October / November 2021

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1	Q2 – Q5	Q6	Q7, Q8
Question Selection	Compulsory	Select 2 of 4	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 questions **Q2** through **Q5**, and **Q7** and **Q8**, mark the ⓢ of the questions you select to answer in the "Selection Column" on your answer sheet.

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

   [Sample Question]
   Which of the following should be used for marking your answer on the answer sheet?

   Answer group
   a) Ballpoint pen   b) Crayon   c) Fountain pen   d) Pencil

   Since the correct answer is " d) Pencil ", mark the answer as below:

   [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 in the pseudo-language

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

[Declaration, comment, and process]

Notation		Description
type: *var1*, ..., *array1*[], ...		Declares variables *var1*, … , and/or arrays *array1*[], … , by data *type* such as INT and CHAR.
FUNCTION: *function*(*type*: *arg1*, ...)		Declares a *function* and its arguments *arg1*, … .
/* comment */		Describes a comment.
Process	*variable* ← *expression*;	Assigns the value of the *expression* to the *variable*.
	function (*arg1*, ...);	Calls the *function* by passing / receiving the arguments *arg1*, … .

	IF (*condition*) {
	    process1
	}
	ELSE {
	    process2
	}
		Indicates the selection process.
		If the *condition* is true, then *process1* is executed.
		If the *condition* is false, then *process2* is executed, when the optional ELSE clause is present.

	WHILE (*condition*) {
	    process
	}
		Indicates the “WHILE” iteration process.
		While the *condition* is true, the *process* is executed repeatedly.

	DO {
	    process
	} WHILE (*condition*);
		Indicates the “DO-WHILE” iteration process.
		The *process* is executed once, and then while the *condition* is true, the *process* is executed repeatedly.

	FOR (*init*; *condition*; *incr*) {
	    process
	}
		Indicates the “FOR” iteration process.
		While the *condition* is true, the *process* is executed repeatedly.
		At the start of the first iteration, the process *init* is executed before testing the *condition*.
		At the end of each iteration, the process *incr* is executed before testing the *condition*.

[Logical constants]

true, false

[Operators and their precedence]

Type of operation	Unary	Arithmetic		Relational	Logical	
Operators	+, −, not	×, ÷, %	+, −	>, <, ≥, ≤, =, ≠	and	or
Precedence	High ←——→ Low					

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

Question **Q1** is **compulsory**.

Q1. Read the following description of public key cryptography, and then answer Subquestions 1 and 2.

Public key cryptography is a widely used method to secure e-mail communications. It makes use of a pair of keys to encrypt and sign the e-mail messages. The same pair of keys can then be used in reverse to decrypt and verify the e-mail messages. Public key cryptography uses two keys: a private key that is kept secure, and a public key that can be published publicly. Figure 1 shows a typical application of public key cryptography, message encryption and digital signature.

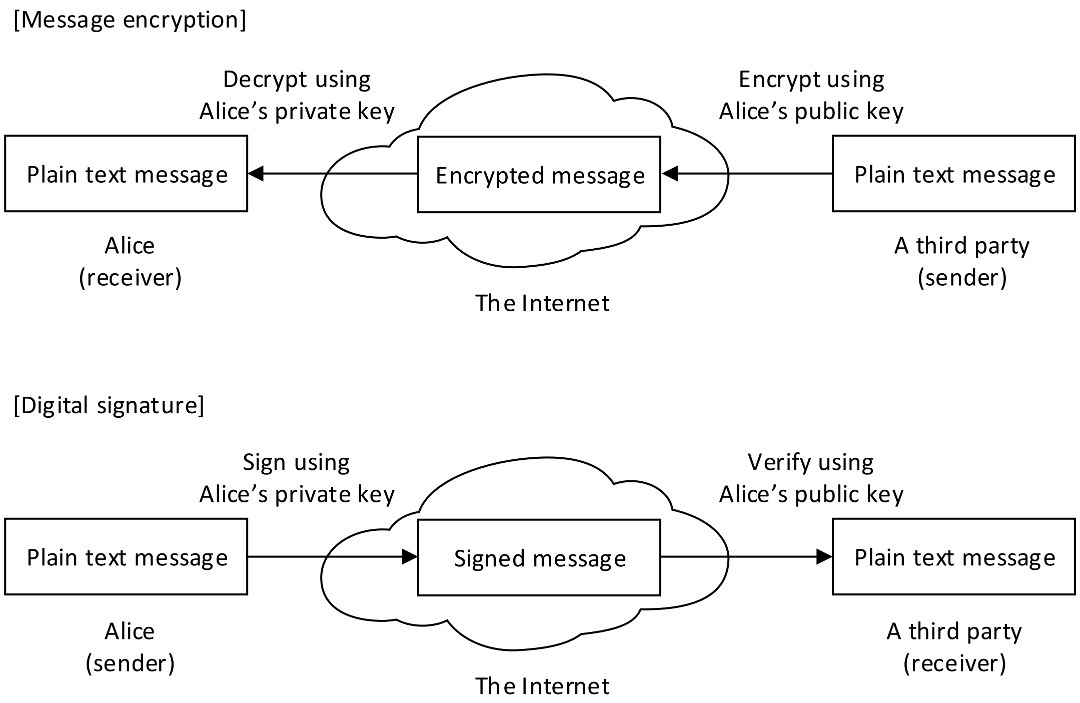


Figure 1 Typical application of public key cryptography

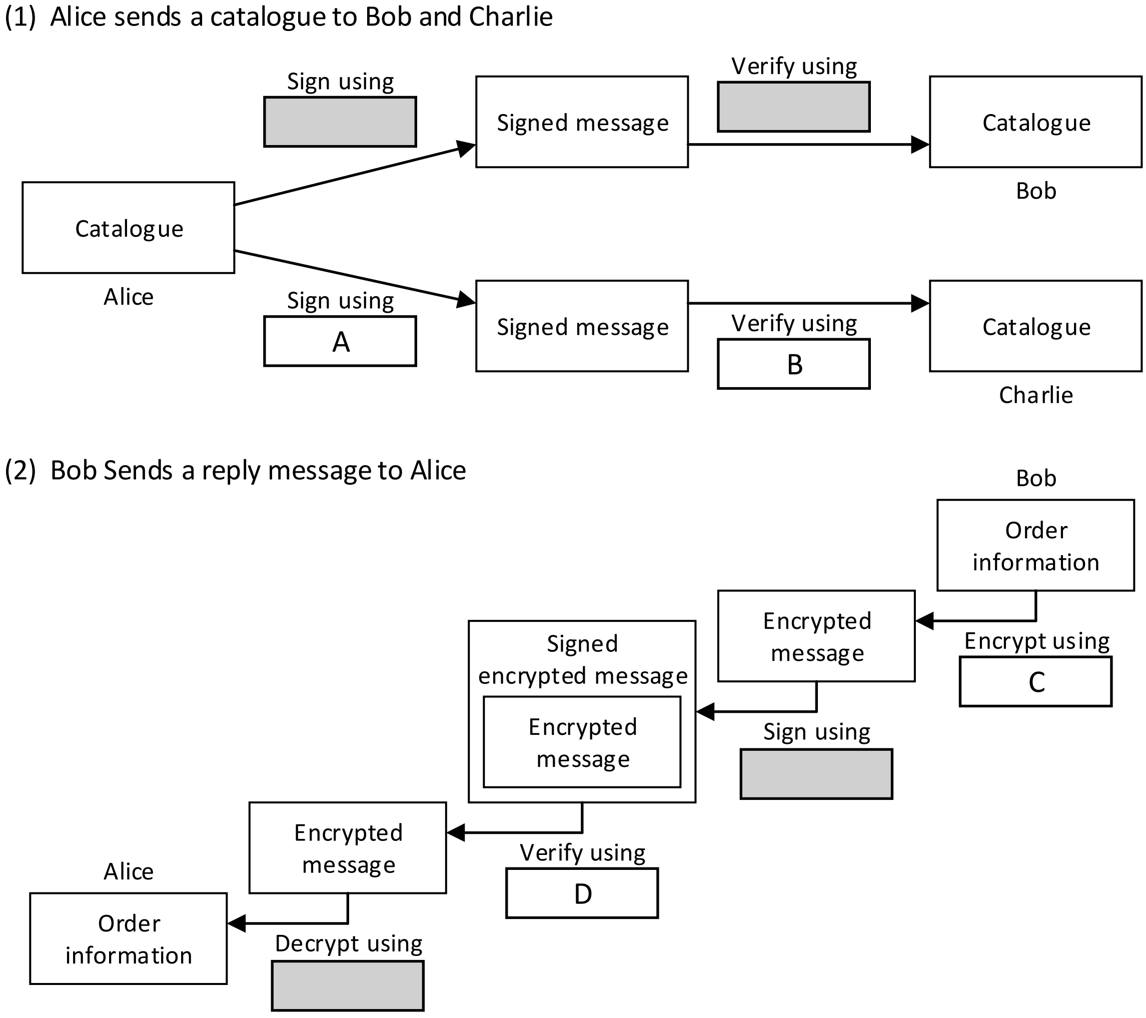
In the scenario shown in Figure 1, Alice has published her public key along with her e-mail address on the Internet. When a third party wishes to communicate with Alice, it can use Alice's public key to encrypt an e-mail message and send it to Alice who can then use her private key to decrypt the message and obtain the original message. Conversely, Alice can use her private key to sign a message before sending it to a third party. The third party can then use Alice's public key to verify the signed message and confirm the authenticity of the original message.

The signing process ensures integrity rather than confidentiality because Alice's public key is publicly available allowing anyone who can access Alice's public key to verify the signed message.

Figure 2 shows the communication flow between Alice, Bob and Charlie using public key cryptography.

(1) Alice sends a product catalogue to Bob and Charlie. The information on the catalogue is not confidential; thus the e-mail messages are not encrypted. However, the messages are signed to ensure integrity and non-repudiation of the information on the catalogue.

(2) After receiving the catalogue, Bob wishes to make a purchase; hence, he sends a reply message to Alice. As the message includes sensitive financial information, the message is encrypted to make sure that only Alice can read the message. The message is also signed to allow Alice to confirm that the purchase is actually made by Bob.



Note: Shaded parts are not shown

Figure 2 Communication flow using public key cryptography

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank $\boxed{\qquad}$ in Figure 2. If needed, the same answer can be selected more than once.

Answer group for A through D

a)	Alice’s private key	b)	Alice’s public key
c)	Bob’s private key	d)	Bob’s public key
e)	Charlie’s private key	f)	Charlie’s public key

Subquestion 2

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

In addition to the use of the correct keys for the intended functions, the order of processing is important. Some e-mail programs allow users to choose whether to encrypt the message first and then sign the encrypted message later or the other way around. Figure 3 shows the encrypt-then-sign method and sign-then-encrypt method.

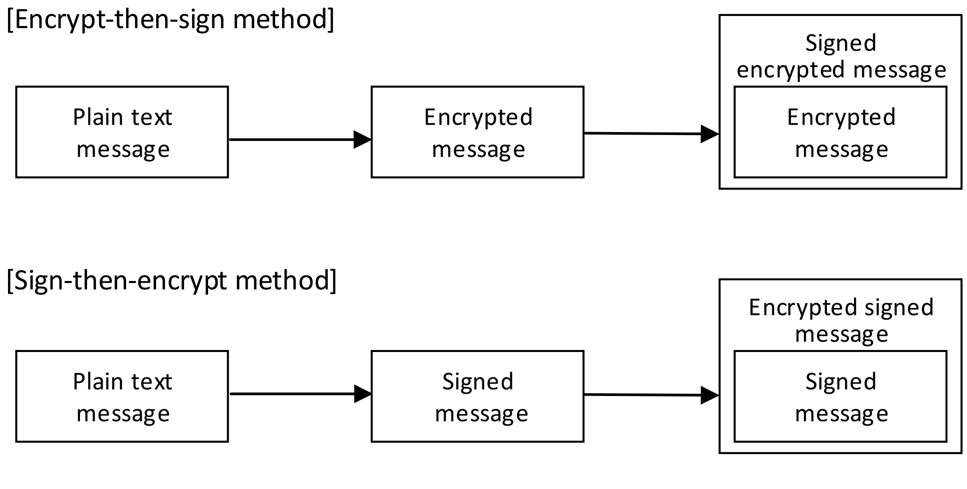


Figure 3 Encrypt-then-sign method and sign-then-encrypt method

The order of the signing and encryption processes in each method provides a unique benefit over the other method. For instance, it is possible to $\boxed{\quad\text{E}\quad}$ by encrypting the message first and then signing the message later. Conversely, if the original message is signed first, it can help to $\boxed{\quad\text{F}\quad}$. Thus, it is important for users to choose the appropriate method accordingly.

Answer group for E and F

a) detect and discard unauthenticated message earlier

b) ensure that the private key can be recovered if the original key is lost

c) hide the signatory of the private key from unintended parties

d) increase the encryption strength

e) significantly reduce the time required to sign the message

Concerning questions **Q2** through **Q5**, **select two** of the four questions. For each selected question, mark the Ⓢ in the selection area on the answer sheet, and answer the question. If three or more questions are selected, only the first two questions will be graded.

Q2. Read the following description of the Gray code, and then answer Subquestions 1 through 3.

In this question, the symbols “ • ”, “+”, “⊕”, and “¯” indicate the logical operators AND, OR, exclusive OR, and NOT respectively.

The Gray code is a code in which the Hamming distance between any two adjacent codes is 1. The Hamming distance is the number of positions at which the corresponding bits differ when comparing binary numbers with the same number of bits. For example, the Hamming distance between two binary codes 1001 and 1010 is $\boxed{\text{A}}$. According to Table 1, two adjacent binary codes, whose decimal values are $\boxed{\text{B}}$, have the largest Hamming distance, and the value is $\boxed{\text{C}}$.

For decimal values 0, 1, 2, and 3, the binary codes are 0000, 0001, 0010, and 0011, and the Gray codes are 0000, 0001, 0011, and 0010 respectively as shown in Table 1.

Table 1 Table of 4-bit binary code and 4-bit Gray code

Decimal value	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

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 and C

a) 1 b) 2 c) 3 d) 4

Answer group for B

a) 1 and 2 b) 3 and 4 c) 5 and 6

d) 7 and 8 e) 9 and 10 f) 11 and 12

Subquestion 2

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

Consider the logical formulas that convert binary codes to equivalent Gray codes.
Table 2 shows the truth table for the binary code to the Gray code conversion. The inputs are the 4-bit binary codes (bits are named B1, B2, B3, and B4), and the outputs are the 4-bit Gray codes (bits are named G1, G2, G3, and G4).

Table 2 Truth table for binary code to Gray code conversion

Inputs				Outputs			
B1	B2	B3	B4	G1	G2	G3	G4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

The logical formulas that output G1, G2, G3, and G4 are expressed as follows:

$G1 = B1$

$G2 = B1 \oplus B2$

$G3 = B2 \oplus B3$

$G4 = \boxed{\quad D \quad}$

Answer group for D

a) $B0 \oplus (B3 + B4)$	b) $B0 \oplus (B3 \oplus B4)$
c) $B3 + B4$	d) $B3 \oplus B4$

Subquestion 3

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

Consider the logical formulas that convert Gray codes to equivalent binary codes.
Table 3 shows the truth table for the Gray code to the binary code conversion. The inputs are the 4-bit Gray codes (bits are named G1, G2, G3, and G4), and the outputs are the 4-bit binary codes (bits are named B1, B2, B3, and B4).

Table 3 Truth table for Gray code to binary code conversion

Inputs				Outputs			
G1	G2	G3	G4	B1	B2	B3	B4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

The logical formulas that output B1, B2, and B3 (B4 is omitted) are expressed as follows:

$B1 = G1$

$B2 = \overline{G1} \bullet G2 + G1 \bullet \overline{G2} = G1 \oplus G2$

$B3 = \overline{G1} \bullet \overline{G2} \bullet G3 + \overline{G1} \bullet G2 \bullet \overline{G3} + G1 \bullet \overline{G2} \bullet \overline{G3} + G1 \bullet G2 \bullet G3 = \boxed{\text{E}}$

Answer group for E

a) $(G1+\overline{G2}) \bullet G3 + (\overline{G1}+G2) \bullet G3 + G1 \bullet G2 \bullet G3$

b) $(G1+\overline{G2}) \bullet G3 + (\overline{G1}+G2) \bullet \overline{G3} + G1 \bullet G2 \bullet G3$

c) $(G1 \oplus G2) \bullet G3$

d) $(G1 \oplus G2) + G3$

e) $G1 \oplus G2 \oplus G3$

Concerning questions **Q2** through **Q5**, **select two** of the four questions.

Q3. Read the following description of a database for a DVD rental shop, and then answer Subquestions 1 through 3.

Store U, a DVD rental shop, uses a relational database to manage lending and returning operations at the store.

Disk table

DiskID	FilmName	DirectorName	YearOfProduction	DateEntered
D0280	Alice in AI wonderland	A. I. Carroll	2019	2019-10-25
D0480	Super IT man	Clark Kent Jr.	2020	2021-02-26
D0860	Hello database world!	I. T. Pec	2021	2021-10-10

Disk table contains information about each DVD, including the disk ID, film name, director name, year of production, and date entered in the store.

Customer table

CustomerID	CustomerName	Address	Phone
C001	John	401 North street, East city	1234567
C003	Kathy	44 South street, East city	2345678
C005	Tom	109 Central street, West city	3456789

Customer table contains information about each customer, including the customer ID, customer name, address, and phone number.

Bill table

BillID	CustomerID	RentDateTime	RentalCharge	ReturnDateTime	Penalty
B1680	C005	2021-10-16 18:30	1.00	2021-10-18 17:50	NULL
B2120	C003	2021-10-21 12:10	2.00	NULL	NULL
B2180	C005	2021-10-21 18:40	1.00	NULL	NULL

Bill table contains information about the bills for each customer, including the bill ID, customer ID, rented date and time, returned date and time, rental charge, and penalty.

BillDetail table

BillDetailID	BillID	DiskID
B1680-1	B1680	D0280
B2120-1	B2120	D0860
B2120-2	B2120	D0280
B2180-1	B2180	D0480

BillDetail table contains the details of a bill, including the bill ID and disk ID.

When a customer rents DVDs, a bill record is created. Rental charge is $1.00 per DVD. At that time, NULL is set to the columns ReturnDateTime and Penalty.
When the customer returns the rented DVDs, the value of ReturnDateTime is set. The value of Penalty is set only if the customer misses a deadline and he/she needs to pay a penalty. The deadline is 72 hours from the rented date and time. The penalty is $1.00 per day (24 hours) per DVD.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in the following SQL statement SQL1.

The following SQL statement SQL1 outputs the customers who currently (at the time SQL1 is executed) miss the deadline with the number of DVDs they still have with them.

```
-- SQL1 --
SELECT C.CustomerID, CustomerName, COUNT(*) AS DiskCount
FROM Customer C, Bill B, BillDetail D
WHERE C.CustomerID = B.CustomerID
AND B.BillID = D.BillID
AND       A
AND       B
GROUP BY C.CustomerID, CustomerName
```

From the sample data shown in the table definitions, SQL1 outputs the following result. It is assumed that SQL1 is executed at 13:30 on 2021-10-24.

CustomerID	CustomerName	DiskCount
C003	Kathy	2

Note:

- The GETDATE() function returns the current date and time, in a `'YYYY-MM-DD HH:MI:SS.sss'` format.
- The DATEDIFF(*part*, *start*, *end*) function returns the difference between two date and time values *start* and *end*, based on a date/time part, indicated by *part* (*YY*: *year*, *MM*: *month*, *DD*: *day*, *HH*: *hour*, *MI*: *minute*, *SS*: *second*). For example, `DATEDIFF(HH, '2021-10-23 23:00', '2021-10-24 01:20')` returns `3` (rounded up to the next integer).

Answer group for A

a) `Penalty IS NOT NULL` b) `Penalty IS NULL`

c) `ReturnDateTime IS NOT NULL` d) `ReturnDateTime IS NULL`

Answer group for B

a) `DATEDIFF(HH, RentDateTime, GETDATE()) < 72`

b) `DATEDIFF(HH, RentDateTime, GETDATE()) > 72`

c) `DATEDIFF(HH, ReturnDateTime, GETDATE()) < 72`

d) `DATEDIFF(HH, ReturnDateTime, GETDATE()) > 72`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank □ in the following SQL statement SQL2.

The following SQL statement SQL2 outputs all customers with their bill information. If a customer has no associated bill information, NULL is set to the fields BillID and RentDateTime.

```
-- SQL2 --
SELECT C.CustomerID, CustomerName, BillID, RentDateTime
FROM CUSTOMER C
      C       Bill B
ON C.CustomerID = B.CustomerID
```

From the sample data shown in the table definitions, SQL2 outputs the following result:

CustomerID	CustomerName	BillID	RentDateTime
C001	John	NULL	NULL
C003	Kathy	B2120	2021-10-21 12:10
C005	Tom	B1680	2021-10-16 18:30
C005	Tom	B2180	2021-10-21 18:40

Note:

- INNER JOIN: The keyword selects records that have matching values in both tables.
- FULL OUTER JOIN: The keyword returns all records when there is a match in either left or right table records.
- LEFT OUTER JOIN: The keyword returns all records from the left table, and the matched records from the right table. The result is NULL from the right side, if there is no match.

- RIGHT OUTER JOIN: The keyword returns all records from the right table, and the matched records from the left table. The result is NULL from the left side, if there is no match.

Answer group for C

a) `FULL OUTER JOIN`

b) `INNER JOIN`

c) `LEFT OUTER JOIN`

d) `RIGHT OUTER JOIN`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank □ in the following SQL statement SQL3.

The store manager wants to determine which DVDs were entered in the store a certain time ago and still appeal to customers. The following SQL statement SQL3 outputs DVDs that were entered in the store on and before 2019-10-31 and were rented twice or more during the last 12 months. It is assumed that Bill table contains the bill information for the last 12 months.

```
-- SQL3 --
SELECT D.DiskID, FilmName, COUNT(*) AS RentCount
FROM Disk D, BillDetail B
WHERE D.DiskID = B.DiskID
AND DateEntered <= '2019-10-31'
      D
      E
```

From the sample data shown in the table definitions, SQL3 outputs the following result:

DiskID	FilmName	RentCount
D0280	Alice in AI wonderland	2

Answer group for D and E

a) `AND COUNT(*) >= 2`

b) `HAVING COUNT(*) >= 2`

c) `GROUP BY D.DiskID`

d) `GROUP BY D.DiskID, FilmName`

e) `ORDER BY D.DiskID`

f) `ORDER BY D.DiskID, FilmName`

Concerning questions **Q2** through **Q5**, **select two** of the four questions.

Q4. Read the following description of an internal network of a company, and then answer Subquestions 1 through 3.

Company V designed the internal network infrastructure with the different network addresses 172.16.0.0/24, 172.16.1.0/24, and 172.16.2.0/24. These networks are connected to each other using Firewall, Router A, and Router B. Company V uses the NAT (Network Address Translation) function on Firewall to access the Internet from the internal network. A mail server, a DNS server, and an external Web server are located in the DMZ. PCs used by the tech team are connected to Network A and PCs used by the solution team are connected to Network B. Firewall blocks all incoming communication from the Internet to the internal network including the DMZ, except for explicitly allowed communication. The DMZ uses the subnet 192.168.1.0/27.

Figure 1 shows the internal network of Company V.

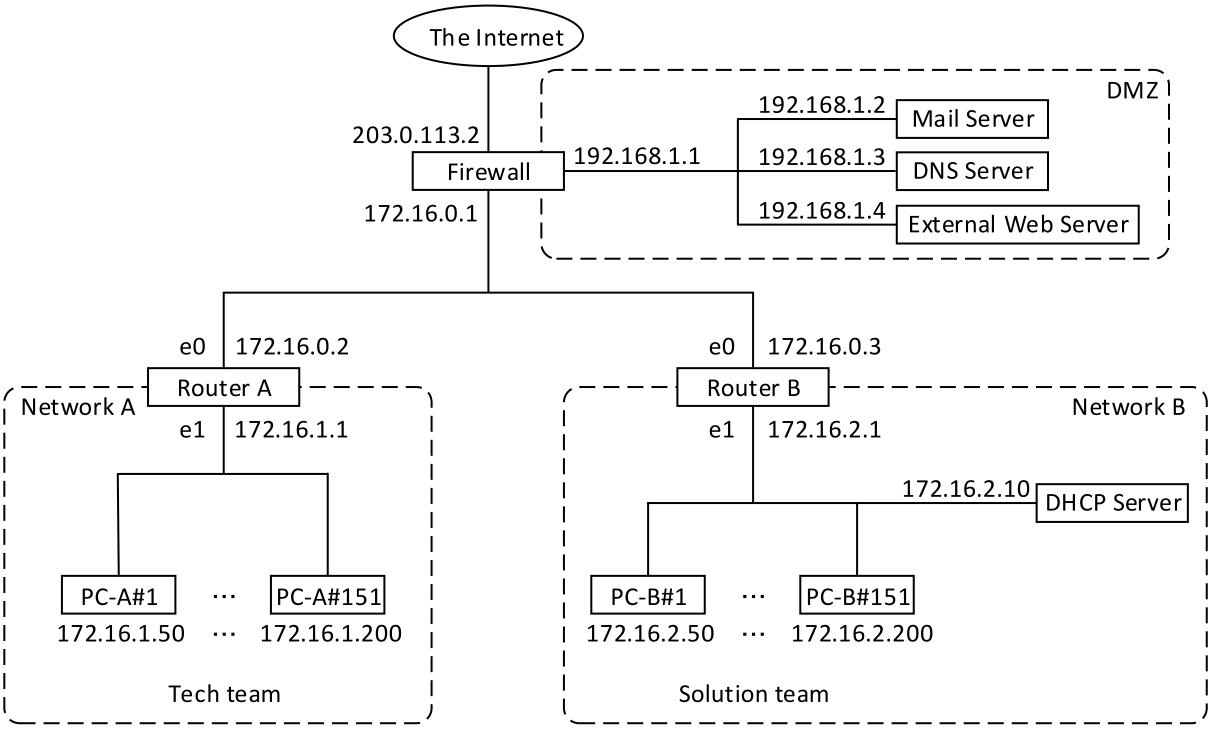


Figure 1 Internal network of Company V

Subquestion 1

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

To assign IP addresses to all PCs efficiently, DHCP server is running on Network B. All PCs will obtain IP address information from the DHCP server. For management devices such as routers and servers, a static IP address scheme is used. When PCs in Network A request DHCP information, they will send DHCP requests to the DHCP server on Network B. The DHCP server can provide IP addresses only when a DHCP request broadcast is received, and this is normally limited to those within the same physical network $\boxed{\quad\text{A}\quad}$. To get the DHCP information from Network A, a DHCP relay agent feature is needed on Router A. The DHCP relay agent is a service that forwards DHCP packets between clients and servers. It forwards received requests from clients to the DHCP server in a unicast fashion. In this network configuration, the destination address of a forwarded request is $\boxed{\quad\text{B}\quad}$. The DHCP relay agent also forwards replies from the DHCP server to clients.
In this network installation, a network administrator enables the DHCP relay agent on interface e1 on Router A which is connected to the DHCP clients. After the DHCP relay agent is configured, the DHCP request broadcast is forwarded to the DHCP server and it can reply with the valid IP address to the client on a different network.

Answer group for A

a) 172.16.0.0/16 b) 172.16.0.0/24 c) 172.16.1.0/24
d) 172.16.2.0/24 e) 172.16.3.0/24

Answer group for B

a) 172.16.0.2 b) 172.16.0.3 c) 172.16.0.255 d) 172.16.1.1
e) 172.16.1.255 f) 172.16.2.1 g) 172.16.2.10 h) 172.16.2.255

Subquestion 2

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

The routing tables are properly configured in the internal network of Company V. The network administrator confirms that PCs in Network A can communicate with each other and also communicate with servers in the DMZ. However, PCs in Network B cannot connect to other networks.

He confirms that the communication among PCs in Network B is working properly. However, he finds out that a ping message to Firewall and Router A fails. He checks the IP address information of several PCs in Network B. Table 1 shows the IP address information of host PC-B#1.

Table 1 IP address information of the host PC-B#1

IP address	Subnet mask	Default gateway
172.16.2.50	255.255.255.0	172.16.1.1

Based on the information in Table 1, he finally identifies that C. After he changes the configuration of the DHCP server as shown in Table 2, PCs in Network B can access servers in the DMZ and Network B.

Table 2 Changed configuration of DHCP server

Network	Subnet mask	Range	Default gateway
172.16.2.0	255.255.255.0	172.16.2.50 to 172.16.2.200	D

Answer group for C

a) the host could not contact the DHCP server and generated its own IP address

b) the host received the incorrect gateway address that was assigned by the DHCP server

c) the host uses a private address that is most likely assigned using DHCP

d) the host uses the IP address of DHCP server as its default gateway

Answer group for D

a) 172.16.0.2 b) 172.16.1.1 c) 172.16.2.1

d) 172.16.2.10 e) 192.168.1.1

Subquestion 3

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

To secure the servers located in the DMZ, the network administrator implements the traffic rules shown in Table 3 on Firewall based on the following rules:

(1) Mail transferring services (SMTP) on the external IP address of Firewall will be mapped to the mail server in the DMZ.

(2) Mail receiving services (IMAP) on the external IP address of Firewall will be mapped to the mail server in the DMZ.

(3) Allow access from the DMZ to the Internet via NAT (IP address translation), which is necessary for correct functionality of the mapped service.

(4) Allow access from the LANs to the DMZ, which makes the servers accessible to local users.

(5) Disable access from the DMZ to the LAN to protect against network intrusions from the DMZ.

Note that the return traffic of packets allowed by the aforementioned rules is also implicitly allowed.

Table 3 Traffic rules on Firewall (other settings are not shown)

Source	Destination	Service	IP version	Action	Translation
Internet	Firewall	E	any	Allow	MAP 192.168.1.2
Internet	Firewall	IMAP	any	Allow	MAP 192.168.1.2
DMZ	Internet	any	any	Allow	F
LANs	DMZ	any	any	Allow	
DMZ	LANs	any	any	deny	
…	…	…	…	…	…

Answer group for E and F

a) any	b) DNAT	c) DNS
d) HTTP	e) HTTPS	f) NAT
g) Port Forwarding	h) SMTP	i) udp

Concerning questions **Q2** through **Q5**, **select two** of the four questions.

Q5. Read the following description of a food ordering application, and then answer Subquestions 1 and 2.

Company W plans to offer takeout food services to its customers and develops a food ordering application. Using this application, a customer can select a restaurant, select foods, and place orders. The customer can indicate the time they wish to pick up the order, so that restaurants can prepare the food at the correct time.

[Main functions]

(1) Member subscription

- There are two types of members: a customer member (hereinafter, a customer) and a restaurant member (hereinafter, a restaurant).
- To be a member of the application, the customers and restaurants must subscribe to the application.
- The subscription information includes the member type, name, and phone number.
- The application registers customer ID or restaurant ID with a password for each member.

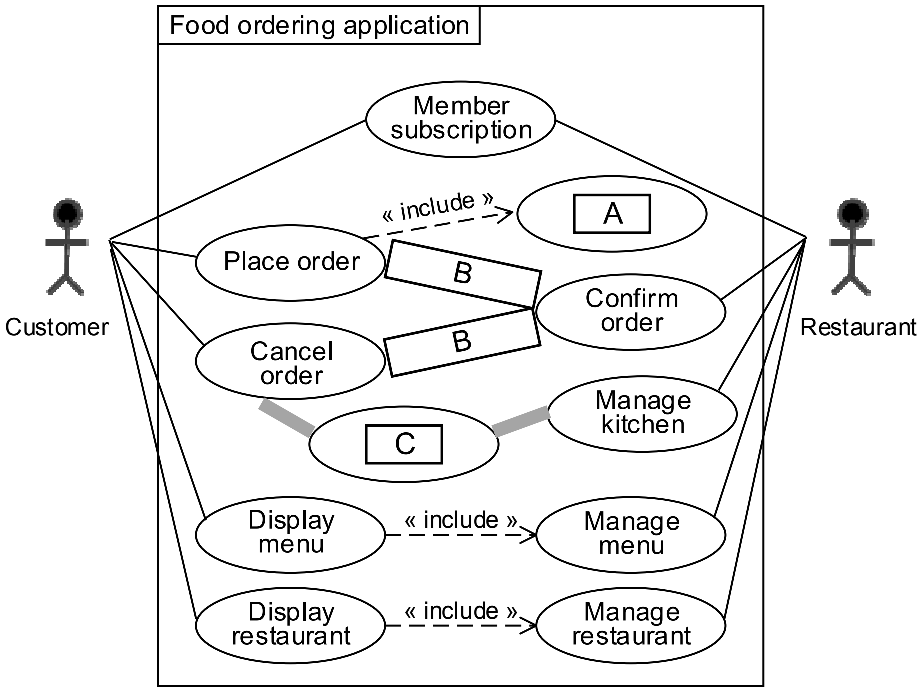
(2) Restaurant type function

- Creating the restaurant information
    - o A restaurant registers its restaurant information that includes name, address, and phone number.
    - o After creating the restaurant information, the restaurant can add a menu to the restaurant information. Each menu is identified by a menu ID.
    - o For each food item on the menu, the restaurant registers the food name, price, food description, and an image of the food item. Each food item is identified by a food ID.
    - o The restaurant information including the menu can be changed or deleted.
- Confirming the customer orders
    - o After a customer has placed an order, an order list is displayed on the restaurant's screen.
    - o The restaurant checks the order list, confirms the availability of each food, and sends a notification to the customer.
    - o When cooking starts (deadline time), the restaurant sends a "Cannot cancel order" message to the customer; the customer cannot cancel the order after this time.

(3) Customer type function

- Placing orders
    - A customer can order food from one restaurant at a time.
    - An order must be placed on the same day of picking up food.
    - To order food, a customer views the menu, selects the food item, and fills-in the number of each food item to be ordered.
    - After a customer confirms the order, the application displays the order summary on the customer's screen. The order summary includes the food name, number of dishes and price for each food item.
    - A customer places the order to the restaurant and informs the restaurant of the time he/she will come to pick up the foods.
    - A customer must wait for the notification from the restaurant.
- Canceling orders
    - A customer can cancel the order before the deadline time.

Figure 1 shows the use case diagram of the food ordering application. A customer has to perform member subscription. After that, the customer can login to the application; the application will then display the list of restaurants. The customer can select a restaurant, display and select food, order (or cancel) some food items, confirm the order, and specify the pickup time the customer will go to the restaurant to collect the food.



Note: shaded parts are not shown

Figure 1 Use case diagram of the food ordering application

Subquestion 1

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

Answer group for A

a) Check acceptability b) Send notification

c) Set pickup time d) Verify member subscription

Answer group for B

a) « extend » <------------ b) « extend » ------------>

c) « include » <------------ d) « include » ------------>

Answer group for C

a) Change order b) Change restaurant

c) Confirm login information d) No cancellation

Subquestion 2

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

To understand the requirement of each function, the system analyst describes the details using a use case description. Table 1 shows the use case description of “Place order”. In “Place order”, after a customer successfully logs in to the application, he/she can start to order food.

Table 1 Use case description of ”Place order”

Use case name	Place order
Primary actor	Customer
Secondary actor	-
Input	Customer ID, Restaurant ID, Menu ID, $\boxed{\text{D}}$, Number of dishes, Pickup time
Output	Order summary as the list of food items.
Trigger	The customer clicks “Place order” button on the customer’s screen.
Availability	Always
Pre-condition	The customer has successfully logged in to the application.
Post-condition	The order list is displayed on the restaurant’s screen.

Table 1 Use case description of ”Place order” (continued)

Basic flow	Actor	Application
	1. Select a restaurant From the list of restaurants, the customer selects a restaurant.	
	2. Select a menu From the available menus such as lunch and dinner, the customer selects a menu.	
	3. Select foods From the list of food items, the customer selects food items. For each selected food item, the customer fills in E and clicks the “Add food” button to add the food to the order.	
		4. Check selected food items For each selected food item, the application checks if E is acceptable or not, and returns the reply “Accepted” or “Not accepted”, accordingly.
	5. Submit the order After the customer confirms all needed food items are added to the basket, the customer fills in F and clicks the “Submit” button to submit this order.	
		6. Check F If it conflicts with the rule, then the application returns the reply “Not acceptable” and prompts the customer to fill it in again. If there is no conflict, the application sends a confirmation request to the restaurant and waits for a reply from the restaurant.
		7. Order completion After the order is confirmed by the restaurant, the application sends a notification to the customer.

Answer group for D

a) Chef ID
b) Deadline time
c) Delivery address
d) Food ID
e) Service charge
f) Total amount

Answer group for E and F

a) amount of tip
b) deadline time
c) delivery address
d) number of dishes
e) pickup time
f) total amount

Question **Q6** is **compulsory**.

Q6. Read the following description of encryption and decryption programs, and then answer Subquestions 1 and 2.

[Caesar cipher]

A Caesar cipher is one of the simplest encryption methods. During encryption (decryption), each alphabetical character in a given plain (cipher) text is replaced by an alphabetical character a given number of positions down (up) the alphabet. It is assumed that a set of uppercase alphabet wraps around from “Z” to “A”, and a set of lowercase alphabet wraps around from “z” to “a”.

Encryption example: If the given number is 2, “A” would be replaced by “C”, “p” would be replaced by “r”, and “Z” would be replaced by “B”.

Decryption example: If the given number is 2, “c” would be replaced by “a”, “R” would be replaced by “P”, and “b” would be replaced by “z”.

[Program Description]

The program consists of the following two functions: `Encrypt` and `Decrypt`.

```
FUNCTION: Encrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd)
FUNCTION: Decrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd)
```

(1) The function `Encrypt` encrypts the plain text given by `input[]` using the keys `keyEven` and `keyOdd`. The cipher text is obtained in `output[]`.
The function `Decrypt` decrypts the cipher text given by `input[]` using the keys `keyEven` and `keyOdd`. The plain text is obtained in `output[]`.

(i) `input[]` and `output[]` are character arrays. The array indexes start at 0.

(ii) `textLen` indicates the number of characters in `input[]` and `output[]`.

(iii) `keyEven` and `keyOdd` are used to determine the number of positions down (up) the alphabet when the text is encrypted (decrypted). The argument values of `keyEven` and `keyOdd` range between 0 and 25.

(2) The steps for encrypting the plain text are as follows:

(i) For each character `input[i]` (*i*: 0, 1, … , `textLen` - 1), repeat (ii).

(ii) If `input[i]` is an alphabetical character (“A” - “Z” or “a” - “z”), find an alphabetical character `keyEven` or `keyOdd` position down the alphabet from `input[i]`, and store it into `output[i]`. Here, `keyEven` is used if *i* is an even number (0, 2, 4, …), and `keyOdd` is used if *i* is an odd number (1, 3, 5, …).
Otherwise, copy `input[i]` to `output[i]`.

(3) The steps for decrypting the cipher text are as follows:

(i) For each character `input[i]` (*i*: 0, 1, … , `textLen` - 1), repeat (ii).

(ii) If `input[i]` is an alphabetical character (“A” - “Z” or “a” - “z”), find an alphabetical character `keyEven` or `keyOdd` position up the alphabet from `input[i]`, and store it into `output[i]`. Here, `keyEven` is used if *i* is an even number (0, 2, 4, …), and `keyOdd` is used if *i* is an odd number (1, 3, 5, …).
Otherwise, copy `input[i]` to `output[i]`.

(4) Characters are represented by ASCII; alphabetical characters “A” - “Z” have consecutive internal codes 65 - 90, and “a” - “z” have consecutive internal codes 97 - 122.

(5) The “%” operator indicates a remainder operation. In this question, it is assumed that the sign of the non-zero remainder is the same as that of the left-hand operand.
For example, 2 % 26 = 2, 30 % 26 = 4, (-2) % 26 = -2, and (-30) % 26 = -4.

(6) Sample input and output examples:

Function called	Input			output
	`input[]`	`keyEven`	`keyOdd`	`output[]`
`Encrypt`	“`ABCxyz`”	2	2	“`CDEzab`”
`Encrypt`	“`ABCxyz`”	2	3	“`CEEaac`”
`Encrypt`	“`non=no(on)`”	3	2	“ A ”
`Decrypt`	“`CDEzab`”	2	2	“`ABCxyz`”
`Decrypt`	“`CEEaac`”	2	3	“`ABCxyz`”
`Decrypt`	“ B ”	3	2	“`f(x)=x+1`”

[Program]

```
FUNCTION: Encrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd) {
   INT: i, key

   FOR (i ← 0; i < textLen; i ← i + 1) {
      IF (    C    )
         key ← keyEven;
      ELSE
         key ← keyOdd;
      IF (input[i] ≥ 'A' and input[i] ≤ 'Z')
         output[i] ←     D1     ;
      ELSE {
         IF (input[i] ≥ 'a' and input[i] ≤ 'z')
            output[i] ←     D2     ;
         ELSE
            output[i] ← input[i];
      }
   }
}

FUNCTION: Decrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd) {
   INT: i, key

   FOR (i ← 0; i < textLen; i ← i + 1) {
      IF (    C    )
         key ← keyEven;
      ELSE
         key ← keyOdd;
      IF (input[i] ≥ 'A' and input[i] ≤ 'Z')
         output[i] ←     E1     ;
      ELSE {
         IF (input[i] ≥ 'a' and input[i] ≤ 'z')
            output[i] ←     E2     ;
         ELSE
            output[i] ← input[i];
      }
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in the above description and the program.

Here, the answers to be inserted in D1 and D2 should be selected as the correct combination from the answer group for D, and the answers to be inserted in E1 and E2 should be selected as the correct combination from the answer group for E.

Answer group for A

a) `pqp=pq(qp)` b) `prp=pr(rp)`

c) `prp=rp(pr)` d) `qqq=qq(qq)`

Answer group for B

a) `h(z)=a+1` b) `h(z)=z+3`

c) `i(a)=a+4` d) `i(a)=z+1`

Answer group for C

a) `i % 2 = 0` b) `i % 2 = 1`

c) `input[i] = 0` d) `input[i] = 1`

Answer group for D

	D1	D2
a)	'A' + (input[i] - 'A' + key - 26) % 26	'a' + (input[i] - 'a' + key - 26) % 26
b)	'A' + (input[i] - 'A' + key) % 26	'a' + (input[i] - 'a' + key) % 26
c)	'A' + (input[i] - 'A' - key + 26) % 26	'a' + (input[i] - 'a' - key + 26) % 26
d)	'A' + (input[i] - 'A' - key) % 26	'a' + (input[i] - 'a' - key) % 26
e)	(input[i] + key) % 26	(input[i] + key) % 26
f)	(input[i] - key) % 26	(input[i] - key) % 26

Answer group for E

	E1	E2
a)	'A' + (input[i] - 'A' + key - 26) % 26	'a' + (input[i] - 'a' + key - 26) % 26
b)	'A' + (input[i] - 'A' + key) % 26	'a' + (input[i] - 'a' + key) % 26
c)	'A' + (input[i] - 'A' - key + 26) % 26	'a' + (input[i] - 'a' - key + 26) % 26
d)	'A' + (input[i] - 'A' - key) % 26	'a' + (input[i] - 'a' - key) % 26
e)	(input[i] + key) % 26	(input[i] + key) % 26
f)	(input[i] - key) % 26	(input[i] - key) % 26

Subquestion 2

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

The algorithms and codes used in the functions `Encrypt` and `Decrypt` are very similar. Therefore, the function `Encrypt` can perform the decryption operation if the arguments are given appropriately.

Consider a case where a main program calls the function `Decrypt` using the following statement. Here, `keyEven` and `keyOdd` values satisfy the condition described in (1) (iii).

```
    Decrypt(input, output, textLen, keyEven, keyOdd);
```

If this statement is replaced by the following statement:

```
    Encrypt(input, output, textLen, 26 - keyEven, 26 - keyOdd);
```

then decryption can be done by the function `Encrypt` in most cases. However, there is a problem that the values of the 4th and/or 5th arguments do not satisfy the condition described in (1) (iii) when the values of `keyEven` and/or `keyOdd` are $\boxed{\text{F}}$.

Moreover, to facilitate a user's convenience, the function `Encrypt` is to be modified so that it can accept any `keyEven` and `keyOdd` values.
Concretely, the following sequence of the statements are added to the function `Encrypt` immediately before the `FOR` loop. These statements convert any given `keyEven` and `keyOdd` values into the prescribed `keyEven` and `keyOdd` values that range between 0 and 25.

/* Converts any `keyEven` value into the prescribed range between 0 and 25. */

$\boxed{\text{G}}$

/* Converts any `keyOdd` value into the prescribed range between 0 and 25. */

▨ // Note: Shaded parts are not shown.

Here, the sequence of the statements $\boxed{\text{G}}$ can be replaced by the following assignment statement:

`keyEven` ← $\boxed{\text{H}}$;

Answer group for F

a) 0
b) 0 or 25
c) 25
d) not 0

Answer group for G

a)
```
IF (keyEven < 0)
   keyEven ← - keyEven;
keyEven ← keyEven % 26;
```

b)
```
IF (keyEven < 0)
   keyEven ← keyEven + 26;
keyEven ← keyEven % 26;
```

c)
```
keyEven ← keyEven % 26;
IF (keyEven < 0)
   keyEven ← - keyEven;
```

d)
```
keyEven ← keyEven % 26;
IF (keyEven < 0)
   keyEven ← keyEven + 26;
```

Answer group for H

a) `((- keyEven % 26) + 26) % 26`
b) `((keyEven % 26) + 26) % 26`
c) `(- keyEven % 26) + 26`
d) `(26 - keyEven) % 26`

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 then answer Subquestions 1 and 2.

The number of dots composing an equilateral (regular) triangle with N dots on a side is called an N-th triangular number (T_N). For example, 10 is the 4th triangular number because 10 dots can form an equilateral triangle with 4 dots on each side. Figure 1 shows examples of triangular numbers.

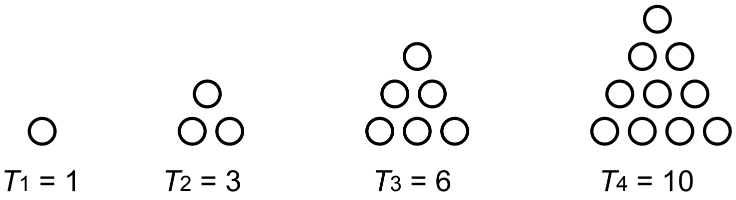


Figure 1 Examples of triangular numbers

The sum of two consecutive triangular numbers $T_{N-1} + T_N$ is known to be N^2. For example, $T_3 + T_4 = 16$, which is 4^2.

[Program Description]

For given input value of N ($1 \le N \le 1000$), the program draws a triangle using the characters `'o'` and `' '` (space character) and determines the triangular number by counting the number of `'o'`s in the triangle. For T_N where $N \ge 2$, it also draws the previous triangle and determines its triangular number, and checks if their summation equals to N^2.

Table 1 shows the triangles drawn by the program for the first four triangles. In Table 1, the character `'.'` indicates a space character.

Table 1 Triangles drawn by the program

N	T_N	Triangle drawn by the program
1	1	`o.`

2	3	`.o.`
		`o.o.`

3	6	`..o.`
		`.o.o.`
		`o.o.o.`

4	10	`...o.`
		`..o.o.`
		`.o.o.o.`
		`o.o.o.o.`

The program outputs the following result for the input value 1.

```
Input value for generating a triangular number: 1
O
T(1) = 1
```

The program outputs the following result for the input value 3.

```
Input value for generating a triangular number: 3
  O
 O O
O O O
T(3) = 6

The previous triangle
 O
O O
T(2) = 3

T(2) + T(3) = 3 ^ 2
Square property held
```

Table 2 shows the description of the functions used in the program.

Table 2 Functions used in the program

Function	Function Description
`int DrawTriangle(int n)`	Draws an equilateral triangle with `n` dots on a side and prints the `n`-th triangular number.
`void PrintSpace(int n)`	Prints `n` space characters.

[Program]

```
#include <stdio.h>

int DrawTriangle(int n);
void PrintSpace(int n);                                                 // α
```

```
int main() {
    int N, S, Ta, Tb;

    printf("Input value for generating a triangular number: ");
    scanf("%d", &N);
    Ta = DrawTriangle([    A    ]);
    if (N > 1) {
        printf("The previous triangle\n");
        Tb = DrawTriangle([    B    ]);
        S = N * N;                                                  // β
        if ([    C    ]) {
            printf("T(%d) + T(%d) = %d ^ 2\n", N - 1, N, N);      // γ
            printf("Square property held\n");
        } else {
            printf("Square property not held\n");
        }
        printf("\n");
    }
    return 0;
}

int DrawTriangle(int n) {
    int i, j, result = 0;

    for (i = 1; i <= n; i++) {
        PrintSpace([    D    ]);
        for (j = 1; [    E    ]; j++) {
            printf("O ");
            result++;
        }
        printf("\n");
    }
    printf("T(%d) = %d\n\n", n, result);
    return result;
}

void PrintSpace(int n) {
    int i;

    for (i = 1; i <= n; i++) {
        printf(" ");
    }
}
```

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 and B

a) `N` b) `N + 1` c) `N – 1` d) `Ta` e) `Ta - 1`

Answer group for C

a) `N > 2` b) `Ta + Tb == S`

c) `Ta == S + Tb` d) `Tb == S + Ta`

Answer group for D

a) `i` b) `i - 1` c) `n - 1`

d) `n - i` e) `n - i + 1`

Answer group for E

a) `j < i` b) `j < i - 1` c) `j <= i` d) `j <= i + 1`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank □ in the following description. It is assumed that the correct answer in Subquestion 1 is inserted in $\boxed{\text{E}}$.

The number of dots composing a square with N dots on a side is called N-th square number (S_{N}). Figure 2 shows examples of square numbers.

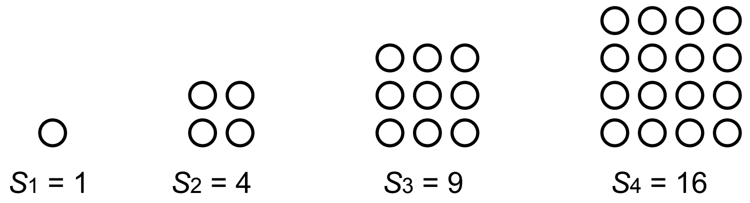


Figure 2 Examples of square numbers

Using the square number, the property of two consecutive triangular numbers can be expressed by the formula $T_{\mathrm{N-1}} + T_{\mathrm{N}} = S_{\mathrm{N}}$.

A new function is to be added so that the program also draws a square using the characters `'o'`, `'*'`, and `' '` (space character), and determines the square number by counting the number of `'o'`s and `'*'`s in the square.

The square with N dots on a side consists of two triangles; the triangle with N dots on a side (drawn with `'o'`s) and the triangle with N - 1 dots on a side (drawn with `'*'`s).

Table 3 shows the description of the new function added to the program.

Table 3 New function added to the program

Function	Function Description
`int DrawSquare(int n)`	Draws a square with `n` dots on a side and prints the `n`-th square number.

Concretely, the program is to be modified as follows:

(1) Immediately after the line designated by α, insert the following line.

```
int DrawSquare(int n);
```

(2) Replace the line designated by β with the following two lines.

```
printf("The square number\n");
S = DrawSquare(N);
```

(3) Replace the line designated by γ with the following line.

```
printf("T(%d) + T(%d) = S(%d)\n", N - 1, N, N);
```

(4) After the last line of the program, insert the following new function.

```
int DrawSquare(int n) {
    int i, j, result = 0;

    for (i = 1; i <= n; i++) {
        for (j = 1;      E      ; j++) {
            printf("o ");
            result++;
        }
        for (j =      F      ; j <= n; j++) {
            printf("* ");
            result++;
        }
        printf("\n");
    }
    printf("S(%d) = %d\n\n", n, result);
    return result;
}
```

After the modification, the program outputs the following result for the input value 3.

```
Input value for generating a triangular number: 3
  O
 O O
O O O
T(3) = 6

The previous triangle
 O
O O
T(2) = 3

The square number

     G

S(3) = 9

     H
Square property held
```

Answer group for F

a) `i` b) `i + 1` c) `i - 1` d) `n - i`

Answer group for G

```
a) * * O      b) * O O      c) O * *      d) O O *
   * O O         * * O         O O *         O * *
   O O O         * * *         O O O         * * *
```

Answer group for H

a) `T(2) + T(3) = 3 ^ 2` b) `T(2) + T(3) = 9`

c) `T(2) + T(3) = S(3)` d) `T(3) + T(2) = 3 ^ 2`

e) `T(3) + T(2) = 9` f) `T(3) + T(2) = S(3)`

Concerning questions **Q7** and **Q8**, **select one** of the two questions.

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

[Program Description]

This is a calculator program that performs addition, subtraction, multiplication, and division of integer values. The calculator has digit keys, arithmetic operation keys for addition, subtraction, multiplication, and division, an equal key, and a clear key. When a key is pressed, the program executes the process corresponding to that key. Numbers etc. are displayed by calling `System.out.println`.

(1) The interface `key` defines the method that executes the process when a calculator key is pressed.
The method `operateOn` executes the process corresponding to the key with respect to the instance of the class `java.util.Stack` (hereinafter, stack) given by the parameter. The class `Stack` supports the method `peek` that returns the object at the top of the stack without removing it from the stack, in addition to the methods `push` and `pop`. The method `clear` removes all the objects from the stack.

(2) The enumeration `DigitKey` defines the constants `DIGIT0` through `DIGIT9` that represent the digit keys.
The method `operateOn` processes the keys as an input of decimal numbers. The value stored at the top of the stack given by the `stack` parameter is 0 (initial value) or the numeric value that has been entered. The method updates this value.

(3) The enumeration `OperationKey` defines the constants `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `EQUAL`, and `CLEAR` that represent the respective arithmetic operation keys for addition, subtraction, multiplication, and division, the equal key, and the clear key.
The method `operateOn` executes the corresponding arithmetic operation with respect to the values on the stack.
Each of the constants, except for `EQUAL` and `CLEAR`, defines its operation using the functional interface `java.util.function.BinaryOperator<T>`. Its method `apply` takes two parameters and returns the result of the operation. The type of the parameters and the return value is specified by type `T`.

(4) The class `Calculator` represents the calculator unit. The field `stack` holds the stack that represents the status of numerical values inside the calculator. The field `pendingKey` holds an arithmetic operation key until the input of numerical values required for the arithmetic operation is completed. Moreover, it holds the equal key when the equal key is pressed.

For example, when the constants of the keys DIGIT2, ADD, and DIGIT4 are processed in sequence, the values stored on the stack are 4 and 2 from the top, and the value of pendingKey is ADD. When the constant of the key EQUAL is processed next, the addition process of the arithmetic operation key ADD is executed. Then, the value stored on the stack becomes 6, and the value of pendingKey becomes EQUAL. Here, the results of the arithmetic operations with respect to two numerical values follow the results of int type arithmetic operations of Java.

The method onKeyPressed is called when a calculator key is pressed. The key pressed is given by the parameter. A process is executed on the basis of the key pressed and the internal status of the calculator.

(5) The class CalculatorTest is a program for testing the class Calculator.
First, the method main creates the mappings between characters and their corresponding key constants; it creates an instance of the class Calculator. Next, it converts each character in the string given by the parameter into its corresponding key constant. It then calls the method onKeyPressed of the Calculator instance with the converted key constant. For example, when a string "2*3=" is given by the parameter of the method main, each character is converted into its corresponding key constant, and the given string is converted into a sequence of the key constants DIGIT2, MULTIPLY, DIGIT3, and EQUAL, and the method onKeyPressed is called. Figure 1 shows the output when the method main is executed.

```
DIGIT2
2
MULTIPLY
2
DIGIT3
3
EQUAL
6
```

Figure 1 Output when the method main is executed

[Program 1]

```
import java.util.Stack;

public interface Key {
   public void operateOn(Stack<Integer> stack);
}
```

[Program 2]

```
import java.util.Stack;

enum DigitKey    A    Key {
   DIGIT0, DIGIT1, DIGIT2, DIGIT3, DIGIT4,
   DIGIT5, DIGIT6, DIGIT7, DIGIT8, DIGIT9;

   public void operateOn(Stack<Integer> stack) {
      stack.push(    B    * 10 +    C    );
   }
}
```

[Program 3]

```
import java.util.Stack;
import java.util.function.BinaryOperator;

enum OperationKey    A    Key {
   ADD((val1, val2) -> val1 + val2),
   SUBTRACT((val1, val2) -> val1 - val2),
   MULTIPLY((val1, val2) -> val1 * val2),
   DIVIDE((val1, val2) -> val1 / val2),
   EQUAL(null),
   CLEAR(null);

   private final BinaryOperator<Integer> operator;

   OperationKey(BinaryOperator<Integer> operator) {
      this.operator = operator;
   }

   public void operateOn(Stack<Integer> stack) {
      if (    D    ) {
         Integer val2 = stack.pop();
         Integer val1 = stack.pop();
         stack.push(operator.apply(val1, val2));
      }
   }
}
```

[Program 4]

```
import java.util.Stack;

public class Calculator {
   private final Stack<Integer> stack = new Stack<>();
   private Key pendingKey;

   public Calculator() {
      stack.push(0);
   }

   public void onKeyPressed(Key key) {
      System.out.println(key);
      if (key instanceof DigitKey) {
         if (pendingKey == OperationKey.EQUAL) {
            reset();
         }
         key.operateOn(stack);
         System.out.println(stack.peek());
      } else if (key == OperationKey.CLEAR) {
         reset();
         System.out.println(stack.peek());
      } else {
         try {
            if (pendingKey != null) {
               pendingKey.operateOn(stack);
            }
            System.out.println(stack.peek());
            pendingKey = key;
            if (key != OperationKey.EQUAL) {
               stack.push(0);
            }
         } catch (ArithmeticException e) {
            System.out.println("Error");
            reset();
         }
      }
   }

   private void reset() {
      stack.clear();
      stack.push(0);
      pendingKey = null;
   }
}
```

[Program 5]

```
import java.util.HashMap;
import java.util.Map;

public class CalculatorTest {
   public static void main(String[] args) {
      Map<Character,    E    > map = new HashMap<>();

      // Store the relation between characters and constants of OperationKey into map.
      for (OperationKey key : OperationKey.values()) {
         map.put("+-*/=C".charAt(key.ordinal()), key);
      }

      // Store the relation between numbers and constants of DigitKey into map.
      for (DigitKey key : DigitKey.values()) {
         map.put("0123456789".charAt(key.ordinal()), key);
      }

      Calculator calc = new Calculator();
      String chars = args[0];
      // Convert each char. of chars into the constant of the key, and call onKeyPressed.
      for (int i = 0; i < chars.length(); i++) {
         calc.onKeyPressed(map.get(chars.charAt(i)));
      }
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in the above programs.

Answer group for A

a) `extends`　　b) `implements`　　c) `imports`

d) `inherits`　　e) `requires`　　f) `throws`

Answer group for B and C

a) `ordinal()`　　b) `stack.peek()`

c) `stack.pop()`　　d) `stack.push(0)`

e) `stack.push(ordinal())`　　f) `values()`

Answer group for D

a) `operator != null`	b) `operator == null`	c) `stack != null`
d) `stack == null`	e) `this != operator`	f) `this == operator`

Answer group for E

a) `Calculator`	b) `Character`	c) `DigitKey`
d) `Integer`	e) `Key`	f) `OperationKey`

Subquestion 2

Table 1 shows the last row of output (in case of Figure 1, it is 6) when the method `main` is executed with a string as the parameter. From the answer group below, select the correct answer to be inserted in each blank ▭ in Table 1. Here, assume that the correct answers are filled in all blanks ▭ in the programs.

Table 1 Character string (parameter) and output (last row)

Character string (parameter)	Output (last row)
`2*6/3=`	4
`-2=`	-2
`2*4==`	F
`2*4C2=`	G
`8/2/=`	H

Answer group for F through H

a) 0	b) 2
c) 4	d) 8
e) 16	f) 32
g) 64	h) `ArithmeticException`
i) `Error`	

Company names and product names appearing in the test questions are trademarks or registered trademarks of their respective companies. Note that the ® and ™ symbols are not used within the text.