March 2018

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

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

Instructions:

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

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

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

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

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

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

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

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

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

   [Sample Answer]

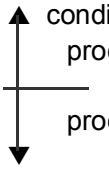
Sample	ⓐ	●	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ

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

Notations used for pseudo-language

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

[Declaration, comment, and process]

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

[Logical constants]

`true`, `false`

(continued on next page)

[Operators and their priorities]

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

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

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

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

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

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

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

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

The salt serves three purposes:

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

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

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

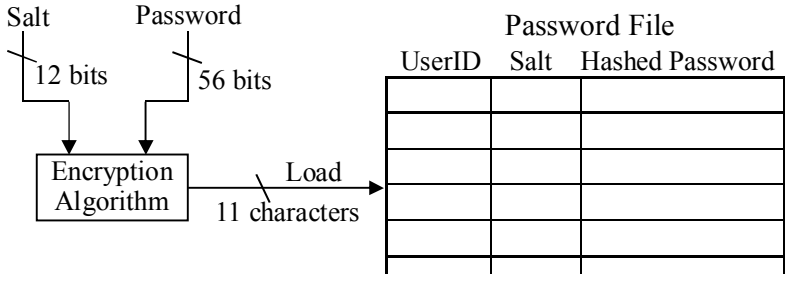


Figure 1 Procedure for loading a new password

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

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

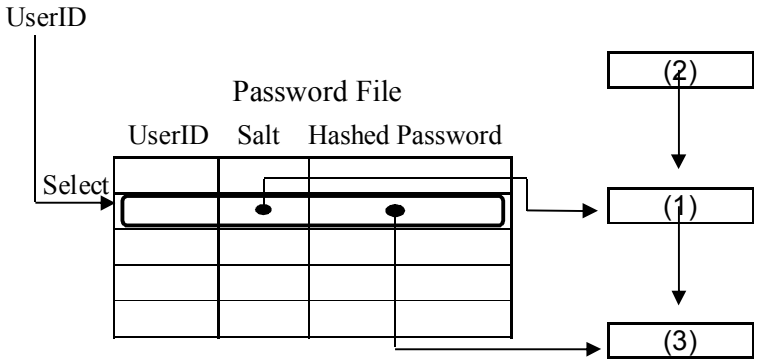


Figure 2 Verifying a password

Subquestion 1

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

Answer group for A

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

Answer group for B

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

Answer group for C

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

Subquestion 2

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

Answer group

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

Subquestion 3

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

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

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

Answer group

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

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

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

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

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

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

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

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

Answer group for B and C

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

Answer group for D

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

Subquestion 2

From the answer group below, select the correct answer to be inserted in the blank [E] in Figure 1.

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

The steps of the method are as follows:

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

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

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

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

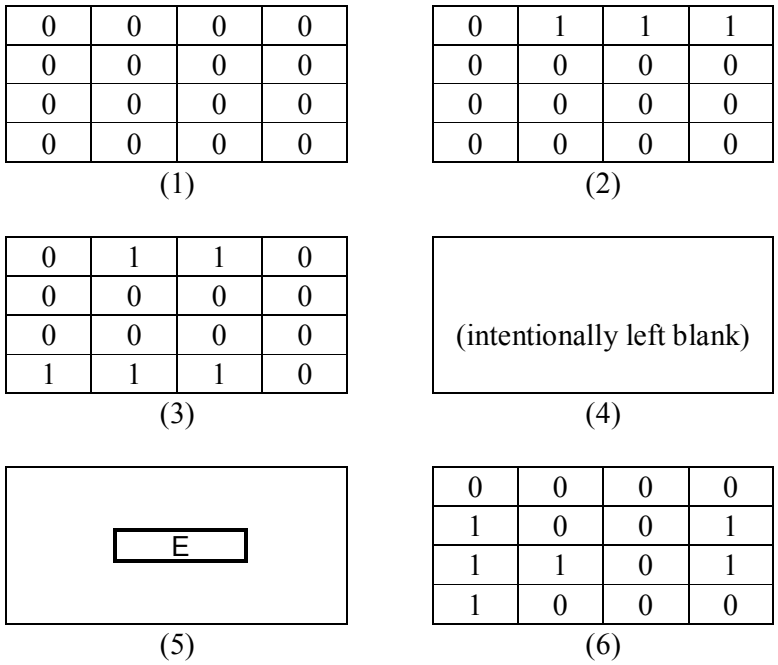


Figure 1 Implementation of LRU using an $n \times n$ matrix

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

Answer group for E

a)

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

b)

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

c)

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

d)

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

Subquestion 3

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

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

Answer group for F

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

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

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

TblMktPlan (PlanID, MktActivity, Department)

TblBgtUsed (BgtID, PlanID, StartDate, EndDate, BgtUsed, Manager)

Note: Underline indicates the primary key

Subquestion 1

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

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

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

The sample output of SQL1 is as follows:

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

Note:

"INNER JOIN" returns all matching rows from both tables that satisfies the condition

"LEFT JOIN" returns all rows from the left table with matching rows in the right table

"RIGHT JOIN" returns all rows from the right table with matching rows in the left table

Answer group for A and B

a) `INNER JOIN`

b) `LEFT JOIN`

c) `RIGHT JOIN`

d) `Mkt.PlanID = Bgt.PlanID`

e) `Mkt.PlanID = Bgt.PlanID GROUP BY Mkt.Department`

f) `Mkt.PlanID = Bgt.PlanID GROUP BY Mkt.PlanID`

Subquestion 2

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

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

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

TblSaleType (STypeID, SaleType)

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

TblDayType (DTypeID, DayType)

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

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

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

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

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

Answer group for C

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

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

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

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

Subquestion 3

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

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

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

The sample output of SQL3 is as follows:

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

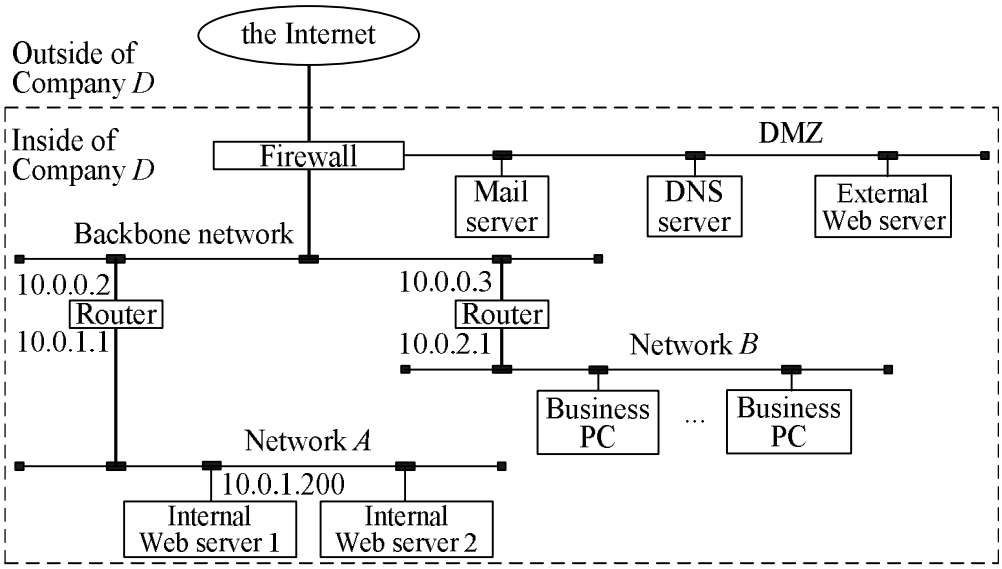
Answer group for D and E

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

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

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

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



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

Figure 1 Current network configuration of Company *D*

Subquestion 1

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

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

[IP address]

10.0.0.2	10.0.0.3	10.0.0.4	10.0.1.1	10.0.1.2
10.0.1.3	10.0.2.1	10.0.2.2	10.0.2.3	10.0.2.4

Answer group for A

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

Answer group for B

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

Subquestion 2

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

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

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

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

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

Answer group for C and D

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

Answer group for E

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

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

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

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

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

Figure 1 shows the use case diagram of the system.

Subquestion 1

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

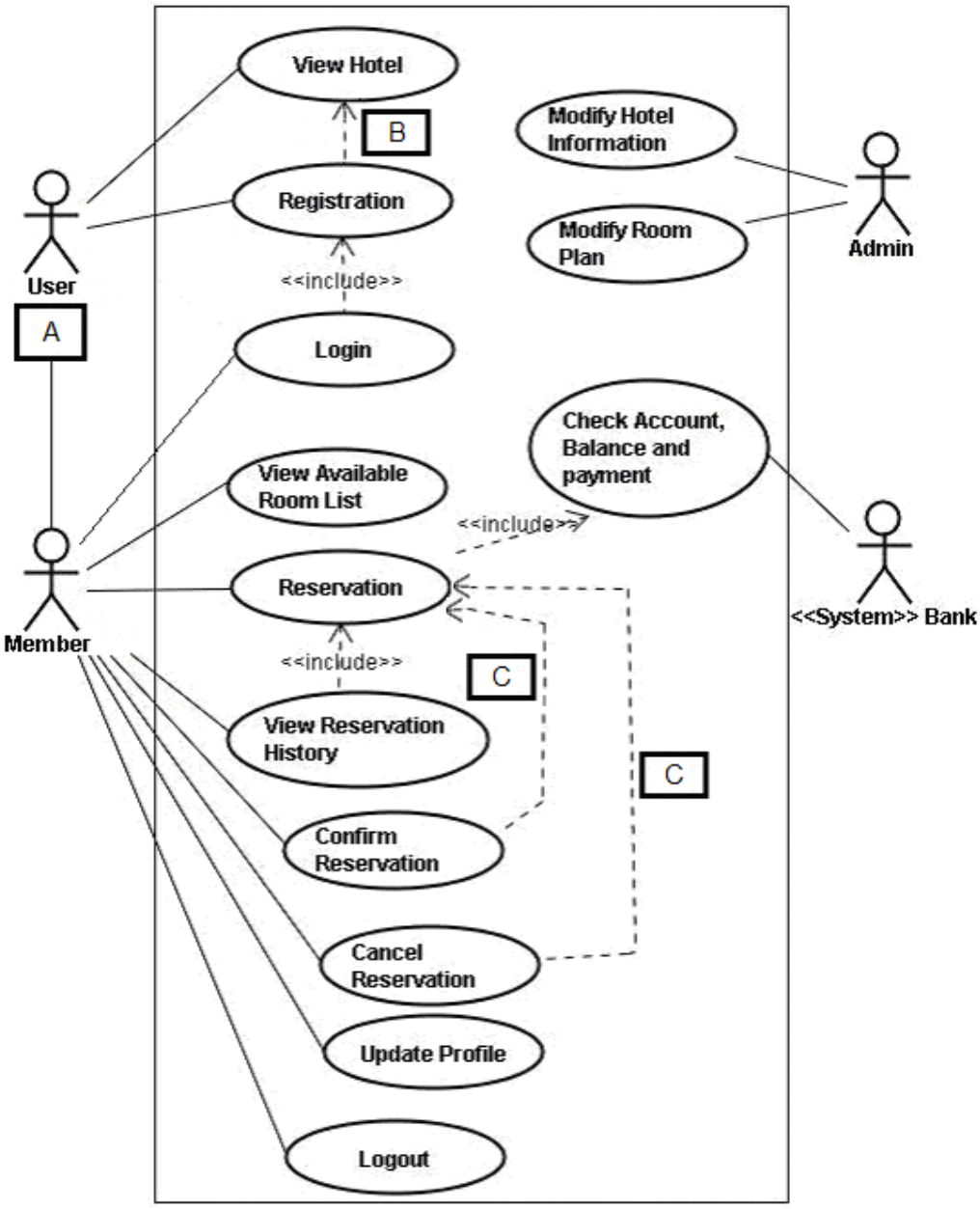


Figure 1 Use case diagram of the system

Answer group for A

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

Answer group for B and C

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

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank [ ] in Figure 2.

Figure 2 shows the state transition diagram of the system.

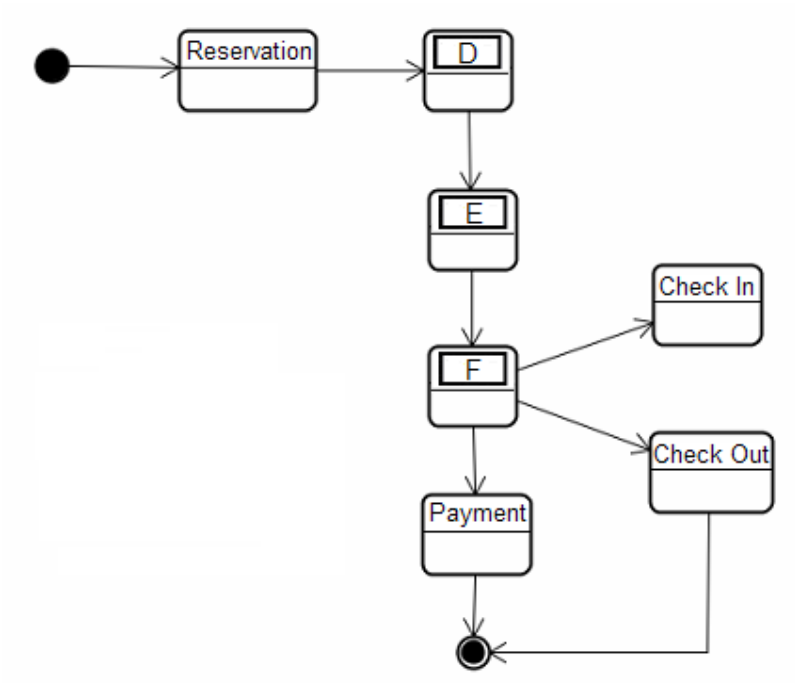


Figure 2 State transition diagram of the system

Answer group

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

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

[Program Description 1]

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

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

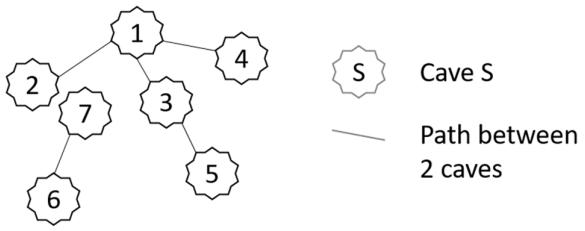


Figure 1 Example of cave map

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

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

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

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

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

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

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

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

Table 2 Argument specification for the subprogram BuildTree

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

Table 3 Argument specification for the subprogram MakeEdge

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

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

[Program 1]

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

  ■ I: 1, N, 1
  |  • T[I] ← -1
  ■

  ■ I: 1, K, 1
  |  • MakeEdge(T, A[I], B[I])
  ■

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

  • T[a] ← b
```

Subquestion 1

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

Assume that arrays `A[]` and `B[]` contain the values shown in Table 4, and `T[]` is array of integers with 12 elements.

After the execution of `BuildTree(12, 10, A[], B[], T[])`, `T[]` should be set as follows:

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

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

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

Answer group for A and B

a) 0 b) 1 c) 2

d) 3 e) 4 f) 5

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank [ ] in Program 2.

[Program Description 2]

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

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

Table 5 Argument specification for the program `GuideNext`

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

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

[Program 2]

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

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

Answer group for C through E

a) `-1`	b) `0`	c) `T[d]`	d) `T[s]`
e) `T[dd]`	f) `da`	g) `dd`	h) `sa`

Subquestion 3

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

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

Answer group for F through H

a) 0	b) 1	c) 2	d) 3
e) 4	f) 5	g) 6	h) 7
i) 8	j) 9		

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

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

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

(1) Given an integer 124, it is composed of 3 digits that are 1, 2, and 4. Comparing each pair of the digits from left to right, 1 and 2, the value increases from 1 to 2. When comparing 2 and 4, the value is also increasing. Therefore, the integer 124 is a non-decreasing number.

(2) Given an integer 221, when comparing 2 and 2, they are the same and are non-decreasing. However, when comparing 2 and 1, the value is decreasing. Therefore, the integer 221 is not a non-decreasing number.

(3) 1-digit numbers are regarded as non-decreasing numbers.

[Program Description]

(1) The program reads two integer numbers, `num1` and `num2`, from the standard input. Here, `num1` and `num2` satisfy the following three conditions: (i) 0 < `num1`, (ii) `num1` ≤ `num2`, and (iii) `num2` < 32768.

(2) The program tests all the integer numbers that lie between `num1` and `num2`.

(3) The program outputs the total count of non-decreasing numbers that lie between `num1` and `num2`. This count is stored in the variable `ndCount`.

(4) An example of the output of this program is as follows:

```
Enter two integers: 18 23
Count of non-decreasing numbers between 18 and 23: 4
```

(5) The following two user-defined functions are used in the program:

(i) `int isNonDecreasingNumber(char *s)`

The function returns 1 if the input number in the string `*s` is a non-decreasing number, otherwise, returns 0.

(ii) `void convert(int num, char *s)`

The function converts the integer number `num` into the character string `*s`. For example, an integer 135 is converted into "135".

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

```
size_t strlen(const char *s)
```

The function returns the length of the string `*s` (not including the terminating '\0').

[Program]

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

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

int isNonDecreasingNumber(char *s) {
   int i;

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

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

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

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

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

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

Subquestion

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

Answer group for A and B

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

Answer group for C and D

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

Answer group for E

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

Answer group for F

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

Answer group for G

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

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

[Program Description]

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

Subquestion 1

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

(1) The `countCharacters` method returns the number of digits present in a specified number `n`. If the specified number is negative, it returns an extra character count for the minus sign. For example,
   (i) if `n` = 7, it returns 1,
   (ii) if `n` = -7, it returns 2,
   (iii) if `n` = 789, it returns 3.

(2) The `convert` method returns a `String` array representing the digits in words present in a specified number `n`. If the specified number is negative, the minus sign is represented as `"minus"`. For example,
   (i) if `n` = 7, it returns an array consisting of a word: `["seven"]`
   (ii) if `n` = -7, it returns an array consisting of the words: `["minus", "seven"]`
   (iii) if `n` = 79, it returns an array consisting of the words: `["seven", "nine"]`

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

[Program 1]

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

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

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

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

       return inWords;
    }
}
```

Answer group for A

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

Answer group for B

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

Subquestion 2

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

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

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

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

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

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

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

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

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

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

[Program 2]

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

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

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

```
}


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

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

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

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

abstract String nextWord();

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

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

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

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

   public boolean hasNext() {
```

```
            return [      E      ];
        }

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

Answer group for C and E

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

Answer group for D and F

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

Subquestion 3

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

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

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

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

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

[Program 3]

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

Answer group for G

a) `left side -> eight five seven minus`
b) `left side -> minus eight five seven`
c) `left side -> seven five eight minus`
d) `right side -> minus eight five seven`
e) `right side -> minus seven five eight`
f) `right side -> seven five eight minus`