October 2020

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

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

Instructions:

1. Use a pencil. If you need to change an answer, erase your previous answer completely and neatly. Wipe away any eraser debris.

2. Mark your examinee information and test answers in accordance with the instructions below. Your answer will not be graded if you do not mark properly. Do not mark or write on the answer sheet outside of the prescribed places.

   (1) **Examinee Number**
   Write your examinee number in the space provided, and mark the appropriate space below each digit.

   (2) **Date of Birth**
   Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.

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

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

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

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

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

   [Sample Answer]

Sample	ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

Do not open the exam booklet until instructed to do so.

Inquiries about the exam questions will not be answered.

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.

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.

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

Q1. Read the following description of the use of public-key cryptography in e-mail communications, and then answer Subquestions 1 and 2.

Pretty Good Privacy (PGP) was developed to enhance privacy, integrity, and authentication of e-mail communications. The term "PGP" refers not only to the protocol specification, but it also refers to the name of software package. There are open source implementations of PGP (e.g., OpenPGP and GPG) as well. However, for the purpose of this question, all implementations are simply referred to as "PGP".

PGP combines the use of public- and secret-key cryptography to provide message protection and authentication. It also employs data compression to reduce the size of messages. Figure 1 demonstrates the process in which Alice sends a confidential message to Bob.

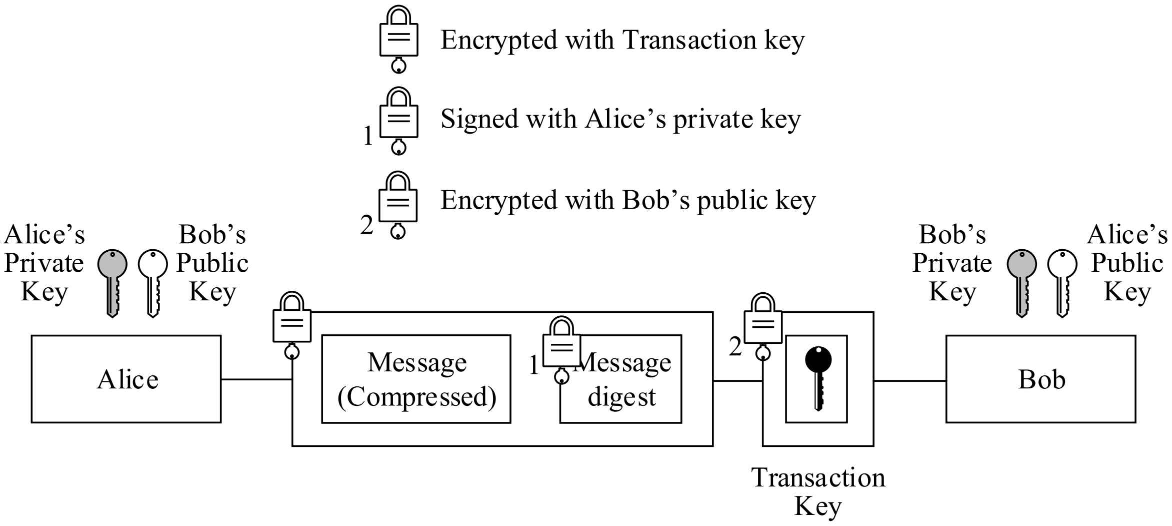


Figure 1 Process of Alice sending a confidential message to Bob

In PGP, the confidentiality of the message is achieved through the use of a transaction key, also known as a shared-session key. This key is a random number created on the sender's side through the use of arbitrary keystrokes as the seed. It is used only once with this particular message. One of the reasons for using transaction keys is that symmetric-key encryption is more efficient than public-key encryption when encrypting larger messages. Meanwhile, authenticity and integrity of the message are ensured via the use of public keys.

Here, the message is compressed using a compression algorithm. Then, the message digest is created using a hash algorithm from the compressed message and is signed with the sender's private key, which forms a digital signature. The entire packet of the compressed message and the signed message digest are then encrypted with the transaction key. The transaction key is required for message decryption and must be sent with the message. Thus, it is encrypted with the recipient's public key to ensure that the recipient is the only person able to use the transaction key.

The algorithm used in each step was selected at send time. Although multiple algorithms are available to choose from, the ones used in this example are shown in table 1.

Table 1 Algorithms used during the sending process

Type	Algorithm
Public-key algorithm	RSA
Symmetric-key algorithm	Triple DES
Hash algorithm	SHA-1
Compression algorithm	ZIP

After Bob receives the message from Alice, all processes are reversed. First, the transaction key must be decrypted with $\boxed{\text{A}}$ using the $\boxed{\text{B}}$ algorithm. Second, the transaction key is used to decrypt the message's envelope, which contains both the compressed message and the signed message digest. Third, Bob must create a message digest of the compressed message using $\boxed{\text{C}}$, and compares the result with the message digest verified using the RSA algorithm and $\boxed{\text{D}}$. This is performed to ensure the authenticity and integrity of the message. Finally, the message is uncompressed to obtain the original message sent by Alice.

[Web of trust]

With PGP, each party is required to maintain individual keyrings, mapping each e-mail address to its public key. Typically, e-mail clients automate the key selection process but it is the user's duty to gather and exchange public keys among related parties rather than using a centralized server. Each entity in the keyring will be either fully trusted, partially trusted or untrusted. Each user can also sign their contact's public key and send to another party for introduction. For example, if Alice wants to introduce Mark to Bob, then Mark's public key will be signed with $\boxed{\text{E}}$ and sent to Bob. Ultimately, this will form a web of trust that will enable a large circle of people to communicate securely with one another.

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, D and E

a) Alice's private key
b) Alice's public key
c) Bob's private key
d) Bob's public key
e) Random number
f) Transaction key

Answer group for B and C

a) Random number generator
b) RSA
c) SHA-1
d) Triple DES
e) ZIP

Subquestion 2

From the answer group below, select the appropriate cause for concern regarding the use of PGP to secure e-mail communication.

Answer group

a) Its decentralized nature makes it easier for malicious users to create fake identities especially prior to a web of trust being formed.
b) Its hybrid cryptography characteristic requires the entire message to be encrypted *n* times if the message has *n* recipients.
c) Messages with large attachments will take longer to encrypt, compared with a system using pure public key cryptography.
d) The transaction key is required to be kept private because it can be used to decrypt all subsequent messages sent to the same recipient.
e) The use of compression makes the message more vulnerable to crypto-analytic attack, owing to the repetitive nature of the compressed content.

Q2. Read the following description of electronic circuits for a traffic signal, and then answer Subquestions 1 and 2.

Note: In this question, digits in italics (*0* and *1*) indicate binary digits, and the symbols “•”, “+”, and “¯” indicate the logical operators AND, OR, and NOT, respectively.

Figure 1 shows an example of a traffic signal that has three lights (red, yellow, and green), and a count-down timer which shows how many seconds later the signal will change. The traffic signal uses two types of electronic circuits. One turns on/off the lights and the other displays digits, each using a 7-segment LED.

[Turning on/off the lights]

This circuit has two input lines, S_1 and S_2, and three output lines, Red, Green, and Yellow. If S_1 is *1* and S_2 is *0*, the circuit outputs *1* to Red and *0* to Green and Yellow to illuminate the Red signal. If S_1 is *0* and S_2 is *1*, the circuit outputs *1* to Green and *0* to Red and Yellow to illuminate the Green signal. If both S_1 and S_2 are *1*, the circuit outputs *1* to Yellow and *0* to Red and Green to illuminate the Yellow signal.

Table 1 shows the truth table that turns on/off the lights.

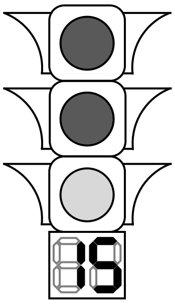


Figure 1 Example of a traffic signal

Table 1 Truth table for turning on/off the lights

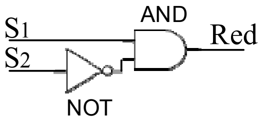
Inputs		Outputs		
S_1	S_2	Red	Green	Yellow
0	*0*	*0*	*0*	*0*
0	*1*	*0*	*1*	*0*
1	*0*	*1*	*0*	*0*
1	*1*	*0*	*0*	*1*

Subquestion 1

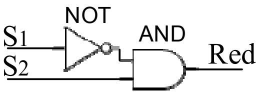
From the answer group below, select the correct circuit for the Red output in Table 1.

Answer group

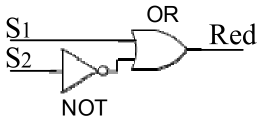
a)



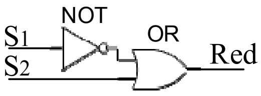
b)



c)



d)



Subquestion 2

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

[Displaying a digit on a 7-segment LED]

A 7-segment LED displays a digit (0 – 9) by turning on/off each segment. Figure 2 shows a 7-segment LED which displays "5". In Figure 2, black segments indicate the turned-on status, whereas the white segments indicate the turned-off status. Segments are identified by the segment names shown in Figure 3.

This circuit has four input lines, I_3, I_2, I_1, and I_0, and seven output lines, A – G. Table 2 shows 4-bit input values and corresponding output values. For example, if the digit to be displayed is "9", its binary value is *1001* ($I_3 = 1$, $I_2 = 0$, $I_1 = 0$, $I_0 = 1$). Then, all segments other than E are set to 1 (turned-on).

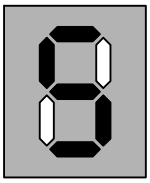


Figure 2 7-Segment LED displaying "5"

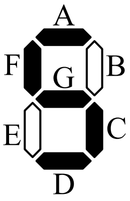


Figure 3 Segment names of 7-segment LED

Table 2 Outputs on segments A – G

Input		Output						
Digit	$I_3 I_2 I_1 I_0$	A	B	C	D	E	F	G
0	*0 0 0 0*	*1*	*1*	*1*	*1*	*1*	*1*	*0*
1	*0 0 0 1*	*0*	*1*	*1*	*0*	*0*	*0*	*0*
2	*0 0 1 0*	*1*	*1*	*0*	*1*	*1*	*0*	*1*
3	*0 0 1 1*	*1*	*1*	*1*	*1*	*0*	*0*	*1*
4	*0 1 0 0*	*0*	*1*	*1*	*0*	*0*	*1*	*1*
5	*0 1 0 1*	A		*1*	*1*	*0*	*1*	*1*
6	*0 1 1 0*	*1*	*0*	*1*	*1*	*1*	*1*	*1*
7	*0 1 1 1*	*1*	*1*	*1*	*0*	*0*	*0*	*0*
8	*1 0 0 0*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
9	*1 0 0 1*	*1*	*1*	*1*	*1*	*0*	*1*	*1*
10 to 15	*1 0 1 0* to *1 1 1 1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Consider a logical expression that determines the output value of segment E. From Table 2, the output value of segment E is *1* if the input value is 0, 2, 6, or 8; else the output value of segment E is *0*. Therefore, a logical expression that outputs *1* when the input value is 0, 2, 6 or 8 can be described as Expression 1:

Output *1* when input value is:

Expression 1:

$$\overbrace{\overline{I_3} \bullet \overline{I_2} \bullet \overline{I_1} \bullet \overline{I_0}}^{0} + \overbrace{\overline{I_3} \bullet \overline{I_2} \bullet I_1 \bullet \overline{I_0}}^{2} + \overbrace{\boxed{\quad \text{B} \quad}}^{6} + \overbrace{I_3 \bullet \overline{I_2} \bullet \overline{I_1} \bullet \overline{I_0}}^{8}$$

Figure 4 shows this state in matrix form. This matrix is called a "Karnaugh map". Note that in Figure 4, outputs of *0* are omitted, and the left-most 2 bits and the right-most 2 bits of the input value (shaded parts) are not arranged in ascending order.

		Input $I_1\ I_0$				
		0 0	*0 1*	*1 1*	*1 0*	
Input $I_3\ I_2$	*1 1*					
	1 0	*1*				(i)
	0 0	*1*			*1*	(ii)
	0 1				*1*	

Figure 4 Output values of segment E

In the Karnaugh map, shown as (i) and (ii) in Figure 4, if there is a pair of adjacent *1*s (vertically or horizontally), the pair's two input values differ only by one bit.

Based on this property, in (i) of Figure 4, the first and the last terms of Expression 1 (for input values 0 and 8) can be combined into a simple term $\overline{I_2} \bullet \overline{I_1} \bullet \overline{I_0}$ that outputs *1* when $I_2\ I_1\ I_0$ is *000* implying that the input value is either 0 or 8. Similarly, in (ii) of Figure 4, the second and the third terms of Expression 1 (for input values 2 and 6) can be combined into a simple term $\overline{I_3} \bullet I_1 \bullet \overline{I_0}$ that outputs *1* when $I_3\ I_1\ I_0$ is $\boxed{\quad \text{C} \quad}$.

Consequently, Expression 1 can be simplified as Expression 2:

Expression 2: $\overline{I_2} \bullet \overline{I_1} \bullet \overline{I_0} + \overline{I_3} \bullet I_1 \bullet \overline{I_0}$

Answer group for A

a) $\boxed{\mathit{0}\,|\,\mathit{0}}$ b) $\boxed{\mathit{0}\,|\,\mathit{1}}$ c) $\boxed{\mathit{1}\,|\,\mathit{0}}$ d) $\boxed{\mathit{1}\,|\,\mathit{1}}$

Answer group for B

a) $I_3 \bullet I_2 \bullet \overline{I_1} \bullet \overline{I_0}$ b) $I_3 \bullet \overline{I_2} \bullet \overline{I_1} \bullet I_0$ c) $\overline{I_3} \bullet I_2 \bullet I_1 \bullet \overline{I_0}$ d) $\overline{I_3} \bullet \overline{I_2} \bullet I_1 \bullet I_0$

Answer group for C

a) *010* b) *011* c) *100* d) *101*

Q3. Read the following description of a database for an airline company, and then answer Subquestions 1 and 2.

Airline company U has built a database system to manage data related to its business. Currently, the database uses four tables. The table structure and sample data of each table are shown below.

(1) Flight table

Flight table contains information about each flight, including a unique flight code, departure airport, arrival airport, and flight length (km).

FlightCode	DepartureAirport	ArrivalAirport	Length
FE276	DAD	CXR	1283
FE280	SGN	HPH	11979
FE338	SGN	BMV	4081
FE374	HAN	VII	510
FE375	VII	CXR	752
FE440	SGN	BMV	4081
FE464	SGN	DLI	2002

(2) Airplane table

The airline owns many types of airplanes. Airplane table contains information about each airplane, including a unique airplane code, type, and distance that is the farthest distance (km) the airplane can fly without refueling.

AirplaneCode	Type	Distance
154	Aero Jet 154	6565
319	Airbus A319	2888
320	Airbus A320	4168
340	Airbus A340 - 300	11392
727	Boeing 727	2406
737	Boeing 737 - 800	5413

(3) Employee table

The airline has many crew members, including pilots and attendants. Employee table contains information about each employee, including a unique employee code and name.

EmployeeCode	Name
27487	Mai Quoc Minh
35554	Tran Thi Hoai An
35618	Nguyen Vinh Bao
55015	Nguyen Thi Cam
55245	La Que
56735	Quan Cam Ly

(4) Certificate table

If the employee is a pilot, he or she has one or more certificates describing his or her ability to operate a particular airplane. A pilot can operate only certified airplanes.

EmployeeCode	AirplaneCode
35554	727
35618	319
55015	154
55245	320
55245	737
56735	340
56735	737

Subquestion 1

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

The SQL statement named SQL1 outputs all flights that can be operated by any type of Airbus airplane.

```
-- SQL1 -
SELECT FlightCode, Length
FROM Flight
WHERE Length [   A   ]  (SELECT Distance FROM Airplane
                         WHERE Type [   B   ])
```

From the sample data on the previous page, SQL1 creates the following output:

```
FlightCode  Length
FE276         1283
FE374          510
FE375          752
FE464         2002
```

Here, the percent sign wildcard `'%'` represents any string of zero or more characters.
For example, `'o%f'` determines the character string starting with `'o'` and end with `'f'` such as `of` and `off`.

Answer group for A

a) `<=` b) `<= ALL`

c) `>=` d) `>= ALL`

Answer group for B

a) `IN ('%Airbus')` b) `IN ('Airbus%')`

c) `LIKE '%Airbus'` d) `LIKE 'Airbus%'`

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank ☐ in the following SQL statement. Here, assume that the correct answer is inserted in blank B in the following SQL statement.

The airline is planning to select suitable pilots for a special flight for a group of tourists. The SQL statement named SQL2 outputs a list of pilots that can operate airplanes that can fly 3200 km or farther. One of them is an Airbus.

```
-- SQL2 -
SELECT DISTINCT Name
FROM Employee E, Certificate C
WHERE E.EmployeeCode = C.EmployeeCode
  AND AirplaneCode IN (SELECT A.AirplaneCode FROM Airplane A
                       WHERE [  C  ])
  AND [  D  ] (SELECT 1 FROM Airplane A
                 WHERE [  E  ]
                   AND Type [  B  ])
```

From the sample data on the previous pages, SQL2 creates the following output:

```
Name
La Que
Quan Cam Ly
```

Answer group for C and E

a) `C.AirplaneCode = A.AirplaneCode`

b) `C.EmployeeCode = E.EmployeeCode`

c) `Distance >= 3200`

Answer group for D

a) `EXISTS`

b) `IN`

c) `NOT EXISTS`

d) `NOT IN`

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

Company V has a branch office (BO) located about 20 km from the head office (HO). The branch office has a LAN (BO LAN) of 50 users and the head office has a LAN (HO LAN) of 80 users who share devices and information. The head office has a DNS server and a Web server (www.example.com) in the server network to facilitate users of both LANs to use information stored in the Web server. BO LAN is connected to the router R1. Both HO LAN and the server network are connected to the router R2. R1 and R2 are connected each other. PCs in BO LAN obtain IP addresses automatically from the DHCP server. PCs in HO LAN are configured statically.

Figure 1 shows the network connectivity and Table 1 shows the information about networks and devices.

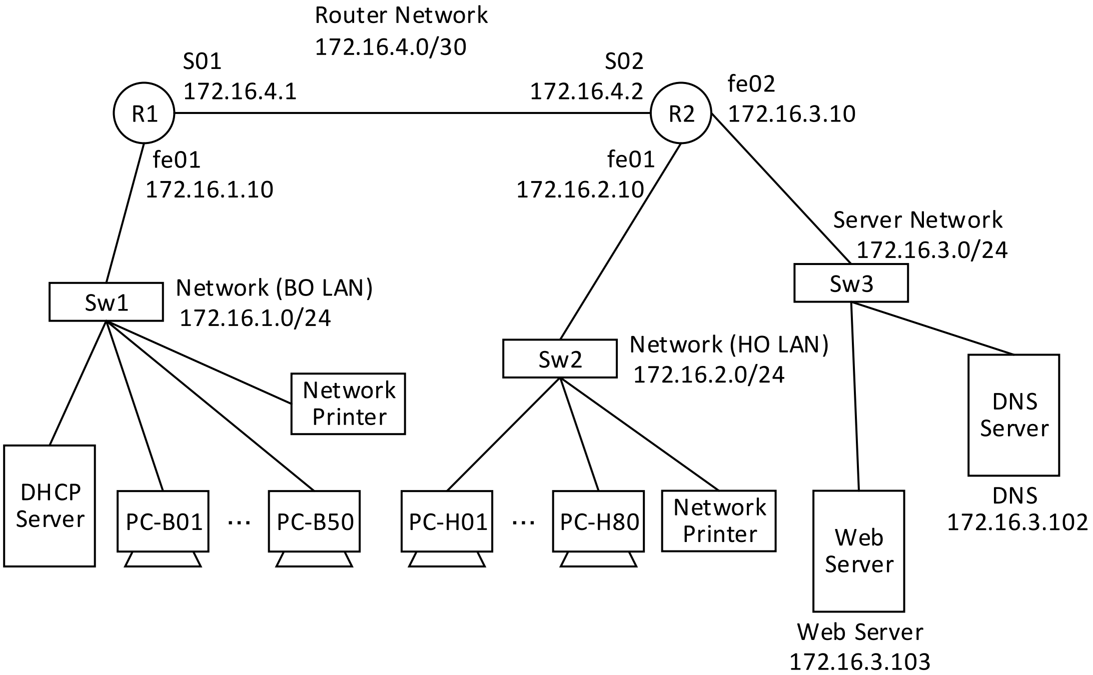


Figure 1 Network connectivity

Table 1 Information about networks and devices

Network name (Location)	Network address	Devices	Gateway
BO LAN (Branch office)	172.16.1.0/24	PC-B01 to PC-B50,DHCP server, A network printer	Fast Ethernet 01(fe01) of R1
HO LAN (Head Office)	172.16.2.0/24	PC-H01 to PC-H80, A network printer	Fast Ethernet 01(fe01) of R2
Server Network (Head Office)	172.16.3.0/24	DNS Server, Web Server	Fast Ethernet 02(fe02) of R2
R1 to R2	172.16.4.0/30	R1 and R2	

Subquestion 1

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

The IP addresses are sub-netted to separate the network layer broadcast domain and enable routers to route the messages. When a new PC in BO LAN requires an IP address and corresponding network information from DHCP server, it broadcasts a DHCP discovery message to ensure that the message can reach all machines in the same IP network. In this case, the destination IP address is 255.255.255.255. When a new PC broadcasts a DHCP discovery message, the PC does not have any IP address. Therefore, it uses 0.0.0.0 as its source IP address. In this case, the DHCP discovery message reaches all machines in [A].

A new network printer is connected to the network switch of BO LAN to allow its users to print with it. The users installed the printer driver from supplied DVD media and attempted to set it up. However, they failed. The network administrator discovered that the default IP address (10.0.0.1) and the default subnet mask (255.0.0.0) of the printer was not changed to enable users of BO LAN to print with it. The network administrator reconfigured the printer IP address. He assigned the last assignable IP address [B] of the BO LAN and the subnet mask [C] to the printer. Users can now print their documents using this printer.

Answer group for A

a) BO LAN
b) BO LAN and HO LAN
c) BO LAN and Server network
d) BO LAN, HO LAN, and Server network
e) HO LAN
f) HO LAN and Server network
g) Server network

Answer group for B and C

a) 0.0.0.0
b) 172.16.1.0
c) 172.16.1.254
d) 172.16.1.255
e) 172.16.255.254
f) 172.16.255.255
g) 255.255.0.0
h) 255.255.255.0
i) 255.255.255.255

Subquestion 2

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

The routing tables in both routers are configured as static routing. Users of both LANs are complaining about various issues. The hosts in BO LAN cannot access to the Web server (www.example.com). They also cannot print documents to the printer located at the head office. However, the hosts at the head office can print with it.

The network administrator checked the DNS server and Web server configurations and found that everything is functional. The routing table entries were also checked at each router and were found to have some configuration errors.

Table 2 shows the simplified routing information.

Table 2 Simplified routing information at R1 and R2

Entry	Router	Routing table entries		
		Destination Network	Next hop	Local interface
(i)	R1	172.16.1.0/24	Directly Connected	fe01
(ii)		172.16.4.0/30	Directly Connected	s01
(iii)		172.16.2.0/24	172.16.2.10	s01
(iv)		172.16.3.0/24	172.16.3.10	s01
(v)	R2	172.16.2.0/24	Directly Connected	fe01
(vi)		172.16.3.0/24	Directly Connected	fe02
(vii)		172.16.4.0/30	Directly Connected	s02
(viii)		172.16.1.0/24	172.16.4.1	s02

The network administrator corrected the routing table entry(entries) by changing [D] of Table 2. Now the users in BO LAN can print documents on the printer located at the head office. However, they still cannot browse the Web server when using the server name www.example.com. It was found that they can browse the Web server once they browse it by using the address http://172.16.3.103. However, the users in HO LAN can browse the Web server by using the server name http://www.example.com and the address http://172.16.3.103. The network administrator has found that [E] is incorrect and has corrected it. Now, users have no problem with the network.

Answer group for D

a) next hop to 172.16.4.1 of entry (vi)
b) next hop to 172.16.4.2 of entries (iii) and (iv)
c) next hop to 172.16.4.2 of entry (vi)
d) next hop to 172.16.4.2 of entry (viii)

Answer group for E

a) DNS server address provided by the DHCP server
b) gateway address configured in the network configuration on the Sw3
c) gateway address configured in the network configuration on the Web server
d) IP address of the name ‘www.example.com’ in the DNS server configuration

Q5. Read the following description of a travel service, and then answer Subquestion.

Company W provides a worldwide cruising service. Passengers can include children, handicapped persons, and tour guides.
Before boarding, passengers first check in. Then, they pass through the security screening station. There are two ways of checking-in: counter check-in and kiosk check-in.
Sometimes, passengers have no luggage. Thus, their check-in and handling are optional during check-in. If passengers have baggage, they can also perform baggage check-in during their check-in process.
Company W offers optional services for passengers with special needs. Passengers must pay charges for these optional services.

Figure 1 shows the use case diagram of the system, and Table 1 shows the use case relationships used in Figure 1.

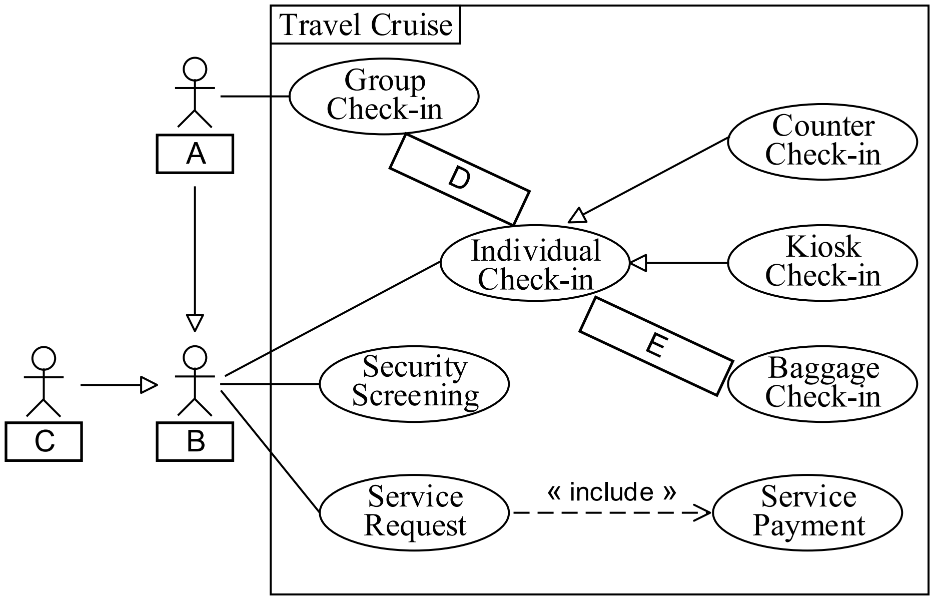


Figure 1 Use case diagram of the system.

Table 1 Use case relationships

Relationship	Symbol	Meaning
Generalization	X ◁——— Y	Y is a more specific version of X.
Extend	« extend » X <- - - - - - - - - Y	X is an additional (optional) function of Y.
Include	« include » X <- - - - - - - - - Y	Y has X, i.e., X is included in Y.

Table 2 shows the description of each use case shown in Figure 1

Table 2 Description of each use case shown in Figure 1

No.	Description
1	Group Check-in Tour guides perform check-in for their group passengers.
2	Individual Check-in Passengers perform check-in via counter or machine.
3	Counter Check-in A passenger can check-in at the check-in counter.
4	Kiosk Check-in A passenger can check-in using a kiosk machine.
5	Baggage Check-in Passengers can perform baggage check-in during the check in process.
6	Security Screening Passengers must pass security screening before boarding.
7	Service Request Passengers can request optional services provided on the ship.
8	Service Payment Passengers must pay for requested optional services.

Subquestion

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

Answer group for A through C

a) Baggage
b) Check-in counter
c) Kiosk machine
d) Passenger
e) Passenger with special needs
f) Tour guide

Answer group for D and E

a) « extend » <- - - - - - - - - - - -
b) « extend » - - - - - - - - - - - ->
c) « include » <- - - - - - - - - - - -
d) « include » - - - - - - - - - - - ->

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

[Program Description]

The function `listByID(S[], H[], N)` prints student IDs in ascending order by body height. Here, `S[i]` contains the `i`-th student's student ID, `H[i]` contains the `i`-th student's body height (cm), and `N` (1 ≤ `N` < 100) is the total number of students. All elements of `S[]` and `H[]` are positive integers less than or equal to 10000. There is no duplicate student IDs. Table 1 shows an example of students' student IDs and body heights.

Table 1 Example of student ID and body height

Index `i`	1	2	3	4	5
`i`-th student's student ID `S[]`	1674	5421	8390	3065	2422
`i`-th student's body height `H[]`	180	165	174	158	171

The following shows the output of the function `listByID(S, H, 5)` with parameters `S[]` and `H[]`, as shown in Table 1.

```
The result is: 3065, 5421, 2422, 8390, 1674
```

(1) The program uses a quicksort algorithm to sort the information in ascending order by body height. The function `quickSort(S[], H[], start, end)` sorts the subarray (it consists of elements from index `start` to index `end` of the array `H[]`) using the function `partition`.

(2) The function `quickSort(S[], H[], start, end)` calls the function `partition(S[], H[], start, end)` to compute the index `pivot`, which is used to split the subarray `H[]` (it consists of elements from index `start` to index `end` of the array `H[]`) and returns the index `pivot`. It does this to select a pivot element and to place it at the correct position.

(3) To compute the correct position of the pivot element, it rearranges the array `A[start ... end]` into two subarrays `A[start ... p-1]` and `A[p+1 ... end]` such that each element of `A[start ... p-1]` is less than or equal to the pivot element and each element of `A[p+1 ... end]` is greater than the pivot element. The subarray can be empty.
Here, `p` indicates the position of the pivot element, and `A[start ... end]` indicates a subarray consisting of the elements from index `start` to index `end` of array `A`.

(4) The algorithm can be described as follows:

Step 1 During the initialization step, determine the starting and ending indices `start` and `end` that represent a subarray.

Step 2 While the index `end` satisfies `A[pivot] < A[end]`, move the index `end` leftward by 1.

Step 3 While the index `start` satisfies `A[pivot] ≥ A[start]`, move the index `start` rightward by 1.

Step 4 While the index `end` is greater than the index `start`, repeat Step 2 and Step 3.

Step 5 Swap the values at the index `pivot` and the index `start`. `A[start]` becomes a new pivot element and returns it.

(5) This program also calls the following functions:

The function `swap(A[], i, j)` that swaps the values `A[i]` and `A[j]`.

The function `print()` that displays the input parameters.

(6) The indices of all arrays start at 1.

[Program]

```
FUNCTION: listByID(INT: S[], INT: H[], INT: N) {
    INT: i

    quickSort(S, H, 1, N);
    print("The result is: ");
    FOR (i ← 1; i < N; i ← i + 1) {
        print(S[i], ", ");
    }
    print(S[N]);
}

FUNCTION: quickSort(INT: S[], INT: H[], INT: start, INT: end) {
    INT: pivot

    IF (start < end) {
        pivot ← partition(S, H, start, end);  /* α */
        quickSort(S, H, start, pivot - 1);
        quickSort(S, H, pivot + 1, end);
    }
}
```

```
FUNCTION: partition(INT: S[], INT: H[], INT: start, INT: end) {
    INT: pivot ← start

    WHILE (start < end) {
        WHILE ((start < end) and ([   A   ])) {
            end ← end - 1;
        }
        WHILE ((start < end) and ([   B   ])) {
            start ← start + 1;
        }
        IF (start < end) {
            swap(H, [   C   ]);
            swap(S, [   C   ]);
        }
    }
    swap(H, [   D   ]);
    swap(S, [   D   ]);
    return start;
}

FUNCTION: swap(INT: A[], INT: i, INT: j) {
    INT: tmp ← A[i]

    A[i] ← A[j];
    A[j] ← tmp;
}
```

Subquestion 1

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

Answer group for A and B

a) `H[pivot] < H[end]`

b) `H[pivot] < H[start]`

c) `H[pivot] ≥ H[end]`

d) `H[pivot] ≥ H[start]`

e) `S[pivot] < S[end]`

f) `S[pivot] < S[start]`

g) `S[pivot] ≥ S[end]`

h) `S[pivot] ≥ S[start]`

Answer group for C and D

a) `pivot + 1, end`

b) `pivot - 1, end`

c) `pivot - 1, pivot + 1`

d) `start, end`

e) `start, pivot`

f) `start, pivot + 1`

g) `start, pivot - 1`

Subquestion 2

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

```
N ← 6;
S[] ← {1484, 5586, 9023, 2789, 7265, 1596};
H[] ← {174, 182, 163, 152, 174, 152};
```

Figure 1 Test case

When the program `listByID(S, H, N)` is executed for the test case shown in Figure 1, the statement pointed by α will be executed 3 times. Variable `pivot` is set to $\boxed{\text{E}}$ on the first execution, set to 4 on the second execution, and set to 2 on the third execution. The output is the following:

`The result is:` $\boxed{\text{F}}$

Answer group for E

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

Answer group for F

a) `1596, 2789, 9023, 1484, 7265, 5586`
b) `1596, 2789, 9023, 7265, 1484, 5586`
c) `2789, 1596, 9023, 1484, 7265, 5586`
d) `2789, 1596, 9023, 7265, 1484, 5586`

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

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

[Program Description]

This program converts an expression specified in commonly used infix notation into an expression in postfix notation where the operator comes after its operands. For example, expression “(3+2)*5” in infix notation is converted to expression “32+5*” in postfix notation.

The expression accepted by the program is defined as follows:

(1) It consists of operands and operators, and it may be inside parentheses. Space characters are not allowed in an expression.
(2) The operand is a one-digit number (0, 1, …, 9) or another expression.
(3) The operator is one of ‘+’, ‘-’, ‘*’ or ‘/’, and they take two operands.
(4) The operators ‘*’ and ‘/’ have higher priority than operators ‘+’ and ‘-’. An expression surrounded by parentheses has the highest priority. For example, the priority for the operators in “3+2*(5-7)” is ‘-’, ‘*’ and ‘+’ in descending order.
(5) Operators at the same priority associate from left to right. For example, “3+2-5” is equivalent to “(3+2)-5”, but not “3+(2-5)”.

Assume that the input expression has no syntax errors and it satisfies the above definitions.

The conversion algorithm is as follows:

(1) Scan the input expression from left to right.
(2) If the current character is a numeral, output it as the last character of the postfix notation.
(3) If the current character is ‘(’, push it to the stack.
(4) If the current character is ‘)’, pop a character from the stack and output it as the last character of postfix notation until ‘(’ is popped. The pair of parentheses is then discarded.
(5) If the current character is one of the operators, pop all operators from the stack which have a priority higher than or equal to that of the current operator while the stack is not empty and the top of the stack is not ‘(’. Then, push the current operator to the stack.
(6) Repeat (2) to (5) until the input expression is fully scanned.
(7) Pop a character from the stack and output it as the last character of the postfix notation while the stack is not empty.

For example, if the input expression is “`(3+2)*5`”, the steps proceed as follows:

Step	Current character	Operation	Stack	Output
1	(	push ‘(’	$\boxed{(}$	
2	3	add 3 to the output	$\boxed{(}$	3
3	+	push ‘+’	$\boxed{(}\boxed{+}$	3
4	2	add 2 to the output	$\boxed{(}\boxed{+}$	32
5	)	pop ‘+’ and add it to the output	$\boxed{(}$	32+
		pop ‘(’	(empty)	32+
6	*	push ‘*’	$\boxed{*}$	32+
7	5	add 5 to the output	$\boxed{*}$	32+5
8		pop ‘*’	(empty)	32+5*

The program has two global variables referred to as `stack` and `top`. `stack` is an array of characters that holds characters stacked from the input expression, whereas `top` is an integer variable that indicates the current index of `stack` (or `-1` when it is empty).

The program has five user defined functions:

(1) `void push(char ch)`:
This function takes a character `ch` as input. It increases the value of `top` by 1. Then it inserts `ch` at the `top` index of `stack`.

(2) `char pop(void)`:
This function returns the character at the `top` index of the `stack` and reduces the value of `top` by 1.

(3) `int empty(void)`:
This function returns 1 if `stack` holds no character; 0 otherwise.

(4) `int priority (char ch)`:
This function takes a character `ch` (an operator) as input and returns the priority level of that operator.

(5) `void infixToPostfix(char exp[], char result[])`:
This function takes two arrays `exp` and `result` as input. `exp` contains the input expression. The function converts `exp` into postfix notation and stores it in `result`. The string lengths of the input expression and result are smaller than the array lengths of `exp` and `result` respectively.

[Program]

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

char stack[100];
int top = -1;

void push(char ch) {
    [    A    ];
    [    B    ];
}

char pop(void) {
    char ch = stack[top];
    [    C    ];
    return ch;
}

int empty(void) {
    if (top >= 0) {
        return 0;
    }
    else {
        return 1;
    }
}

int priority(char ch) {
    if (ch == '+' || ch == '-') {
        return 1;
    }
    else {
        return 2;
    }
}
```

```
void infixToPostfix(char exp[], char result[]) {
    int i, j = 0;
    for (i = 0; exp[i] != '\0'; i++) {
        if (isdigit(exp[i])) {
            result[j] = exp[i];
            j++;
        }
        else if (exp[i] == '(') {
            [   D   ];
        }
        else if (exp[i] == ')') {
            while (stack[top] != '(') {
                [   E   ];
                j++;
            }
            pop();          // ← α
        }
        else {
            while (!empty() && stack[top] != '(' && [   F   ]) {
                result[j] = pop();
                j++;
            }
            push(exp[i]);   // ← β
        }
    }
    while (!empty()) {
        result[j] = pop();
        j++;
    }
    result[j] = '\0';
}

int main() {
    char exp[100], result[100];
    scanf("%99s", exp);
    infixToPostfix(exp, result);
    printf("%s\n", result);
    return 0;
}
```

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 through C

a) `ch = stack[top]` b) `stack[top] = ch` c) `top = -1`
d) `top = 1` e) `top++` f) `top--`

Answer group for D

a) `push(')')` b) `push(exp[i])` c) `push(i)`
d) `push(j)` e) `push(result[j])`

Answer group for E

a) `pop()` b) `push(exp[i])` c) `push(result[j])`
d) `result[j] = exp[i]` e) `result[j] = pop()`

Answer group for F

a) `priority(stack[top]) < priority(exp[i])`
b) `priority(stack[top]) <= priority(exp[i])`
c) `priority(stack[top]) > priority(exp[i])`
d) `priority(stack[top]) >= priority(exp[i])`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description. Here, the answers to be inserted in G1 and G2 should be selected as the correct combination from the answer group for G.

When the program is executed by providing the following input expression, the line pointed out by `// ← α` is executed $\boxed{\text{G1}}$ time(s), and the line pointed out by `// ← β` is executed $\boxed{\text{G2}}$ times.

Input expression: `4*3-(5*2+1)`

Answer group for G

	G1	G2
a)	1	3
b)	1	4
c)	2	3
d)	2	4

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

This program is a proof-of-concept prototype for selecting televisions (TVs) that match user preferences, such as screen sizes. The program provides filters each of which performs selection of TVs from specified TVs. The program consists of the following interface and classes.

(1) The `Tv` class represents a TV. It contains the following attributes.

(i) `modelId`: the model id of this TV

(ii) `resolution`: the resolution of this TV, e.g., 4K, 8K

(iii) `screenSize`: the screen size (in inches) of this TV, e.g., 45, 55

(iv) `price`: the price (in dollars) of this TV, e.g., 599, 999

(2) The `Filter` interface is a functional interface that represents a filter used to select TVs that match conditions given by its implementation. The `applyFilter` method returns a `Set` of selected `Tv`s from the specified `Set` of `Tv`s. Note that the returned `Set` may be immutable.

(3) The `RangeFilter` class implements the `Filter` interface for selecting TVs that fit in the value range of the attribute specified by its constructor. The value range is given by the `lo` and `hi` parameters, and the attribute value is obtained using the `getter` parameter. The attribute data type must be `int` or `Integer`. The type of the `getter` parameter is `java.util.function.Function<T, R>`. Its `apply` method takes the parameter of type `T` and returns its resulting value of type `R`.

(4) The `EqualityFilter` class implements the `Filter` interface for selecting TVs with the same attribute value as the `value` of this instance. The attribute and its value are specified by the two parameters, `value` and `getter`, of the constructor.

(5) The `Test` class tests the above `Filter` implementations based on the following user preferences.

(i) resolution: 4K or 1080p

(ii) screen size: 45 to 55 inches

(iii) budget: up to $750

The following output is produced when the `main` method of the `Test` class is executed:

```
Model ID=SX-4567, Resolution=4K, Screen size=45-inch, Price=$599
Model ID=TY-987F, Resolution=1080p, Screen size=50-inch, Price=$299
```

[Program 1]

```
import java.util.Objects;

class Tv {
   private final String modelId;
   private final String resolution;
   private final int screenSize;
   private final int price;

   Tv(String modelId, String resolution, int screenSize, int price) {
      this.modelId = modelId;
      this.resolution = resolution;
      this.screenSize = screenSize;
      this.price = price;
   }

   String getModelId() { return modelId; }

   String getResolution() { return resolution; }

   int getScreenSize() { return screenSize; }

   int getPrice() { return price; }

   @Override
   public boolean equals(Object o) {
      if (this == o) return true;
      if (!(o instanceof Tv)) return false;
      Tv tv = (Tv) o;
      return screenSize == tv.screenSize && price == tv.price &&
        modelId.equals(tv.modelId) &&
        resolution.equals(tv.resolution);
   }

   @Override
   public int hashCode() {
      return Objects.hash(modelId, resolution, screenSize, price);
   }

   @Override
   public String toString() {
      return String.format(
         "Model ID=%s, Resolution=%s, Screen size=%d-inch, Price=$%d",
         modelId, resolution, screenSize, price);
   }
}
```

[Program 2]

```
import java.util.Set;

interface Filter {
    Set<Tv> applyFilter(Set<Tv> tvs);
}
```

[Program 3]

```
import java.util.Set;
import java.util.function.Function;
import java.util.stream.Collectors;

class RangeFilter implements Filter {
   private final int lo, hi;
   private final Function<Tv, Integer> getter;

   RangeFilter(int lo, int hi, Function<Tv, Integer> getter) {
      this.lo = lo;
      this.hi = hi;
      this.getter = getter;
   }

   @Override
   public Set<Tv> applyFilter(Set<Tv> tvs) {
      return tvs.stream()
         // filter() returns a stream of Tvs that match the condition.
         .filter(tv -> lo <= getter.apply(tv) && getter.apply(tv) <= hi)
         // The following collects filtered Tv objects into a Set.
         .collect(Collectors.toSet());
   }
}
```

[Program 4]

```
import java.util.Set;
import java.util.function.Function;
import java.util.stream.Collectors;

class EqualityFilter<T> implements Filter {
   private final T value;
   private final Function<Tv, T> getter;
   EqualityFilter([  A  ] value, Function<Tv, [  A  ]> getter) {
      this.value = value;
      this.getter = getter;
   }

   @Override
   public Set<Tv> applyFilter(Set<Tv> tvs) {
      return tvs.stream().filter(tv -> [  B  ])
         .collect(Collectors.toSet());
   }
}
```

[Program 5]

```
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
import java.util.stream.Collectors;
import java.util.stream.Stream;

public class Test {
   static final Set<Tv> TVS = new HashSet<>(Arrays.asList(
         new Tv("TY-987F", "1080p", 50, 299),
         new Tv("SX-4567", "4K", 45, 599),
         new Tv("SX-5678", "4K", 50, 759),
         new Tv("SX-6789", "4K", 65, 999),
         new Tv("SZ-8765", "8K", 45, 749)
   ));

   public static void main(String[] args) {
      Set<Tv> set4k = new EqualityFilter<>("4K", Tv::getResolution).
            applyFilter(TVS);
      Set<Tv> setfhd = new EqualityFilter<>("1080p", Tv::getResolution)
            .applyFilter(TVS);
      Filter range1 = new RangeFilter(45, 55, Tv::getScreenSize);
      // Stream.concat() merges two Streams into a single Stream.
      Set<Tv> sized = Stream.concat(range1.applyFilter(set4k).stream(),
            range1.applyFilter(setfhd).stream())
            .collect(Collectors.toSet());
      Filter range2 = new RangeFilter(0, 750, Tv::getPrice);
      range2.applyFilter(sized).forEach(System.out::println);
   }
}
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in Program 4.

Answer group for A

a) `int` b) `Integer` c) `R`

d) `String` e) `T`

Answer group for B

a) `getter(tv).equals(value)` b) `getter.apply(tv).equals(value)`

c) `getter.apply(value).equals(tv)` d) `this.apply(tv).equals(value)`

e) `tv.apply(this).equals(value)`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank □ in Program 6 and Program 7.

To simplify the usage of the `Filter` implementations, the prototype introduced the following `Filter` implementations.

(1) The `AndFilter` class implements the `Filter` interface to apply multiple `Filter` implementations specified by the constructor parameter. The `applyFilter` method returns a `Set` of `Tv`s that have passed all of the `Filter`s.

(2) The `OrFilter` class implements the `Filter` interface to apply multiple `Filter` implementations specified by the constructor parameter. The `applyFilter` method returns a `Set` of `Tv`s that have passed any of the `Filter`s.

[Program 6]

```
import java.util.HashSet;
import java.util.Set;
import java.util.stream.Stream;

class AndFilter implements Filter {
   private final Filter[] filters;

   AndFilter(Filter... filters) {
      this.filters = filters.clone();
   }
   @Override
   public Set<Tv> applyFilter(Set<Tv> tvs) {
      Set<Tv> filteredTvs =    C   ;
      // Removes TVs filtered out by each Filter.
      Stream.of(filters) // of() returns a stream of filters.
        // map() returns a stream consisting of the results of calling
        // applyFilter() of the elements of this stream.
        .map(f -> f.applyFilter(filteredTvs))
        // forEach() performs an action for each element of this stream.
        .forEach(filteredTvs::retainAll);
      return filteredTvs;
   }
}
```

[Program 7]

```
import java.util.HashSet;
import java.util.Set;
import java.util.stream.Stream;

class OrFilter implements Filter {
   private final Filter[] filters;

   OrFilter(Filter... filters) {
      this.filters = filters.clone();
   }

   @Override
   public Set<Tv> applyFilter(Set<Tv> tvs) {
      Set<Tv> filteredTvs =    D   ;
      // Adds only TVs not filtered out by each Filter
      Stream.of(filters).map(f -> f.applyFilter(tvs))
            .forEach(filteredTvs::addAll);
      return filteredTvs;
   }
}
```

Answer group for C and D

a) `new HashSet<>()`

b) `new HashSet<>(Arrays.asList(filters))`

c) `new HashSet<>(filters)`

d) `new HashSet<>(tvs)`

e) `null`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank $\boxed{}$ in the following description. It is assumed that the correct answers are inserted in all of the blanks $\boxed{\text{A}}$ to $\boxed{\text{D}}$.

The Test2 class is another test program for the AndFilter and OrFilter implementations. Its main method is an equivalent implementation of the main method of the Test class. The resulting outputs of the two main methods are the same set of TVs. Note that the iteration order of TVs produced by the methods may vary, depending on the Set implementations.

When executing the main method of the Test2 class, the getResolution, getScreenSize, and getPrice methods of the Tv class are called $\boxed{\text{E}}$, $\boxed{\text{F}}$, and $\boxed{\text{G}}$ times, respectively.

[Program 8]

```
public class Test2 {
   public static void main(String... args) {
      Filter filter = new AndFilter(
            new OrFilter(new EqualityFilter<>("4K", Tv::getResolution),
                  new EqualityFilter<>("1080p", Tv::getResolution)),
            new RangeFilter(45, 55, Tv::getScreenSize),
            new RangeFilter(0, 750, Tv::getPrice));
      filter.applyFilter(Test.TVS).forEach(System.out::println);
   }
}
```

Answer group for E, F and G

a) 2	b) 3	c) 4
d) 5	e) 6	f) 8
g) 10	h) 12	