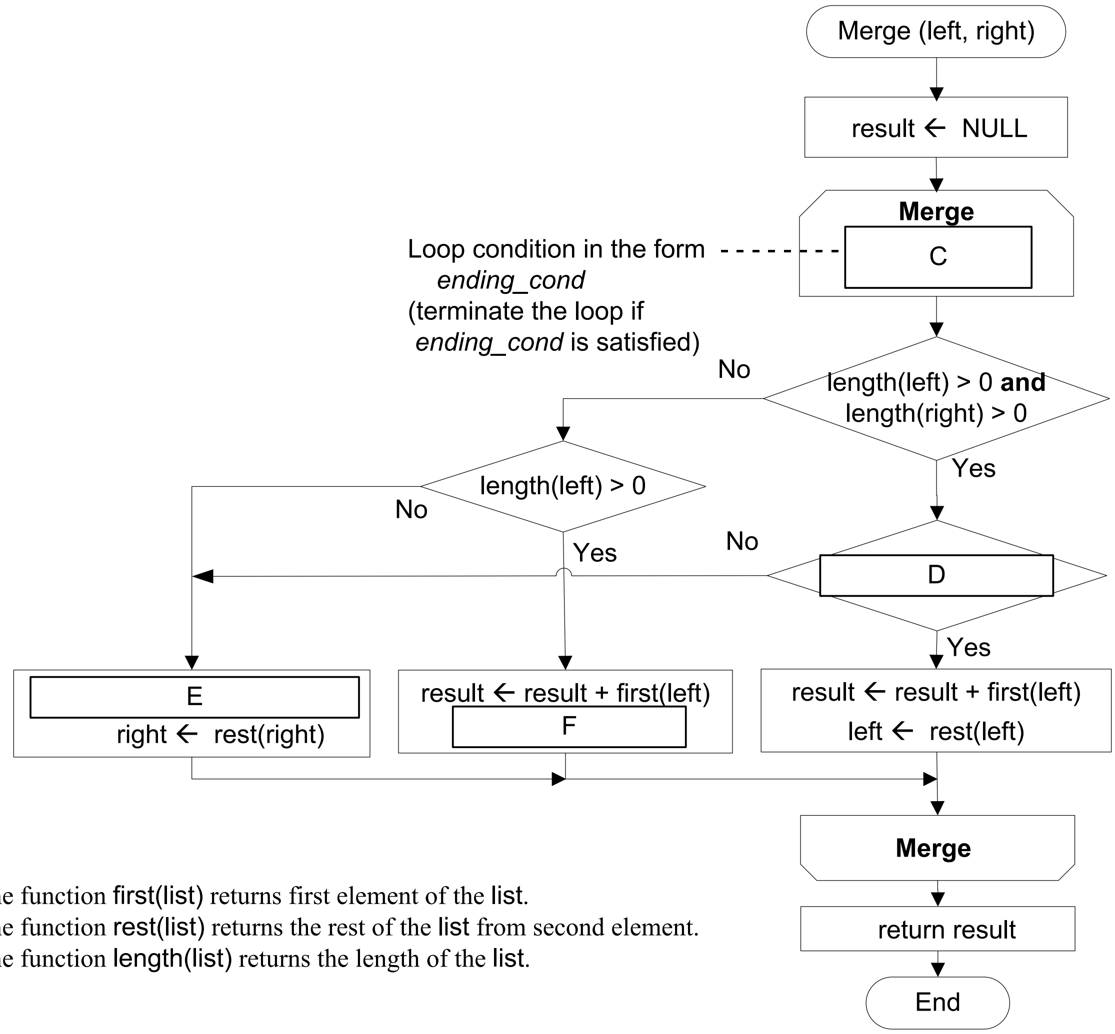


Figure 2. Function Merge_sort(m)

Explanation:

- All variables in Figure 2 are integer type.
- k is a temporary variable.
- left, right and result are temporary arrays to keep the elements during sorting process. They are initialized as empty arrays at the beginning.
- left ← left + m[k] denotes that array m[k] is appended to left array.
- right ← right + m[k] denotes that array m[k] is appended to right array.
- Function length(m) returns the number of elements of array m (returns 0 if m is NULL).

Figure 3 shows the flowchart of the function Merge(left, right) called from the function Merge_sort(m). This function merges 2 sub-arrays left and right.



Note:
(1) The function first(list) returns first element of the list.
(2) The function rest(list) returns the rest of the list from second element.
(3) The function length(list) returns the length of the list.

Figure 3. Function Merge(left, right)

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in Figure 2 and Figure 3.

Answer group for A

a) length(m) < 1
b) length(m) ≤ 1
c) length(m) = 1
d) length(m) > 1
e) length(m) ≥1

Answer group for B

a) k: middle+1, 1, length(m)
b) k: middle+1, -1, length(m)
c) k: middle, 1, length(m)
d) k: middle, -1, length(m)

Answer group for C

a) length(left) ≠ 0 and length(right) ≠ 0
b) length(left) = 0 and length(right) = 0
c) length(left) = 0 or length(right) = 0
d) length(left) > 0 and length(right) > 0
e) length(left) ≥ 0 and length(right) ≥ 0

Answer group for D

a) first(left) ≤ first(right)
b) first(left) = first(right)
c) first(left) = first(right) + 1
d) first(left) = first(right) - 1
e) first(left) > first(right)

Answer group for E and F

a) left ← rest (left)
b) left ← rest (right)
c) result ← result + first (left)
d) result ← result + first (right)
e) right ← rest (left)
f) right ← rest (right)

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark the Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestions 1 and 2.

[Program Description]

A word or a character sequence is called a "palindrome", if it remains the same when written backwards (right to left). For example, the word "MADAM" is a palindrome, because it is the same when it is written backwards.

A word is called a "mirrored word", if each character of the word has its mirror character and the word written backwards gives the original word. For example, the word "2TOTS" is a mirrored word, because "T" and "O" are mirror character of their own, and "2" and "S" are the mirror character of each other.

Some words are palindrome as well as mirrored. Such kind of words are called "mirrored palindrome". The word "XOYOX" is a mirrored palindrome, because it is a palindrome as well as a mirrored word. Here, "X", "O" and "Y" are all their own mirror character.

Table 1 shows the acceptable characters and their mirror characters. Acceptable characters consist of upper case alphabets, digits and space. '\0' (null) represents there is no mirror character for that character. Here, "0" (zero) and "O" (oh) will be considered as the same character, and hence only the character "O" is given in Table 1.

Table 1. Acceptable characters and their mirror characters

Character	Mirror Character
A	A
B	\0
C	\0
D	\0
E	3
F	\0
G	\0
H	H
I	I
J	L
K	\0
L	J

Character	Mirror Character
M	M
N	\0
O	O
P	\0
Q	\0
R	\0
S	2
T	T
U	U
V	V
W	W
X	X

Character	Mirror Character
Y	Y
Z	5
1	1
2	S
3	E
4	\0
5	Z
6	\0
7	\0
8	8
9	\0

Input character string contains words to check and each separated by exactly one space. The length of input character string should not exceed 100. The last character in input character string must not be a space.

The function `gets()` reads input character string, and adds '\0' after the last character.

For each word in the input character string, the program prints one of the following message according to the conditions.

Message	Condition
`XXX is an ordinary word.`	If `XXX` is neither a palindrome nor a mirrored word
`XXX is an ordinary palindrome.`	If `XXX` is a palindrome and is not a mirrored word
`XXX is a mirrored word.`	If `XXX` is not a palindrome and is a mirrored word
`XXX is a mirrored palindrome.`	If `XXX` is a palindrome as well as a mirrored word

Example of an input character string and its output messages are as follows:
(Input character string)

```
ITPEC MADAM 2TOTS XOYOX
```

(Output messages)

```
ITPEC is an ordinary word.
MADAM is an ordinary palindrome.
2TOTS is a mirrored word.
XOYOX is a mirrored palindrome.
```

The program has 2 user defined functions:

(1) `int check(char str[], int i, int j)`
This function checks the word, and returns 3 for mirrored palindrome, 2 for ordinary palindrome, 1 for mirrored word, and 0 for ordinary word. Here, `str[]` is the input character string, `i` is the start index, and `j` is the end index of the word.

(2) `int getEndIndex(char str[], int i)`
This function returns the end index of the word in the input character string `str[]` whose start index is `i`.

[Program]

```
#include <stdio.h>
#include <string.h>

char reverse[128];

int check(char str[], int i, int j) {
   int val = 3;

   for(; [    A    ]; i++, j--) {
      if (str[i] != str[j]) {
         [    B    ];
      }
      if (str[i] != reverse[str[j]]) {
         [    C    ];
      }
   }
   return val;
}

int getEndIndex(char str[], int i) {
   while ([    D    ]) {
      i++;
   }
   return i - 1;
}
```

```
void main() {
   int i, j, sIndex = 0, eIndex;
   char str[101], s[101];

   for (i = 0; i < 128; i++) {
      reverse[i] = '\0';
   }

   reverse['A'] = 'A'; reverse['E'] = '3'; reverse['H'] = 'H';
   reverse['I'] = 'I'; reverse['J'] = 'L'; reverse['L'] = 'J';
   reverse['M'] = 'M'; reverse['O'] = 'O'; reverse['S'] = '2';
   reverse['T'] = 'T'; reverse['U'] = 'U'; reverse['V'] = 'V';
   reverse['W'] = 'W'; reverse['X'] = 'X'; reverse['Y'] = 'Y';
   reverse['Z'] = '5'; reverse['1'] = '1'; reverse['2'] = 'S';
   reverse['3'] = 'E'; reverse['5'] = 'Z'; reverse['8'] = '8';

   gets(str);

   while (str[sIndex] != '\0') {
      while (str[sIndex] == ' ') {
         sIndex++;
      }
      eIndex = getEndIndex(str, sIndex);
      for (i = 0, j = sIndex; i <= eIndex - sIndex; i++, j++) {
         s[i] = str[j];
      }
      s[i] = '\0';

      switch (check(str, sIndex, eIndex)) {
         case 3:
            printf("%s is a mirrored palindrome.\n", s);
            break;
         case 2:
            printf("%s is an ordinary palindrome.\n", s);
            break;
         case 1:
            printf("%s is a mirrored word.\n", s);
            break;
         default:
            printf("%s is an ordinary word.\n", s);
      }
      [    E    ];
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the above program.

Answer group for A

a)	`i != j`	b)	`i < j`	c)	`i <= j`
d)	`i == j`	e)	`i > j`	f)	`i >= j`

Answer group for B and C

a)	`val &= 1`	b)	`val &= 2`	c)	`val += 1`
d)	`val += 2`	e)	`val |= 1`	f)	`val |= 2`

Answer group for D

a) `str[i] != ' ' && str[i] != '\0'`
b) `str[i] != ' ' || str[i] != '\0'`
c) `str[i] == ' ' && str[i] == '\0'`
d) `str[i] == ' ' || str[i] == '\0'`

Answer group for E

a)	`eIndex = sIndex + 1`	b)	`eIndex = sIndex - 1`
c)	`sIndex = eIndex + 1`	d)	`sIndex = eIndex - 1`

Subquestion 2

From the answer group below, select the correct answer to be inserted into the blank $\boxed{}$ in the following description.

The function `getEndIndex(char str[], int i)` will return the value $\boxed{}$, if the value of `str[]` is "`MADAM ADAM PALINDROME`", and the value of `i` is 6.

Answer group

a)	6	b)	7	c)	8
d)	9	e)	10		

Q8. Read the following description of Java programs and the programs themselves, and then answer Subquestions 1 and 2.

[Program Description]

The programs implement the operation of electronic device for measuring blood pressure.

(1) The device automatically measures systolic and diastolic blood pressure reading in mmHg, and displays the measurement results. Table 1 is used to determine the measurement results.

Table 1. Interpretation of blood pressure measurement results

Category of Blood Pressure	Systolic BP (mmHg)	Diastolic BP (mmHg)	Advice on Results	Color Indicator
Hypotension	< 100	< 60	Practice a healthy lifestyle. Consult your doctor only if suffering symptoms of low BP (e.g. fainting).	WHITE
Normal	< 140	< 90	Practice a healthy lifestyle.	GREEN
PreHypertension	140−159	90−99	Practice a healthy lifestyle. Re-measure BP monthly over next 3 months. If high levels (≥ 140/90) persist (e.g. 2 high readings on 2 separate occasions), consult doctor.	YELLOW
Moderate Hypertension	160−179	100−109	Practice a healthy lifestyle. Re-measure BP monthly over next 4 months. If high levels (≥ 160/100) persist (e.g. 2 high readings on 2 separate occasions), consult doctor.	ORANGE
Severe Hypertension	≥ 180	≥ 110	Re-measure BP in a few days. If BP ≥ 180/110, consult doctor.	RED

(2) If systolic BP and diastolic BP fall into different categories, the higher value should be taken for classification. For example, 160/92 should be classified as “Moderate Hypertension” and 180/92 should be classified as “Severe Hypertension”.

(3) The programs are composed of 5 classes; `BloodPressureTest`, `Tension`, `Pressure`, `Device`, and `InvalidPressuresException`.

[Program 1]

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class BloodPressureTest {
   static Device device;

   public static void main(String[] args) {
      device = new Device();    // Create a new device.
      Scanner input = new Scanner(System.in);
      System.out.print("Enter the High and Low blood pressures: ");
      try {
         int h = input.nextInt();
         int l = input.nextInt();
         if (l >= h) {
            throw new InvalidPressuresException
               ("The Low value must be less than the High value");
         } else {
            device.turnOn(l, h);    // a user launches the device
            System.out.println(device);
         }
      } catch (InvalidPressuresException e) {
         System.out.println(e.getMessage());
      } catch (InputMismatchException e) {
         System.out.println("You must input only integers");
      }
   }
}
```

[Program 2]

```
import java.util.EnumSet;
import java.util.HashMap;
import java.util.Map;

public enum Tension {
   HYPOTENSION, NORMAL, PREHYPERTENSION,
   MODERATE_HYPERTENSION, SEVERE_HYPERTENSION;
   protected static final Map<    A    >
      lookup = new HashMap<    A    >();
   static {
      int ordinal = 0;
      for (Tension suit : EnumSet.allOf(Tension.class)) {
         lookup.put(ordinal, suit);
         ordinal += 1;
      }
   }
```

```
    public static Tension fromOrdinal(int ordinal) {
       return lookup.get(ordinal);
    }
 }
```

[Program 3]

```
 public class Pressure {
    private static String newline = "\n";
    static String advices[] = {
       "Practice a healthy lifestyle." + newline
          + "Consult your doctor only" + newline
          + "if suffering symptoms of low BP (e.g. fainting).",

       "Practice a healthy lifestyle.",

       "Practice a healthy lifestyle." + newline
          + "Re-measure BP monthly over next 3 months." + newline
          + "If high levels (>=140/90) persist, consult doctor.",

       "Practice a healthy lifestyle." + newline
          + "Re-measure BP monthly over next 4 months." + newline
          + "If high levels (>=160/100) persist, consult doctor.",

       "Re-measure BP in a few days." + newline
          + "If BP>=180/110, consult doctor."
    };

    private [     B     ] status;
    private int sbp;
    private int dbp;

    public Pressure() {
       this.sbp = 0;
       this.dbp = 0;
    }

    public Pressure(int low, int high) {
       this.sbp = high;
       this.dbp = low;
    }

    public void setStatus([     B     ] state) {
       this.status = state;
    }

    public [     B     ] getStatus() {
       return this.status;
    }
```

```
    public int getHValue() {
        return this.sbp;
    }

    public int getLValue() {
        return this.dbp;
    }

    public String toString(int i) {
        return advices[i];
    }
}
```

[Program 4]

```
public class Device {
    enum Indicators {
        WHITE,
        GREEN,
        YELLOW,
        ORANGE,
        RED
    };

    private static final int high[] = {100, 140, 160, 180};
    private static final int low[] = {60, 90, 100, 110};
    static String newline = "\n";

    Pressure pressures;
    Indicators indication;
    private int sbp;    //pressure's high value
    private int dbp;    //pressure's low value

    public void turnOn(int l, int h) {
        pressures = new Pressure(l, h);
        pressures.setStatus(diagnose());
    }

    private [     B     ] diagnose() {
        Tension state = null;
        sbp = pressures.getHValue();
        dbp = pressures.getLValue();
        int i = 0;
        while((i < high.length) && ([     C     ])){
            i++;
        }
        state = Tension.fromOrdinal(i);
        return state;
    }
```

```
      private Indicators getIndicator() {
         switch (pressures.getStatus().ordinal()) {
            case 0:
               indication = Indicators.WHITE;
               break;
            case 1:
               indication = Indicators.GREEN;
               break;
            case 2:
               indication = Indicators.YELLOW;
               break;
            case 3:
               indication = Indicators.ORANGE;
               break;
            case 4:
               indication = Indicators.RED;
               break;
         }
         return indication;
      }

      public String toString() {
         return "BLOOD PRESSURE MONITOR" + newline
            + "------------------------" + newline + "SBP: "
            + pressures.getHValue() + " | " + "DBP: "
            + pressures.getLValue() + newline + "Your blood pressure is "
            + pressures.getStatus() + newline
            + pressures.toString(pressures.getStatus().ordinal())
            + newline + "------------------------" + newline;
      }
   }
```

[Program 5]

```
   class InvalidPressuresException extends Exception {
      public InvalidPressuresException (String msg) {
         super(msg);
      }
   }
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{\qquad}$ in the above programs.

Answer group for A

a) `Integer, Pressure` b) `Integer, Tension`
c) `int, Pressure` d) `int, Tension`

Answer group for B

a) `Pressure` b) `Tension`
c) `Tension.HYPOTENSION` d) `enum Tension`

Answer group for C

a) `sbp < high[i] || dbp < low[i]` b) `sbp <= high[i] || dbp <= low[i]`
c) `sbp > high[i] || dbp > low[i]` d) `sbp >= high[i] || dbp >= low[i]`

Subquestion 2

From the answer group below, select the correct answer to be inserted into each blank $\boxed{\qquad}$ in Table 2.

In the program testing phase, various test cases are prepared and tested. Each test case has a pair of high (systolic) and low (diastolic) BP values.

Table 2 shows the test cases (1), (2) and (3), with their test data and output results.

Table 2. Test cases and output results

Test case	Test data		Output result
	High	Low	
Test case (1)	88.0	138	$\boxed{\text{D}}$
Test case (2)	138	90	$\boxed{\text{E}}$
Test case (3)	160	98	$\boxed{\text{F}}$

Answer group (Note: shaded parts   are not shown.)

a)
```
BLOOD PRESSURE MONITOR
------------------------
SBP:      | DBP:     
Your blood pressure is MODERATE_HYPERTENSION
Practice a healthy lifestyle.
Re-measure BP monthly over next 4 months.
If high levels (>=160/100) persist, consult doctor.
------------------------
```

b)
```
BLOOD PRESSURE MONITOR
------------------------
SBP:      | DBP:     
Your blood pressure is NORMAL
Practice a healthy lifestyle
------------------------
```

c)
```
BLOOD PRESSURE MONITOR
------------------------
SBP:      | DBP:     
Your blood pressure is PREHYPERTENSION
Practice a healthy lifestyle.
Re-measure BP monthly over next 3 months.
If high levels (>=140/90) persist, consult doctor.
------------------------
```

d) `The Low value must be less than the High value`

e) `You must input only integers`

FE Afternoon Exam -- October 2014

Question No	Subquestion No		Correct Answer
1	1	A	b
		B	a
		C	b
		D	d
	2	E	b
2		A	d
		B	b
		C	h
		D	d
		E	a
3	1	A	c
		B	d
	2	C	c
		D	b
		E	d
4		A	f
		B	c
		C	b
		D	d
		E	e
		F	f
5		A	b
		B	e
		C	g
		D	e
		E	d
		F	b
6		A	b
		B	a
		C	b
		D	a
		E	d
		F	a
7	1	A	c
		B	a
		C	b
		D	a
		E	c
	2		d
8	1	A	b
		B	b
		C	d
	2	D	e
		E	c
		F	a

[Note] Answers for C and D can be exchanged. (Question 1)

[Note] Answers for A and B can be exchanged. (Question 3)

[Note] Answers for D and E can be exchanged. (Question 3)

ITEE