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.

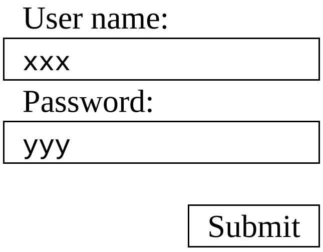
Q1. Read the following description concerning a security vulnerability, and then answer Subquestions 1 and 2.

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 $\boxed{\quad \text{A} \quad}$ 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.



User name: xxx

Password: yyy

Figure 1 Simple sign-in form

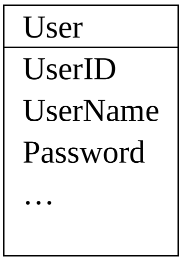


Figure 2 Table named User

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 □ 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 □ in the following description.

To strengthen the protection against SQL injection attacks such as cases 1 and 2 described in Subquestion 1, one should E.

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 (▨) 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 $\boxed{\quad\text{A}\quad}$, 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:

$$(\neg I_1 \bullet \neg I_2 \bullet \neg I_3) + (I_1 \bullet I_2 \bullet I_3) \qquad \dots (*1)$$

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 $\boxed{\quad\text{B}\quad}$, 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 $\boxed{}$ 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 □ 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.

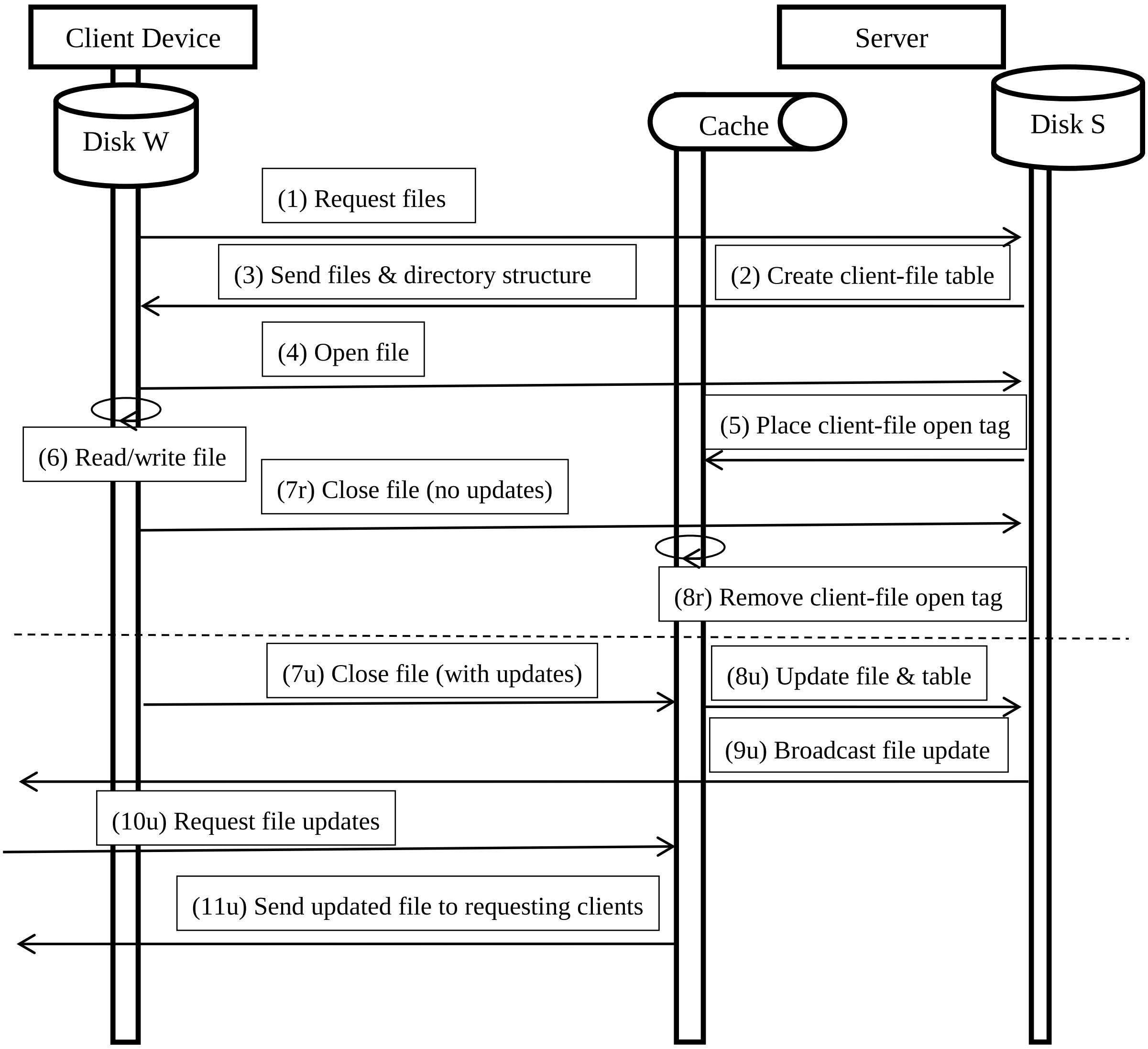

Client Device
Server
Disk W
Cache
Disk S
(1) Request files
(3) Send files & directory structure
(2) Create client-file table
(4) Open file
(5) Place client-file open tag
(6) Read/write file
(7r) Close file (no updates)
(8r) Remove client-file open tag
(7u) Close file (with updates)
(8u) Update file & table
(9u) Broadcast file update
(10u) Request file updates
(11u) Send updated file to requesting clients

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 $\boxed{}$ 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 $\boxed{\text{A}}$ to know the files each device has. In the event that some of the files are updated by the server, the $\boxed{\text{B}}$ 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 $\boxed{\text{C}}$ and $\boxed{\text{D}}$ will increase, due to the process of step $\boxed{\text{E}}$ 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.

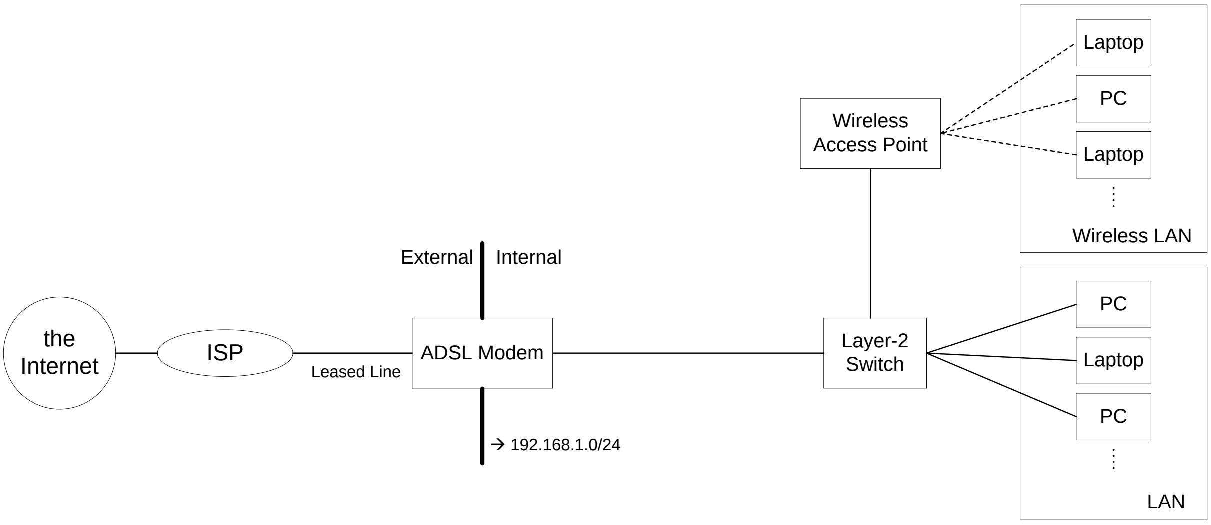


Figure 1 Current network configuration

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.

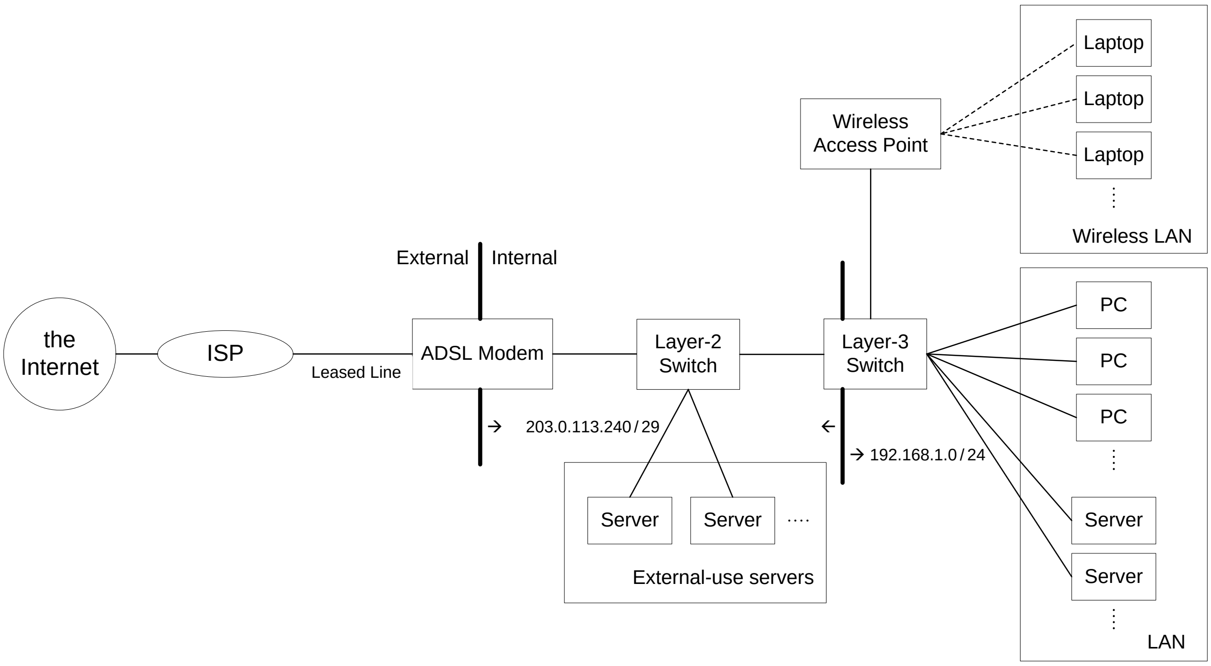


Figure 2 Planned network configuration

In this planned network configuration, Company X can use a maximum of $\boxed{\quad\text{C}\quad}$ 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 $\boxed{\quad\text{D}\quad}$ still remains.

If WPA2-Enterprise is used, the level of security will be improved, but the installation of $\boxed{\quad\text{E}\quad}$ 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)                           (Example 2)

Bit number 7 6 5 4 3 2 1 0            Bit number 7 6 5 4 3 2 1 0
     Data  0 1 0 1 0 1 0 1                 Data  0 0 1 1 0 0 1 1
           | | |                                       |       |
     Mask  1 1 1 0 0 0 0 0                 Mask  0 0 0 1 0 0 0 1

Return value   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 □ in Program 1.

Answer group for A and B

a) `(Data & Mask) = "00000000"B`

b) `(Data & Mask) = Data`

c) `(Data & Mask) = Mask`

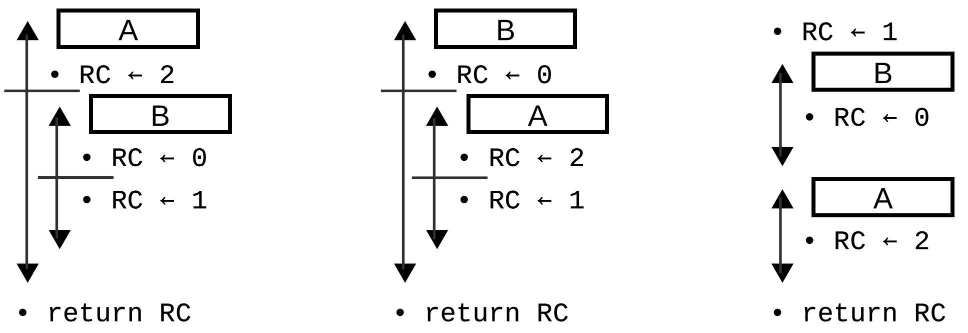
d) `(Data | Mask) = "00000000"B`

e) `(Data | Mask) = Mask`

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank ▭ 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 [A] and [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 [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 □ 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`.

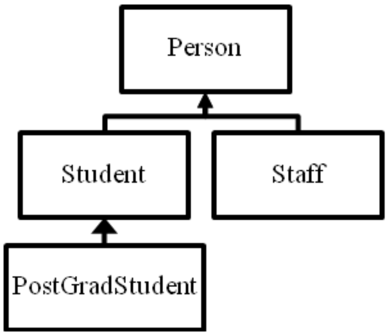


Figure 1 Inheritance relation

[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 □ 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 □ 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 □ 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.