April 2022

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1	Q2 – Q5	Q6	Q7, Q8
Question Selection	Compulsory	Select 2 of 4	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)			

Instructions:

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

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

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

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

   (3) **Question Selection**
   For questions **Q2** through **Q5**, and **Q7** and **Q8**, mark the ⓢ of the questions you select to answer in the "Selection Column" on your answer sheet.

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

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

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

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

   [Sample Answer]

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

Notations used in the pseudo-language

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

[Declaration, comment, and process]

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

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

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

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

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

[Logical constants]

true, false

[Operators and their precedence]

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

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

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

Q1. Read the following description of risk assessment of information assets, and then answer Subquestions 1 through 4.

Company R is a medium-sized system integrator (SI) vendor with 100 employees. Prior to beginning a project, Company R conducts risk assessment of the information assets to be used in the project according to the internal procedures, as shown in Figure 1. The company recently decided to conduct a risk assessment for the newly launched Project P.

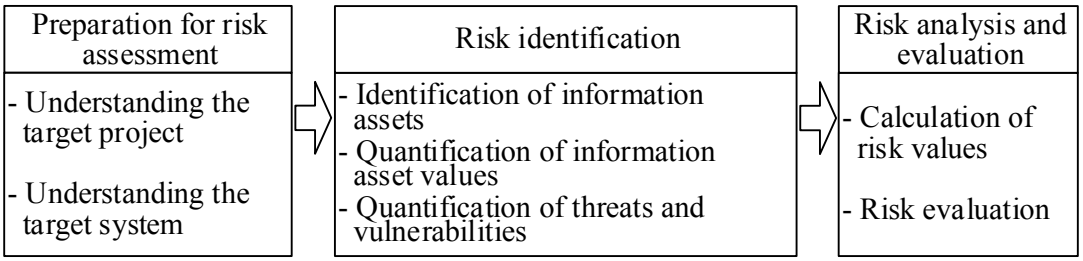


Figure 1. Company R's risk assessment procedures

[Description of Project P (excerpt)]

(1) The project involves developing a purchasing system to be used by the customer.
(2) The test data used during development will be provided by the customer.
(3) Project members will receive the customer's USB storage devices (hereinafter, USB storage) containing the customer test data from the customer, take the USB storage to Company R, copy the customer test data to the development server, and then delete the data from the USB storage.
(4) Travel from Company R to the customer's office takes 90 minutes by train.
(5) The programs will be developed on development PCs, and uploaded to the development server as appropriate.

[Company R's development environment (excerpt)]

(1) The development server and development PCs are used in program development.
(2) The development server is located in a locked server room.
(3) The development server is placed on controlled access, and can be accessed only by project members and system administrators.
(4) Development PCs are loaned from the systems department to project members at the start of a project, and are returned when the project is completed.

[Company R's development standards (excerpt)]

(1) During development, project members download the necessary portions of customers test data from the development server to their own development PCs, and delete the data when no longer needed.

(2) After completion of a project, the project manager deletes the customer test data from the development server and confirms that the customer test data has been deleted from all development PCs.

[Company R's risk value calculation method]

Company R calculates the risk value of each information asset using the following formula.

Risk value = Value of information asset × Threat × Vulnerability

Here, "value of information asset" refers to the degree of impact if the information asset is damaged. The degree of impact is evaluated considering three perspectives: Confidentiality (C) , Integrity (I) , and Availability (A) , each given a value from 1 to 3. For "threat", the degree of possibility of occurrence is evaluated with a value from 1 to 3. For "vulnerability", the degree to which damage manifests when a threat occurs is evaluated with a value from 1 to 3. For the values 1 to 3, 3 is large and 1 is small.

If the risk values calculated for C, I, and A are all 12 or less, the risk is accepted; otherwise, additional measures are implemented to counter the risk.

[Risk identification]

(1) Identification of information assets

Company R identifies the information assets handled in Project P. The results are shown in Table 1.

Table 1. Results of identification of information assets

No.	Information asset	Creation or acquisition	Storage location	Disposal
		□		
3	In-development programs on the development server	Uploaded to the development server by project members	Development server	Delete when project is completed
4	Customer test data	Received by customer's USB storage, copied to the development server	Customer's USB storage, development server, and development PCs	
		□		

Note: Shaded part is not shown. "□" indicates omission.

(2) Quantification of information asset values

For each information asset in Table 1, the evaluated values of C, I, and A, and the reasoning behind the evaluation are shown in Table 2.

Table 2. Values of information assets and reasoning behind evaluation

No.	Information asset	C	I	A	Reasoning behind evaluation of values
					□
3	In-development programs on the development server	3	3	3	(i) There will be an impact on the progress of the project if the in-development programs are unusable (ii) Customers' trust would be lost if the information were leaked out of the company (iii) If program versions are not controlled, there will be an impact on the progress of the project owing to inconsistency
4	Customer test data	3	2	1	
					□

Note: Shaded part is not shown. "□" indicates omission.

(3) Quantification of threats

Table 3 shows the contents and values of threats related to the information asset no. 4 (customer test data), which is one of the information assets in Table 2.

Table 3. Contents and values of threats related to information asset no. 4

No.	Threat ID	Threat content	Value
4	T1	Customer's USB storage containing the customer test data is lost on the way to Company R	3
	T2	Development server is accessed without authorization from outside the company, and the customer test data is stolen	1
	T3	Customer test data are destroyed or leaked due to viral infection	2
	T4	Customer test data is leaked from the customer's USB storage after the data is copied to the development server	3
	T5	Customer test data that are no longer needed are leaked from development PCs after the completion of testing	2
	T6	Customer test data are copied to a portable storage from the development server by a project member or system administrator and are taken out of the server room	1
	T7	Customer test data are destroyed on the development server	1

(4) Quantification of vulnerabilities to threats

Table 4 shows the measures to mitigate the vulnerability and the value of vulnerability for each threat in Table 3. The value of vulnerability is 1 if two or more measures have been taken by systems, rules, or operations; the value is 2 if only one measure has been taken, and it is 3 if no measures have been taken.

Table 4. Measures to mitigate vulnerabilities and values of vulnerabilities for the threats in Table 3

Threat ID	Vulner-ability ID	Measures to mitigate vulnerability	Value
T1	V1	• Having customer encrypt the customer test data when it is saved to the customer's USB storage	2
T2	V2	• No measures against the vulnerability	3
T3	V3	• Installing anti-virus software on the development server and development PCs, and automatically updating virus definition files • Performing virus scan on the customer's USB storage before copying the customer test data to the development server	1
T4	V4	• Project manager confirming that the customer test data have been deleted from the customer's USB storage	2
T5	V5	• Project manager confirming that the customer test data have been deleted from the development PCs	2
T6	V6	• Performing physical access controls using employee ID badges • Installing monitoring cameras in the server room	1
T7	V7	• No measures against the vulnerability	3

[Risk analysis and evaluation]

The risk analysis and evaluation are conducted with respect to the information asset no. 4 (customer test data) based on Tables 2 to 4. Table 5 shows the calculated risk values.

Table 5. Risk values of information asset no.4

No.	Values of information asset			Threat		Vulnerability		Risk values			
	C	I	A	Threat ID	Value	Vulnerability ID	Value	Risk value ID	C	I	A
4	3	2	1	T1	3	V1	2	R1	18	12	6
				T2	1	V2	3	R2	9	6	3
				T3	2	V3	1	R3			
				T4	3	V4	2	R4			
				T5	2	V5	2	R5		A	
				T6	1	V6	1	R6			B
				T7	1	V7	3	R7			

Note: Shaded parts are not shown.

The project manager of Project P creates an action plan for the risks based on the results of the risk analysis and evaluation. After this, the company takes measures against the risks.

Subquestion 1

From the answer group below, select the correct answer to be inserted in each blank [ ] in Table 5.

Answer group for A and B

a) 1 b) 3 c) 6 d) 8
e) 9 f) 12 g) 18

Subquestion 2

In table 2, (ii) and (iii) are the reasoning behind the evaluation when the value of information assets has been evaluated in terms of either C, I, or A. From the answer group below, select the appropriate combination of C, I, and A that corresponds to (ii) and (iii).

Answer group

	(ii)	(iii)
a)	A	C
b)	A	I
c)	C	A
d)	C	I
e)	I	A
f)	I	C

Subquestion 3

From the answer group below, select the correct number of threats for which additional measures against the risks are required owing to the risk analysis and evaluation related to the information asset no. 4 (customer test data).

Answer group:

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

Subquestion 4

From the answer group below, select the correct answer to be inserted in each blank [ ] in the following description related to the occurrence of, and response to, a security incident in Company R.

The systems department received a report that the customer test data for Project P were stored on a development PC used in Project Q, which was launched after the completion of Project P. Investigations revealed that the PC was used as a development PC in Project P, and after being returned to the systems department, it was loaned by the systems department to members of Project Q. To address the risk of leakage of customer test data, Company R decided to add the following two measures:

- o [C]
- o [D]

Answer group for C and D

a) A development server will be newly prepared for each project.
b) A user will enter the download date, deletion date, and person in charge in an administration log when downloading / deleting customer test data to / from development PCs for use.
c) A warning message will be displayed when customer test data are saved to development PCs.
d) Project members will be given read-only access to customer test data on the development server.
e) The systems department will add processes to completely erase all data from returned development PCs.
f) The systems department will regularly check the development server's access logs.

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

Q2. Read the following description of a language processing system, and then answer Subquestions 1 through 3.

A language processing system is a software that reads a source program written in a programming language (hereinafter, a language), analyzes the source program according to the syntax and semantics of the language, and converts the source program to an object program in suitable format for execution, or simply executes it by itself.

An interpreter is a kind of language processing system that reads, analyzes, and executes a source program at a time. A source program developed for an interpreter is executable on any computer where the interpreter is executable. However, this may be a disadvantage of the interpreter method because a source program must be analyzed each time before its execution. On typical interpreters, deep and complex analysis extends the total execution time of a source program. Most interpreters run a source program sequentially as it is written.

A compiler is another kind of language processing system that reads and analyzes a source program, and outputs an object program that consists of machine-readable instructions. The object program is executable on the computer, which equips the specific instruction set with the compiler execution results at different times. This may be an advantage of the compiler method when compared with the interpreter method, because it [A].

Subquestion 1

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

Answer group for A

a) allows programmers to check the cause of an error during execution
b) allows programmers to modify the source program and run it immediately
c) can execute the source program while checking it interactively
d) can optimize the executable program for the target computer

Subquestion 2

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

There is another method that uses a computer virtually constructed by software, called a virtual machine. As with the compiler method, the language processing system in this method outputs an executable program; however, it consists of instructions dedicated for use by the virtual machine, called intermediate code. Figure 1 shows an example of the conversion of a source program into intermediate code and its execution with the virtual machine.

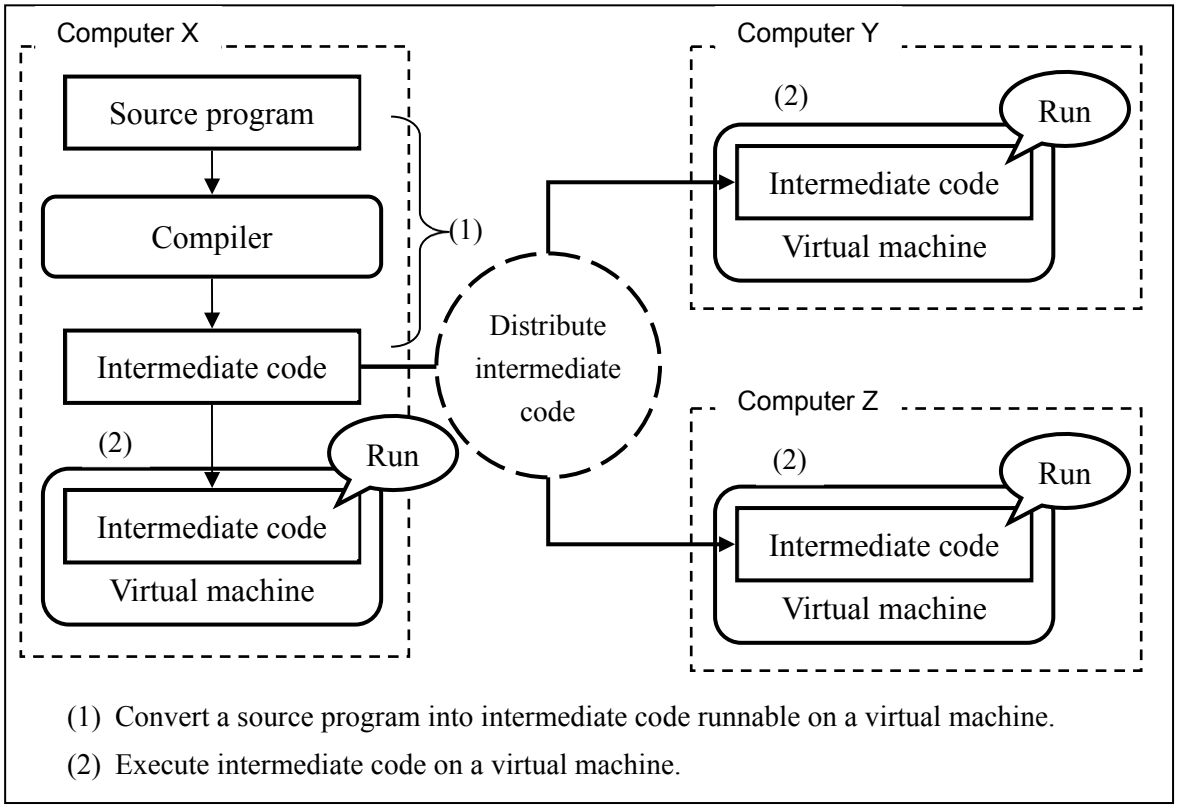


Figure 1 Example of running a program on a virtual machine

By implementing a virtual machine for various operating systems (OSs) or hardware, it is possible to distribute a program that [B].

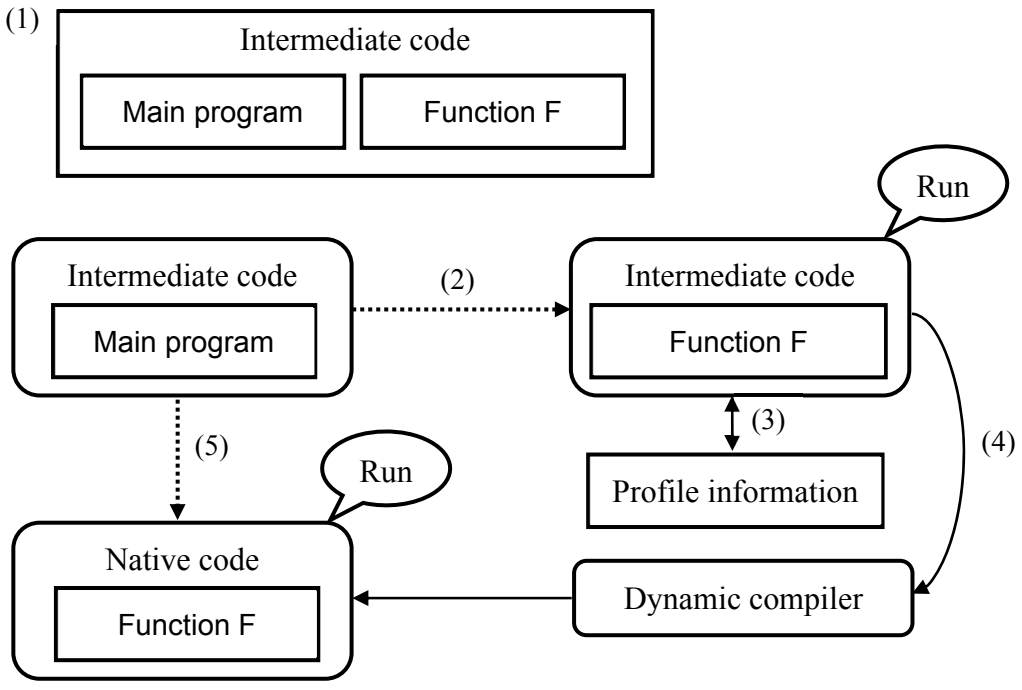
Answer group for B

a) is independent of any specific OS or hardware
b) depends on a specific OS and hardware
c) maximizes the performance of specific hardware
d) reduces memory usage on certain hardware

Subquestion 3

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

Some virtual machines collect information about what processes or functions are frequently executed (hereinafter referred to as profile information) and feeds that information back to the intermediate-code execution. Dynamic compilation is a method employed to compile a part of an intermediate code to a native code for subsequent execution based on the profile information. This method is used in practical language processing systems such as the Java JIT (Just-In-Time) compiler. Figure 2 shows an example of the method.



(1) The main program and function F are distributed in the form of intermediate code.
(2) At first, function F is called in the form of intermediate code.
(3) The decision whether to compile function F dynamically or not is made based on the profile information.
(4) If function F is determined to be compiled, the dynamic compiler starts compilation.
(5) After the compilation, function F is called in the form of the native code.

Figure 2 Process flow showing application of the dynamic compilation method

When adapting a specific system to dynamic compilation, it is necessary to consider certain overheads, which are the startup time and compilation time of the dynamic compiler, in addition to the execution time of the program.

Consider the total execution time when function F shown in Figure 2 is executed with and without the dynamic compilation method under the following conditions:

(1) The total execution time of function F consists of the following times only.
   - Execution time of the intermediate-code instructions
   - Execution time of the native-code instructions
   - Startup time of the dynamic compiler
   - Compilation time of the dynamic compiler

(2) Function F before the dynamic compilation consists of 400 intermediate-code instructions. Because function F contains a loop structure, 2,000 intermediate-code instructions are executed once it is called.

(3) Function F after the dynamic compilation also consists of 400 native-code instructions. In addition, 2,000 native-code instructions are executed once it is called.

(4) An intermediate-code instruction takes 0.5 μs (microseconds) to execute whereas a native-code instruction takes 0.01 μs.

(5) The dynamic compiler starts compiling function F at the beginning of its 101st call. It takes 100 ms (milliseconds) to start up the dynamic compiler and 100 ms to compile 1,000 intermediate-code instructions to native-code instructions of the same number. The execution of function F of intermediate-code stops during the dynamic compilation and resumes function F of native-code after the compilation is finished.

When function F is called from the main program 200 times,

- total execution time of function F for the first 100 calls: 100 ms
- execution time for the dynamic compilation: [C] ms
- total execution time of function F for the last 100 calls: [D] ms

The dynamic compilation consumes a considerable amount of time. Therefore, there is a tradeoff with respect to the time at which the dynamic compilation is started.

Consequently, invocation of the dynamic compiler before the first call of function F reduces the total execution time when function F is called more than [E] times.

Answer group for C and D

a) 1	b) 2	c) 10	d) 20
e) 40	f) 100	g) 140	

Answer group for E

a) 40	b) 102	c) 142	d) 175

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

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

Company U has operated five large-scale shopping centers in the country since 2010. The customer support division of Company U issues various credit and loan cards for local customers, such as credit cards valid worldwide, credit cards valid only in the shopping centers, and prepaid discount cards valid only for restaurants in the shopping centers.
When a customer wants to receive a card, he/she comes to one of the shopping centers, and submits an application for account registration. A customer can make an application for multiple types of cards. During the first account registration, customer registration is also carried out. Therefore, each registered customer has at least one account.
For each account, an employee from the customer support division is assigned.
Company U uses a relational database to manage shopping center operations. The relational database is composed of the following tables:

Branch table

BranchID	BranchName	BranchAddress

Employee table

EmpID	EmpName	Title	AssignedBranchID	SuperiorEmpID

Customer table

CustID	CustName	CustAddress

Account table

AcctID	CustID	OpenDate	BranchID	EmpID	LoanBalance

Transaction table

TransID	TransDate	TransType	Amount	AcctID

Note: Underline indicates the primary key and dotted-line indicates the foreign key.

Subquestion 1

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

An entity relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships among those entities. The elements of an ERD are as follows: Entities, Relationships and Attributes.

Figure 1 shows the ERD, and Table 1 shows the list of cardinality of the relationship.

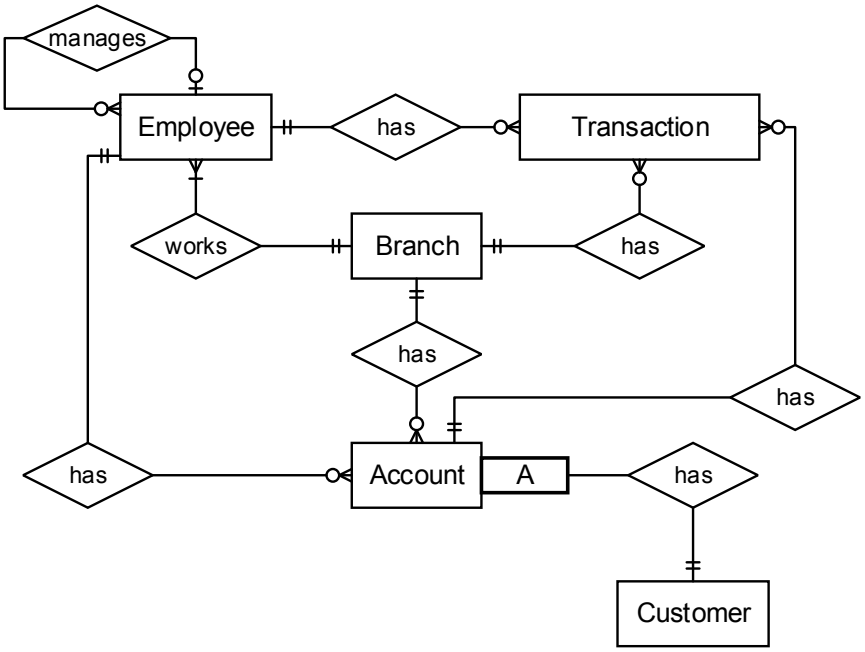


Figure 1 Entity Relationship Diagram

Table 1 List of cardinality of relationship

Termination point	Cardinality	Existence	Description
⊣⊢	One	Mandatory	Must exist, one and only one
+O–	One	Optional	May exist, one, or none
⊱⊣	Many	Mandatory	Must exist, one or more
⊱O–	Many	Optional	May exist, one or more or none

Answer group for A

a) ⊣⊢ b) +O– c) ⊱⊣ d) ⊱O–

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank [ ] in the SQL statement SQL1.

The following SQL statement SQL1 outputs employees with the names of their superiors. If an employee has no superior, it displays "(none)" in the SuperiorName field.

Note: The function `ISNULL(`*expression*`,` *replacement*`)` returns the value of *replacement* when the value of *expression* is NULL.
LEFT OUTER JOIN returns all records from the left table, and the matched records from the right table. The result is NULL from the right side, if there is no match.

```
-- SQL1 --
SELECT E.EmpID, E.EmpName, E.Title,
       ISNULL(S.EmpName, '(none)') AS SuperiorName
FROM Employee E
LEFT OUTER JOIN [ B ] S
ON [ C ]
```

When the SQL statement SQL1 is executed with the following sample data:

Employee table

EmpID	EmpName	Title	AssignedBranchID	SuperiorEmpID
1001	Alice	Receptionist	20	1003
1002	Bob	Loan manager	10	NULL
1003	Charlie	Loan examiner	10	1002
1004	Don	Receptionist	30	1003

From the sample data shown above, SQL1 outputs the following result:

EmpID	EmpName	Title	SuperiorName
1001	Alice	Receptionist	Charlie
1002	Bob	Loan manager	(none)
1003	Charlie	Loan examiner	Bob
1004	Don	Receptionist	Charlie

Answer group for B and C

a) `Account`
b) `Customer`
c) `E.EmpID = S.EmpID`
d) `E.EmpID = S.SuperiorEmpID`
e) `Employee`
f) `S.EmpID = E.SuperiorEmpID`
g) `S.SuperiorEmpID IS NOT NULL`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank [ ] in the SQL statement SQL2.

Mr. C, an IT engineer trainee, is creating an SQL statement SQL2. The purpose of SQL2 is to output account IDs that were registered before 2020 and never used in 2020 and 2021. It is assumed that the Trans2Years table contains all transactions in 2020 and 2021 extracted from the Transaction table.

To test the program, Mr. C provides the following sample data:

Account table

AcctID	CustID	OpenDate	BranchID	EmpID	LoanBalance
50001	3001	2016-08-26	20	1001	2534
50002	3002	2017-04-27	30	1004	0
50003	3001	2018-01-02	20	1001	2117
50004	3002	2019-09-19	30	1004	0
50005	3001	2020-04-16	20	1001	679
50006	3002	2021-01-10	30	1004	0

Trans2Years table

TransID	TransDate	TransType	Amount	AcctID
700001	2020-01-24	Restaurant	59	50003
700002	2020-04-21	Cashing	230	50005
700003	2020-05-09	Revolving	781	50001
700004	2021-01-19	Cashing	640	50001
700005	2021-06-07	Restaurant	62	50005
700006	2021-09-04	Revolving	862	50003

From the sample data shown above, SQL2 should output the following result:

AcctID
50002
50004

The SQL statement SQL2 is as follows. At the moment, SQL2 is incomplete and contains error(s).

Note: The function `YEAR(`*date*`)` returns the four-digit year of the *date* as a number.

Line No.

```
1    -- SQL2 --
2    SELECT A.AcctID
3    FROM Account A
4    WHERE EXISTS(SELECT *
5                 FROM Trans2Years T
6                 WHERE A.AcctID = T.AcctID)
7          AND YEAR(OpenDate) < 2020
```

If SQL2 shown above is executed, it outputs [D].

To output the correct result, SQL2 should be changed. Concretely, [E].

Answer group for D

a) 2 rows of account IDs: 50001 and 50003
b) 3 rows of account IDs: 50001, 50003 and 50005
c) 3 rows of account IDs: 50002, 50004 and 50006
d) 4 rows of account IDs: 50001, 50001, 50003 and 50003
e) 4 rows of account IDs: 50002, 50002, 50004 and 50004

Answer group for E

a) change "`Account A`" to "`Trans2Years T`" on line 3, and
   change "`Trans2Years T`" to "`Account A`" on line 5
b) change "`EXISTS`" to "`IN`" on line 4
c) change "`EXISTS`" to "`IN`" on line 4,
   change "`Account A`" to "`Trans2Years T`" on line 3, and
   change "`Trans2Years T`" to "`Account A`" on line 5
d) change "`EXISTS`" to "`NOT EXISTS`" on line 4
e) change "`EXISTS`" to "`NOT EXISTS`" on line 4,
   change "`Account A`" to "`Trans2Years T`" on line 3, and
   change "`Trans2Years T`" to "`Account A`" on line 5

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

Q4. Read the following description of the establishment and termination process of TCP connections, and then answer Subquestions 1 and 2.

TCP (Transmission Control Protocol) is a transport layer protocol in the TCP/IP suite. It provides reliable data transmission using a connection-oriented mechanism to transport data for various application layer protocols over an IP (Internet Protocol) network.

[TCP connection establishment process]

The connection must be established initially before the actual data transmission can take place. Figure 1 shows the TCP connection establishment process, which is called a three-way handshake.

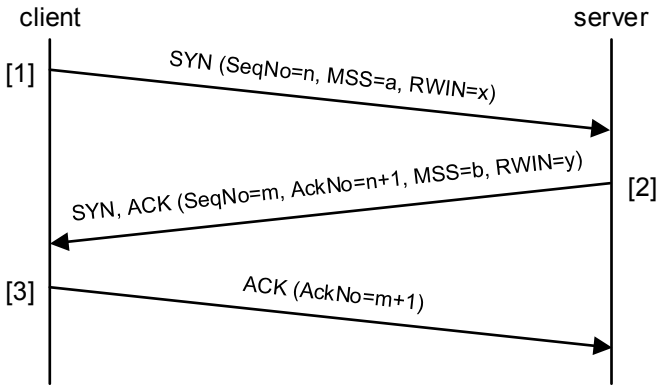


Figure 1 TCP connection establishment process

[1] First, the client establishes a connection by sending to the server a packet with the SYN flag set to 1. This packet contains the initial sequence number, which is a random number n set at the client side in the sequence number (SeqNo) field of the packet. It also supplies a maximum segment size (MSS) value and a TCP receive window (RWIN) value. MSS is the TCP option that specifies the largest segment size that the client will accept. RWIN is the size of the buffer for the incoming data.

[2] Then, the server replies with a packet with both SYN and ACK flags set to 1. This packet includes the server generated arbitrary sequence number m and the acknowledgment number (AckNo) set to n + 1 to follow up the sequence number in the first SYN request. The server also supplies its own MSS and RWIN values to advertise its capacity.

[3] Finally, the client replies with another packet with the ACK flag set to 1. Here, the value of AckNo is set to m + 1 in accordance with the server's sequence number.

At this point, three-way handshake is complete and the connection is established. The MSS that both parties can handle is known by both parties, and a smaller number will be used to ensure that both sides are able to handle it. Both parties also know each other's receiving buffer capacity (RWIN) and will be able to send data within the limitation. Although the party establishing the connection here is designated as the client, both parties can equally send and receive data in subsequent transmissions until the connection is terminated. The four-way handshake TCP connection termination process is shown later.

[Sequence number and Acknowledgment number]

TCP uses sequence numbers and acknowledgment numbers to ensure reliable communication. Figure 2 shows how the server sends a total of 30 bytes of data to the client, divided into 10 and 20 bytes.

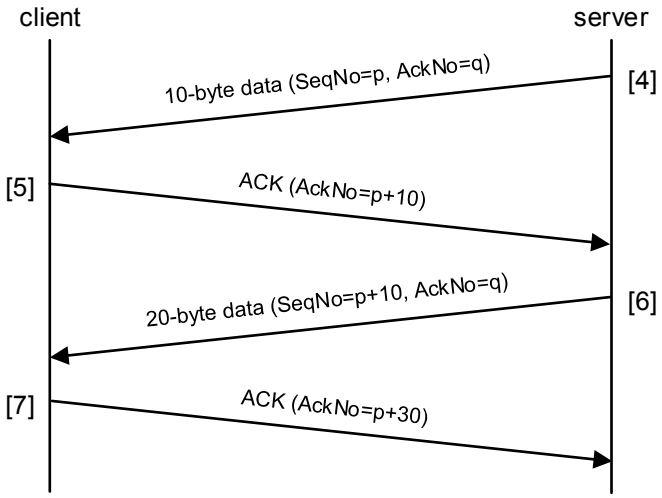


Figure 2 Sequence numbers and Acknowledgment numbers

[4] The server sends 10 bytes of data to the client. The sequence number p indicates the current sequence of the data, which started from the initial sequence number n in step [1] and increased continuously with each data packet previously sent during the connection. The acknowledgment number q is also the value response to the latest sequence number from the client in the same fashion.

[5] The client acknowledges the data by sending a packet with the ACK flag set along with the acknowledgment number p + 10. It also indicates that the client expects p + 10 as the sequence number of the next packet from the server.

[6] The server sends 20 bytes of data to the client. The sequence number is p + 10. The acknowledgment number is maintained as q because the server has not received any data from the client.

[7] The client acknowledges the data again by sending a packet with the ACK flag set along with the acknowledgment number p + 30.

Packets with data consume sequence numbers according to their data length, as described earlier. Note that packets with the SYN or FIN flag also consume sequence number by 1 as seen in steps [1] through [3] and steps [8] through [11]; however, packets with only an ACK flag do not consume sequence number.

[TCP connection termination process]

Figure 3 shows the termination process of a TCP connection.

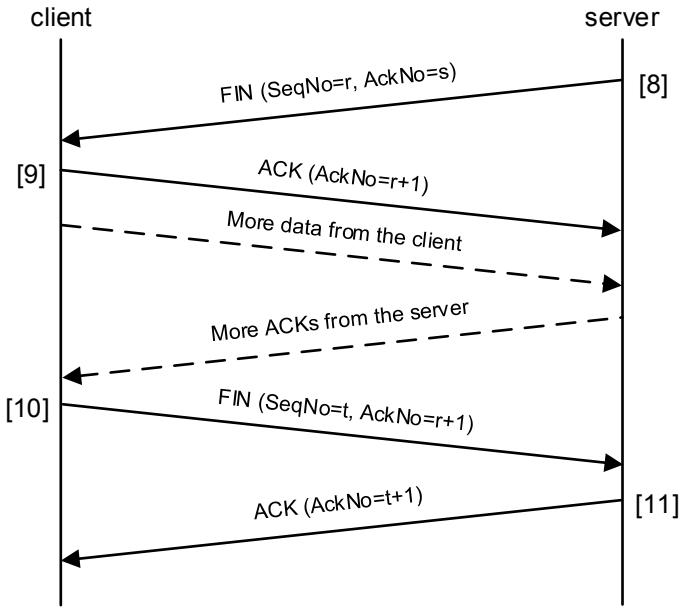


Figure 3 TCP connection termination process

Although it is illustrated here that the termination request is initiated from the server side, either side can send a packet with the FIN flag set to 1 to request a connection termination if it has no more data to send. Figure 3 indicates that the client acknowledges the termination request from the server side. However, if the client has more data to send, the communication continues as represented by the dashed lines.

The overall process can be described as follows:

[8] The server sends a FIN flagged packet to notify the client that it wishes to terminate the connection. The sequence number is r and the acknowledgment number is s.

[9] The client acknowledges the termination request by sending a packet with the ACK flag set along with the acknowledgment number r + 1. At this point, the connection is half closed and the server will not be able to send data. However, the client can continue to send data segments as needed, and the server will respond with the corresponding acknowledgment numbers accordingly.

[10] Once the client finishes sending all its data, it also sends back to the server a packet with the FIN flag set along with its current sequence number t and the acknowledgment number r + 1. Note that the acknowledgment number remains the same. This is because there are no more data segments from the server side because it is already half closed.

[11] To finalize the termination process, the server sends back to the client the final ACK packet with the acknowledgment number t + 1.

To illustrate the working of these processes, consider a client connecting to a Daytime server to obtain the current date and time value.

(1) The client initiates the connection from its TCP port number 55133 to the server by sending a SYN packet with SeqNo = 20, MSS = 1460, and RWIN = 65535.

(2) The server responds with a SYN/ACK packet with SeqNo = 10, AckNo = 21, MSS = 1240, and RWIN = 8192.

(3) The client completes the three-way handshake process by sending an ACK packet with AckNo = 11.

(4) At this point, the TCP connection is established and both parties know each other's capabilities, and will be able to perform data transmission accordingly. The segment size of the subsequent transmission is agreed upon and set to [A] bytes.

(5) The server sends back to the client a date and time value which is 22 bytes long along with SeqNo =11.

(6) After the client receives the data, it responds with an ACK packet that has an acknowledgment number equals to the server's SeqNo plus the size of the data, which increases the AckNo to 33.

(7) In this case, the server starts the connection termination process once it receives the acknowledgment from the client. Thus, a FIN packet is sent from the server with SeqNo = [B] and AckNo = 21.

(8) The client responds with a/an [C] packet with AckNo = 34. Then, because the daytime operation is completed, the client also sends a/an [D] packet to the server with SeqNo = [E] and AckNo = 34.

(9) The server completes the four-way handshake termination process by sending an ACK packet with AckNo = 22.

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) 1240 b) 1460 c) 8192
d) 55133 e) 65535

Answer group for B and E

a) 11 b) 12 c) 21
d) 22 e) 33 f) 34

Answer group for C and D

a) ACK b) FIN c) SYN d) SYN/ACK

Subquestion 2

During the connection establishment process, it is possible that malicious clients may keep sending SYN packets to the server without supplying ACK packets to complete the three-way handshake. Without the acknowledgment from the client, these requests will remain in the incoming connection queue of the server until it times out. This is called SYN flooding and it may prevent the server from responding to legitimate SYN requests.
From the answer group below, select the appropriate characteristic or the cause of concern regarding SYN flooding.

Answer group

a) During a SYN flooding attack, the server under attack is unable to initiate an outgoing TCP connection to another host because its outgoing connection queue is already overwhelmed with large numbers of incoming SYN requests.
b) It is possible to reduce the effects of SYN flooding by increasing the timeout value that controls how long the system should wait for an ACK as this method helps to improve the stability of legitimate clients with slower links.
c) Owing to the connection-oriented nature of TCP, SYN flooding will potentially prevent new clients from connecting to the server rather than disconnecting already established connections.
d) To stop an ongoing SYN flooding attack, the server's administrator must restore the system from a backup because the system may be damaged owing to the overwhelming number of unsuccessful three-way handshake attempts.

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

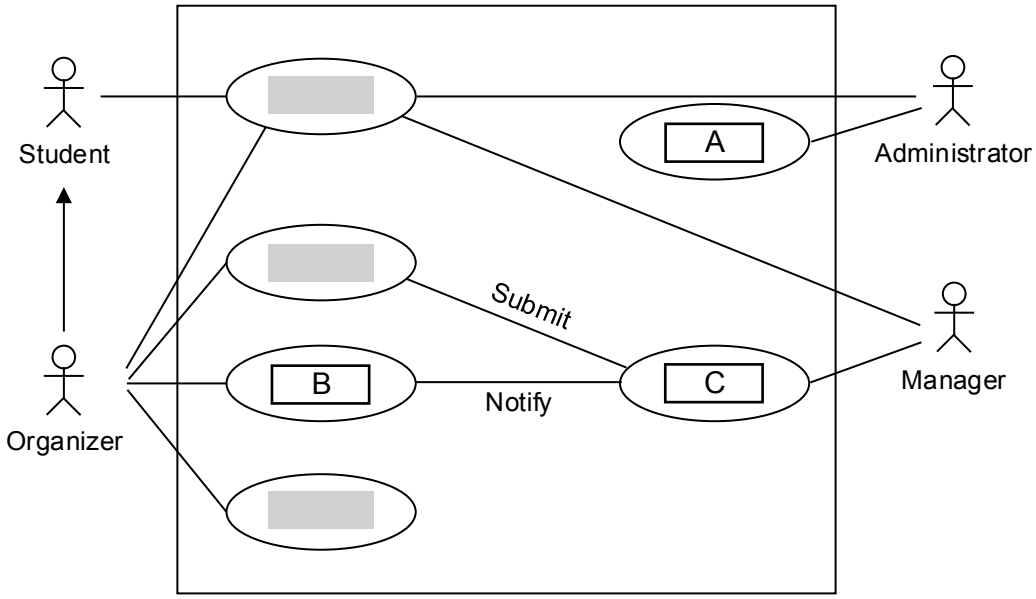
Q5. Read the following description of an information system in a university, and then answer Subquestions 1 through 3.

Student Event Calendar (hereinafter, SEC) is a system that brings students closer to events and extra-curricular activities taking place at University W. The system provides information about all the events taking place at University W in an intuitive form that allows students to track events quickly. With SEC, students have opportunities to write comments and provide feedback on events so that the university can collect data and improve the quality of future events.

The system has four types of users:

- Students: Students who are studying in University W and who desire to join the events in University W to improve their general skills or to expand their human network.
- Organizers: A subset of the students who are leaders of some clubs in University W. They can represent their clubs and make decisions regarding the organization of some events. An Organizer creates event information, updates it, and submits it to a Manager. After the event ends, the Organizer reports the execution result of the event.
- Managers: Staff members of the personal development department of University W.
  Managers review event information and either approve or reject it. Approved event information is public and open to the students. Rejected event information is sent to the Organizer who can update it and resubmit it to the Manager.
  Managers can keep track of all event information in order to gain insight about the activities associated with all events during a given period.
- Administrators: Staff members of the personal development department of University W.
  Administrators manage users on the SEC system. When a student becomes the leader of a club, the Administrator grants the Organizer the authority to organize events.

Figure 1 shows the use case diagram of the SEC system that focuses on the main functions. Table 1 shows the description of use cases that are used in Figure 1. Some use cases are used for multiple types of users, and others are used for only a specific type of user.



Note: Shaded parts are not shown.

Figure 1 Use case diagram of SEC system

Table 1 Use case description

Use case	Description
Create information	Creates information about an event.
Report result	After the event, the Organizer reports results, issues, etc.
Review information	Reviews the event information.
Set authority for the Organizer	Sets the authority to organize events.
Update information	Updates the event information.
View events	Views detailed information of the events in University W.

Subquestion 1

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

Answer group for A through C

a) Create information
b) Report result
c) Review information
d) Set authority for the Organizer
e) Update information
f) View events

Subquestion 2

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

The process of organizing an event is as follows:

(1) An Organizer creates event information by filling out the form with event information that contains the organization plan and inputs the form into the SEC system.

(2) The SEC system verifies the information systematically by checking the database for reasonableness in terms of the budget and availability of equipment to determine whether the required equipment is needed by other similar events that overlap with the time period. After the information is verified, the SEC system sends the information to a specific Manager.

(3) The Manager reviews the event information and decides to approve or reject the event. If the Manager approves the event information, the event is published. If the event is rejected, comments and recommendations on the organization plan are sent back to the Organizer so that the Organizer can identify the problems in the organization plan. After that, the Organizer updates the event information and resubmits it.

(4) The Organizer can modify the information whenever the need arises even after it has been published. The updated information must be reviewed by the Manager.

(5) After the event is over, the Organizer reports the result of the event to the Manager.

Figure 2 shows the activity diagram of publicizing an event. It starts from the Organizer when he/she begins to create an event. The SEC system is responsible for managing the database and controlling the communication between the Organizer and Manager.

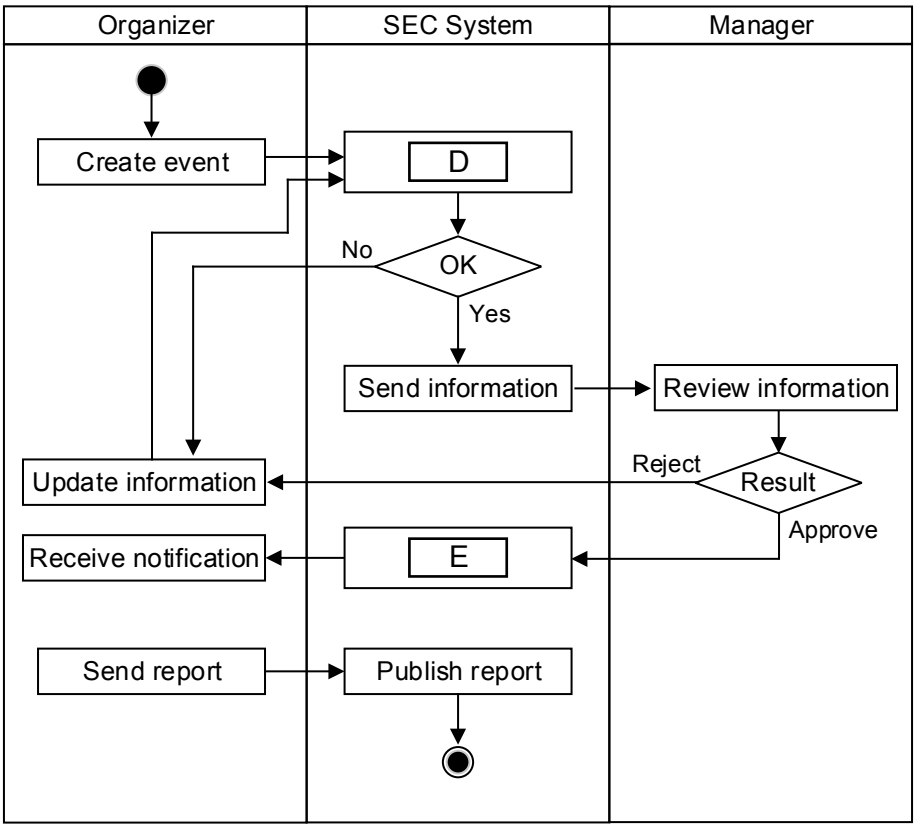


Figure 2 Activity diagram of publicizing an event

Answer group for D and E

a) Publish information
b) Remove information
c) Review information into the database
d) Send information to Organizer
e) Update information
f) Verify information

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank [ ] in Figure 3. Here, the answers to be inserted in F1 and F2 should be selected as the correct combination from the answer group for F.

Figure 3 shows the event information state chart from the event planning to the event end. In Figure 3, a rectangular indicates a state and an arrow indicates a state transition.

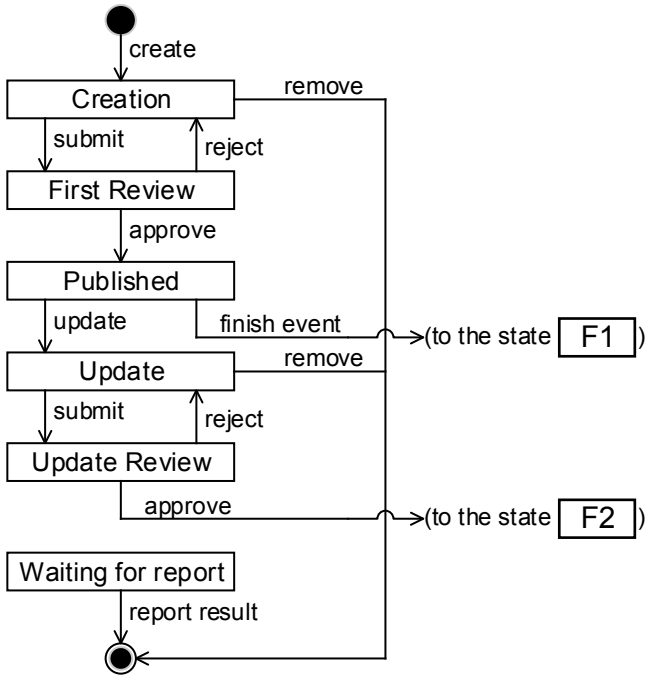


Figure 3 Event information state chart

Answer group for F

	F1	F2
a)	Waiting for report	Published
b)	Waiting for report	Update
c)	Waiting for report	Waiting for report
d)	◉	Published
e)	◉	Update
f)	◉	Waiting for report

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

Q6. Read the following description of bit processing programs, and then answer Subquestions 1 and 2.

The function `BitTest` tests the bit values of the bits in an 8-bit data selected by a mask, and the function `BitCount` counts the number of bits whose values are 1's in an 8-bit data.

In this question, the following additional pseudo-language notations are used:

- The operators "`&`" and "`|`" obtain the logical product and logical sum, respectively, for each pair of bits in the corresponding bit positions of two 8-bit binary-type data, and obtain an 8-bit binary-type result.
- The notation `'....'B` such as `'1'B` and `'11111110'B` expresses a binary-type constant.
- The selection processes (`IF` / `ELSE`) are coded without using curly braces `{ }` in accordance with the language specifications of C and Java.

Subquestion 1

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

[Description of the function `BitTest`]

The function `BitTest` tests the bit values of the bits in an 8-bit data selected by a mask and returns the result.

The function `BitTest` is declared as follows:

```
FUNCTION: BitTest(8-bit Binary: Data, 8-bit Binary: Mask)
```

The argument `Data` contains an 8-bit data to be tested, and the argument `Mask` contains an 8-bit mask. The bits of the `Mask` are made to have a one-to-one correspondence with the bits of the `Data`. A mask bit of 1 indicates that the corresponding bit in `Data` is to be tested. When a mask bit is 0, the corresponding bit in `Data` is ignored. Here, in `Mask`, there is at least one bit whose value is 1.

The function `BitTest` returns the following return value:

Return value 0: All tested bits are 0's
1: Tested bits mixed 0's and 1's
2: All tested bits are 1's

For example, in Example 1 in Figure 1, the 2 bits (bit numbers 5 and 1) of Mask are 1's; thus, the values of the 2 bits (bit numbers 5 and 1) of Data are tested. Because 0 and 1 are both included in these, the return value is 1. In Example 2, the 2 bits (bit numbers 6 and 2) of Mask are 1's; thus, the values of the 2 bits (bit numbers 6 and 2) of Data are tested. Because both are 1's, the return value is 2.

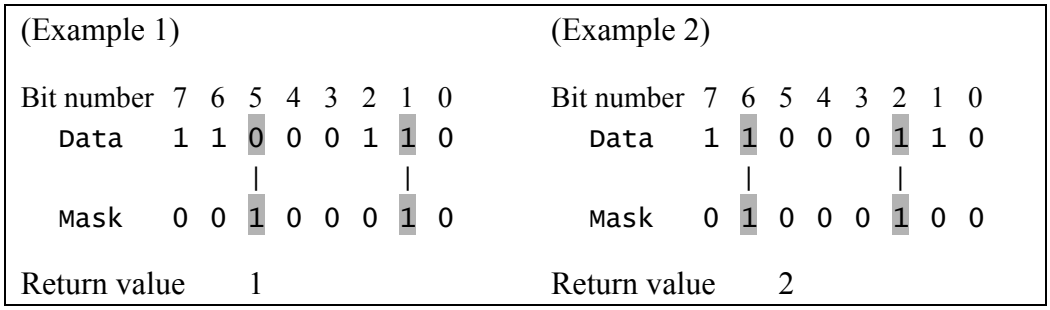


Figure 1 Examples of the BitTest operation

[Program 1]

```
FUNCTION: BitTest(8-bit Binary: Data, 8-bit Binary: Mask) {
   INT: Rv              /* return value */

   IF ( A )
      Rv ← 2;
   ELSE
      IF ( B )
         Rv ← 0;
      ELSE
         Rv ← 1;
   return Rv;           /* returns Rv as the return value */
}
```

Based on the definition of the return value, the function BitTest [C] when Mask has only one bit whose value is 1, and the function BitTest [D] when the contents of Data and Mask are identical.

Program 1 presumes that Mask contains at least one bit whose value is 1.
To improve the usability, the specification of Program 1 is to be changed so that it can accept the case where all the bits in Mask have value 0. In this case, Program 1 returns 0 as the return value. To achieve this, the following change proposals, (1) to (3), are made. Note that change proposal (1) makes no change to Program 1. Here, the correct answers are assumed to be inserted in blanks [A] and [B].

Change proposal (1) (no change)

```
IF ( A )
   Rv ← 2;
ELSE
   IF ( B )
      Rv ← 0;
   ELSE
      Rv ← 1;
return Rv;
```

Change proposal (2)

```
IF ( B )
   Rv ← 0;
ELSE
   IF ( A )
      Rv ← 2;
   ELSE
      Rv ← 1;
return Rv;
```

Change proposal (3)

```
Rv ← 1;
IF ( B )
   Rv ← 0;
IF ( A )
   Rv ← 2;
return Rv;
```

Of these change proposals, the one that operates correctly is [E].

Answer group for A and B

a) `(Data & Mask) = '00000000'B`
b) `(Data & Mask) = Data`
c) `(Data & Mask) = Mask`
d) `(Data | Mask) = '00000000'B`
e) `(Data | Mask) = Mask`

Answer group for C and D

a) always returns the return value 0
b) never returns the return value 0
c) always returns the return value 1
d) never returns the return value 1
e) always returns the return value 2
f) never returns the return value 2

Answer group for E

a) Change proposal (1)
b) Change proposal (2)
c) Change proposal (3)

Subquestion 2

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

[Description of the function `BitCount`]

The function `BitCount` counts the number of bits whose values are 1 in an 8-bit data and returns the result.

The function `BitCount` is declared as follows:

```
FUNCTION: BitCount(8-bit Binary: Data)
```

The argument `Data` contains an 8-bit data to be checked.

Two programs are provided: Program 2, which uses a basic algorithm, and Program 3, which uses an efficient algorithm.
For Programs 2 and 3, the processing time of each line is shown. It indicates the required time in a certain time unit to execute that line of process once. For a line on which the processing time is not shown, its processing time can be ignored.
In Program 3, the operator "-" performs subtraction by treating both operands as 8-bit unsigned integers.

[Program 2]

```
Processing
    time
          FUNCTION: BitCount(8-bit Binary: Data) {
             8-bit Binary: Work
             INT: Count, Loop

    1        Work ← Data;
    1        Count ← 0;
    4        FOR (Loop ← 0; Loop < 8; Loop ← Loop + 1) {
    3           IF (Rightmost bit of Work is '1'B)        /*** α ***/
    1              Count ← Count + 1;
    1           Shift Work one bit to the right logically;
             }
    2        return Count;     /* returns Count as the return value */
          }
```

[Program 3]

```
Processing
    time
          FUNCTION: BitCount(8-bit Binary: Data) {
             8-bit Binary: Work
             INT: Count

    1        Work ← Data;
    1        Count ← 0;
    2        WHILE (Work ≠ '00000000'B) {
    1           Count ← Count + 1;
    3           Work ← Work & (Work - 1);     /*** β ***/
             }
    2        return Count;     /* returns Count as the return value */
          }
```

With respect to Program 2, the IF statement on the line marked /*** α ***/ can be rewritten by using logical expressions such as:

```
IF ((Work & '00000001'B) = '00000001'B),
IF ((  F  ) = Work), and
IF ((  G  ) ≠ Work).
```

The processing efficiency of Programs 2 and 3 is considered. Table 1 shows the results of the comparison of the processing times of Programs 2 and 3.

Table 1 Comparison of the processing times of Programs 2 and 3

	Minimum	Maximum
Program 2	72	80
Program 3	H	54

With respect to Program 3, an efficient algorithm is used to update the value of `Work` on the line marked /*** β ***/.

Assume that Program 3 is called and the argument `Data` contains `'01100110'B`. Immediately after the first execution of the line marked /*** β ***/, the value of `Work` becomes ' I 'B.

By carrying out the process of bit replacement in this manner, the number of times the process on the line marked /*** β ***/ is executed is the same as the number of bits that have value 1 in the data to be checked.

Answer group for F and G

a) `Work & '00000001'B`
b) `Work & '11111110'B`
c) `Work | '00000001'B`
d) `Work | '11111110'B`

Answer group for H

a) 6
b) 10
c) 20
d) 22

Answer group for I

a) 00010011
b) 00100110
c) 00110010
d) 01100100

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

Q7. Read the following description of a C program, and then answer Subquestions 1 through 3.

The "Pillow Passing Game" is a classic party game that involves players, chairs, music and a pillow in which the pillow is passed from one player to another. In this game, all players are seated on chairs arranged in a loop. While the music is playing, the players pass the pillow to the next player. When the music stops, the player who holds the pillow will be eliminated from the game with his/her chair; the elimination is repeated until only one player remains. The last player is considered as the winner.

[Program Description]
The game can be simulated by a linear circular linked list and random numbers. The list represents the players of the game seated on the looped chairs. Each node of the list holds a unique integer number (ranges from 1 to the number of players) that identifies a player. A random number simulates how many times the pillow is passed among the players while the music is playing.

At the beginning of the program, it receives a positive integer number `num_players` (≤ 100) as the total number of players from the standard input. The program then creates a linear circular linked list consisting of `num_players` of nodes. If `num_players` is 1, the winner will always be the only player. Otherwise, the program sets a pointer `has_pillow` to the first node in the list and then moves `has_pillow` in the forward direction by `rd` times where `rd` takes a random integer number that ranges from 1 to `MAX_NUM_PASSES`. When `has_pillow` has finished moving, the node pointed to by `has_pillow` is deleted from the list. `has_pillow` is then reset to point to the next node of the deleted one and moves randomly again. This process is repeated until the time when only one node remains.

Figures 1 through 5 illustrate an example of the program execution for `num_players` = 4. The program first creates a linear circular linked list of 4 nodes. Figure 1 illustrates the initial status of the list. Here, the pointer `players` points to the representative node of the list.

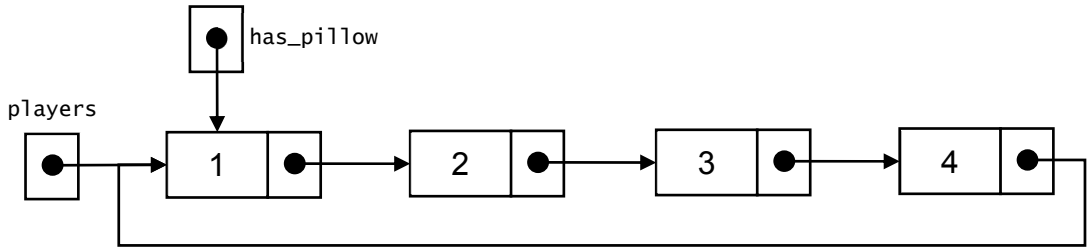


Figure 1 Initial status of the list

Assume that a random number 2 is generated and set to `rd` at the first elimination process. The pointer `has_pillow` is moved forward by 2 nodes as in Figure 2.

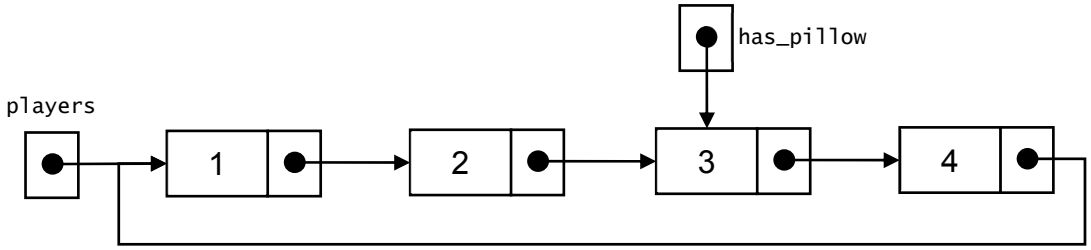


Figure 2 Movement of `has_pillow`

After the node pointed to by `has_pillow` is deleted from the list, `has_pillow` is set to the node immediately after the deleted node and the list appears as in Figure 3.

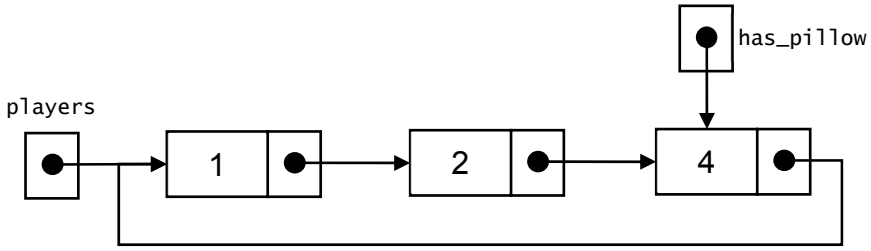


Figure 3 Status of the list at the end of the first elimination

Assume that 6 is set to `rd` at the second elimination process. The 6th player starting at player 4 is player 4 itself. Figure 4 illustrates the list after player 4 is deleted.

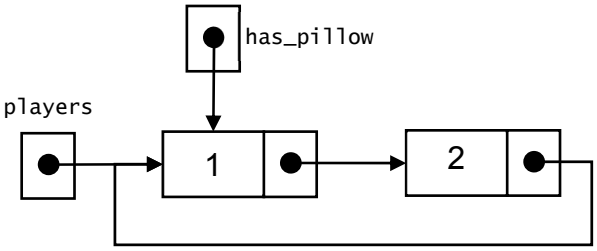


Figure 4 Status of the list at the end of the second elimination

Similarly, assume that 8 is set to `rd` at the third elimination process. Figure 5 illustrates the list after player 1 is deleted.

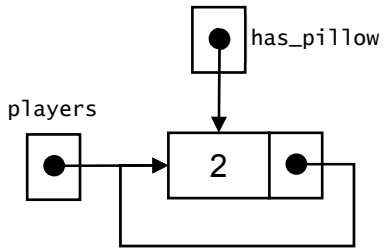


Figure 5 Status of the list at the end of the third elimination

At this point, the program stops the elimination process. The winner is player 2.
For the example shown above, the program outputs the following result:

```
How many players:= 4

Current LinkedList:= 1 2 3 4
Pass = 1
Random Number:= 2

Current LinkedList:= 1 2 4
Pass = 2
Random Number:= 6

Current LinkedList:= 1 2
Pass = 3
Random Number:= 8

Current LinkedList:= 2
Winner = 2
```

Tables 1 and 2 show the descriptions of user defined functions and standard library functions used in the program.

Table 1 User defined functions used in the program

Function Name	Description
`player_t *Setup(int num_players)`	Creates new linear circular linked list holding `num_players` nodes. Returns a pointer to the representative node of the list.
`int PlayGame(player_t *players)`	Determines the winner of the game with given linear circular linked list `players`. Returns the number identifying the winning player.
`void Display(player_t *players)`	Displays information of current linear circular linked list `players`.

Table 2 Standard library functions used in the program

Function Name	Description
`void *malloc(size_t size)`	Allocates *size* bytes of memory block dynamically. Returns a pointer that points to that memory.
`void free(void *ptr)`	Deallocates the memory block pointed to by *ptr*, which was previously allocated by `malloc`.
`void srand(unsigned int seed)`	Sets the starting point for generating a series of pseudo-random numbers.
`int rand(void)`	Returns an integer pseudo-random number from 0 to $2^{15}-1$ (or $2^{31}-1$; environment-dependent).
`time_t time(time_t *timer)`	Returns an integer number that represents the current time in seconds. If *timer* is not `NULL`, the return value is also stored in the variable *timer*.

Note: Functions `printf()` and `scanf()` are not shown.

[Program]

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAX_NUM_PASSES 10

struct player {
   int number;
   struct player *next;
};
typedef struct player player_t;

player_t *Setup(int num_players);
int PlayGame(player_t *players);
void Display(player_t *players);

int main() {
   int win, num_players;
   player_t *players;

   printf("How many players:= ");
   scanf("%d", &num_players);
   players = Setup(num_players);
   Display(players);
   win = PlayGame(players);
   printf("\nWinner = %d\n", [    A    ]);
   return 0;
}
```

```
player_t *Setup(int num_players) {
   player_t *temp, *prev, *players = NULL;
   int i;

   for (i = 1; i <= num_players; i++) {
      temp = (player_t *)malloc(sizeof(player_t));
      if (players == NULL) {
         players = temp;
         prev = temp;
      }
      temp->number = [      B      ];
      temp->next = players;
      prev->next = temp;
      prev = temp;
   }
   return players;
}

int PlayGame(player_t *players) {
   player_t *p, *has_pillow;
   int rd, pass, i;

   srand(time(NULL));
   pass = 1;
   has_pillow = players;
   while ([      C      ]) {
      rd = rand() % (MAX_NUM_PASSES - 1) + 1;

      for (i = 0; i < rd; i++) {
         p = has_pillow;
         has_pillow = has_pillow->next;  /*** α ***/
      }
      [      D      ];
      if (has_pillow == players) {
         players = has_pillow->next;     /*** β ***/
      }
      free(has_pillow);
      has_pillow = p->next;              /*** γ ***/
      printf("\nPass = %d", pass);
      printf("\nRandom Number:= %d", rd);
      printf("\n");
      Display(players);
      pass++;
   }
   return players->number;
}
```

```
void Display(player_t *players) {
   player_t *temp;

   printf("\nCurrent LinkedList:= ");
   temp = players;
   do {
      printf("%d ", temp->number);
      [ E ];
   } while (temp != players);
}
```

Subquestion 1

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

Answer group for A

a) `num_players`
b) `players`
c) `players->number`
d) `win`
e) `win->number`

Answer group for B

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

Answer group for C

a) `has_pillow != players`
b) `p->next != players->next`
c) `players->next != has_pillow->next`
d) `players->next != players`

Answer group for D

a) `p = p->next`
b) `p = p->next->next`
c) `p->next = has_pillow`
d) `p->next = has_pillow->next`

Answer group for E

a) `players = players->next`
b) `players = temp`
c) `temp = players`
d) `temp = temp->next`

Subquestion 2

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

Suppose the program created the linear circular linked list as shown in Figure 6 for `num_players` = 5. Given a series of random numbers 4, 2, 1, and 2 set to the variable `rd` in this order, the function `PlayGame` returns [F].

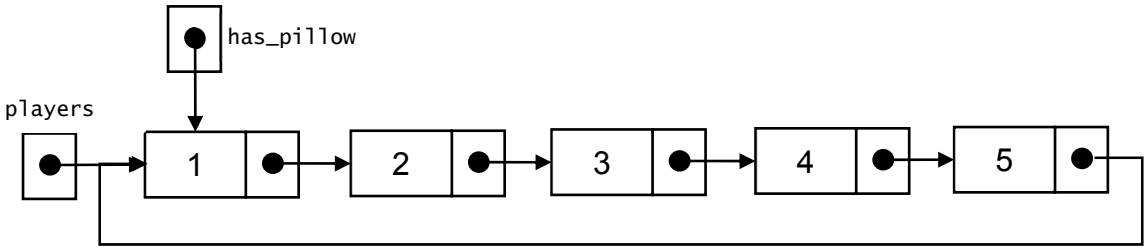


Figure 6 Linear circular linked list for `num_players` = 5

Answer group for F

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

Subquestion 3

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

The program will be modified according to the following specification change:

> Once a player is eliminated from the game, the pillow is passed again from the player who is seated immediately before the eliminated player. To apply this change to the program, the assignment statement on the line marked [G] should be replaced by the following line:
>
> [H];

Answer group for G

a) `/*** α ***/` b) `/*** β ***/` c) `/*** γ ***/`

Answer group for H

a) `has_pillow = p`
b) `has_pillow = players`
c) `players = has_pillow`
d) `players = p`

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

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

[Program Description]

This program is a prototype implementation of a spell checker. The spell checker checks if the given words are spelled correctly. To cross match, this program utilizes a pre-populated dictionary called `WordDictionary`. Given a word, the spell checker tries to find and suggest related words.

The `WordDictionary` class (Program 1) represents a dictionary with pre-populated English words. The class has the following members.

(1) The `dictWords` field is a `String` array containing pre-populated known words. All words are in lowercase and are sorted in ascending order.
(2) The `contains` method looks up the given word to determine whether the dictionary contains the word.
(3) The `getIterable` method returns an `Iterable` to return an `Iterator` that iterates over the words in `dictWords`. In `Iterator`, the `hasNext` method returns `true` if the iteration has more elements, and the `next` method returns the next element in the iteration.

[Program 1]

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

final class WordDictionary {
   private static final String[] dictWords = {
      "about", "because", "could", "food", "is", "other", "people",
      "their", "there", "think", "to", "which", "would"
   };

   private WordDictionary() {
   }
```

```
    static boolean contains(String word) {
       String lowerCaseWord = word.toLowerCase(Locale.ENGLISH);
       for (String element : dictWords) {
          if (     A     ) {
             return true;
          }
       }
       return false;
    }

    static Iterable<String> getIterable() {
       // Note that the default remove() implementation throws
       // an UnsupportedOperationException.
       return () -> new Iterator<String>() {
          private int index;

          @Override
          public boolean hasNext() {
             return     B     ;
          }

          @Override
          public String next() {
             if (hasNext()) {
                return dictWords[     C     ];
             }
             throw new NoSuchElementException();
          }
       };
    }
}
```

The `EditDistance` class (Program 2) calculates related words to suggest for spell checking using the edit distance based on the Wagner–Fischer algorithm.

An edit distance indicates the number of insertions, deletions, and/or substitutions of characters needed to transform one word into another. If a character needs to be inserted or deleted, it is treated as a cost or edit distance of one unit. If a character is substituted, it is treated as a cost or edit distance of two units as a substitution can be considered as one deletion and one insertion of a character. The algorithm compares both strings starting with an empty string to the full-length string, and always chooses the minimum edit distance (cost) from the three possible operations (an insertion, a deletion, or a substitution) at the point. The intermediate edit distances are recorded in a matrix.

There are three situations or operations while transforming a string to another string. The following are examples of the operations.

(1) Insertion: The edit distance from "cat" to "cats" is 1 because an insertion of the character 's' is required.
(2) Deletion: The edit distance from "boot" to "bot" is 1 because a deletion of a character 'o' is required.
(3) Substitution: The edit distance from "let" to "lot" is 2 because a substitution of the character 'e' with the character 'o' is required.

The class has the following members:

(1) The `getSubstitutionCost` method returns an edit distance (cost) of 2 if two characters are different, which means that it requires a substitution, or 0 (zero) otherwise.
(2) The `getDistance` method calculates the edit distance between the two words, `baseWord` that has been given when instantiating this class and the word given by the `dictWord` parameter, based on the Wagner–Fischer algorithm.

The `SpellChecker` class (Program 3) suggests alternative words if any. The class has the following members:

(1) The `THRESHOLD` field is an integer constant to denote the acceptable edit distance to enable very close words to be suggested based on their edit distance of 2.
(2) The `suggestWords` method finds out the words within the edit distance of `THRESHOLD` in the dictionary and suggests those words. If no close words are found, it returns an empty `Set`.

The `Tester01` class (Program 4) tests the words suggested by `SpellChecker` and generates the following output for the given input, "`thier`":

```
other there their
```

[Program 2]

```
import java.util.Locale;

class EditDistance {
   private final String baseWord;

   EditDistance(String baseWord) {
      this.baseWord = baseWord.toLowerCase(Locale.ENGLISH);
   }

   private int getSubstitutionCost(char a, char b) {
      return a    D    b ? 0 : 2;
   }
```

```
    int getDistance(String dictWord) {
       int[][] d
             = new int[baseWord.length() + 1][dictWord.length() + 1];

       for (int i = 0; i < d.length; ++i) {
          for (int j = 0; j < d[0].length; ++j) {
             if (i == 0) {
                d[0][j] = j;
             } else if (j == 0) {
                d[i][0] = i;
             } else {
                int cost = getSubstitutionCost(baseWord.charAt(i - 1),
                      dictWord.charAt(j - 1));
                // Calculate costs of the three different operations
                int substitution = cost + d[i - 1][j - 1];
                int deletion = d[i - 1][j] + 1;
                int insertion = d[i][j - 1] + 1;
                // Choose the cheapest one, and note down the value
                d[i][j] = Math.min(substitution,
                      Math.min(deletion, insertion));
             }
          }
       }
       return d[d.length - 1][d[0].length - 1];
    }
 }
```

[Program 3]

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

 // Location 2

 final class SpellChecker {
    private static final int THRESHOLD = 2;

    private SpellChecker() {
    }
```

```
    static Set<String> suggestWords(String word) {
       EditDistance editDistance = new EditDistance(word);
       Set<String> suggestedWords = new HashSet<>();
       for (String dictWord : WordDictionary.getIterable()) {
          if (editDistance.getDistance(dictWord) <= THRESHOLD) {
             suggestedWords.add(dictWord);
          }
       }
       return suggestedWords;
    }

    // Location 3
 }
```

[Program 4]

```
 import java.util.Set;

 public class Tester01 {
    public static void main(String[] args) {
       Set<String> suggestedWords = SpellChecker.suggestWords("thier");
       if (!suggestedWords.isEmpty()) {
          for (String suggested : suggestedWords) {
             System.out.print(suggested + " ");
          }
          System.out.println();
       } else {
          System.out.println("no suggestions");
       }
    }
 }
```

Subquestion 1

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

Answer group for A

a) `!element.equals(lowerCaseWord)`
b) `element != lowerCaseWord`
c) `element == lowerCaseWord`
d) `element.equals(lowerCaseWord)`

Answer group for B

a) `dictWords[index] != null`
b) `index < dictWords.length`
c) `index <= dictWords.length`
d) `index == dictWords.length`
e) `index > dictWords.length`
f) `index >= dictWords.length`

Answer group for C

a) `--index` b) `++index` c) `index`
d) `index--` e) `index++`

Answer group for D

a) `!=` b) `<` c) `<=`
d) `==` e) `>` f) `>=`

Subquestion 2

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

`SpellChecker` (Program 3) has been modified so that if a word is present in the dictionary, then it remains unchanged in the output. If related words are found, they are shown as alternatives in parentheses in addition to the original word, e.g., `OriginalWord(/alternative1/alternative2/…)`. If none are found, the output is a question mark in parentheses, e.g., `OriginalWord(?)`.

The `Tester02` class (Program 5) tests the implementation of the modified `SpellChecker` and generates the following output for the given input, "`Beacaus theer API-key is abt to expire`": (Note that the following output is wrapped owing to the paper width, and that the actual output is shown in a single line.)

```
Beacaus(/because) theer(/other/there/their) API-key(?) is abt(/about)
to expire(?)
```

Here, suggested words were found and shown in parentheses for the words "`Beacaus`", "`theer`", and "`abt`" which were absent in the dictionary. Each suggestion is separated by a slash character ("/"). However, the words "`API-key`" and "`expire`" did not generate any suggestions; they are therefore followed by a question mark. The words "`is`" and "`to`" were found in the dictionary; hence, they remained unchanged in the output. To facilitate the modification, the following lines have been inserted at the locations marked by "`Location 2`" and "`Location 3`" as a comment in Program 3.

```
// Location 2
import java.util.Scanner;
// Location 3
   static String checkSentence(String sentence) {
      // Scanner breaks `sentence' into tokens delimited by
      // whitespace and iterates over tokens.
      Scanner scanner = new Scanner(sentence);
      StringBuilder suggestions = new StringBuilder();
      while (scanner.hasNext()) {
         String word = scanner.next();
         suggestions.append(word);
         if (!WordDictionary.contains(word)) {
            suggestions.append(     E     );
            Set<String> suggestedWords = suggestWords(word);
            if (suggestedWords.isEmpty()) {
               suggestions.append(     F     );
            } else {
               for (String suggestedWord : suggestedWords) {
                  suggestions.append(     G     ).append(suggestedWord);
               }
            }
            suggestions.append(")");
         }
         if (scanner.hasNext()) {
            suggestions.append(" ");
         }
      }
      return suggestions.toString();
   }
```

[Program 5]

```
public class Tester02 {
   public static void main(String[] args) {
      System.out.println(SpellChecker.checkSentence(
            "Beacaus theer API-key is abt to expire"));
   }
}
```

Answer group for E through G

a) "(" b) ")" c) "/"

d) "?" e) scanner.next() f) word

Subquestion 3

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

If the `dictWords` field in the `WordDictionary` class contains more words, there will be the need for a faster search mechanism than the current linear search in the `contains` method. The following is another implementation of the same method with the binary search.

The following `compareTo` method returns 0 if the argument string is equal to this string, a value less than 0 if this string is lexicographically less than the string argument, and a value greater than 0 if this string is lexicographically greater than the string argument.

```
static boolean contains(String word) {
   String lowerCaseWord = word.toLowerCase(Locale.ENGLISH);
   int low = 0;
   int high = dictWords.length - 1;
   while (low <= high) {
      int mid = (low + high) / 2;
      int comp = dictWords[mid].compareTo(lowerCaseWord);
      if (comp < 0) {
         low = [ H ];
      } else if (comp > 0) {
         high = [ I ];
      } else {
         return true;
      }
   }
   return false;
}
```

Answer group for H and I

a) `high - mid` b) `low + mid` c) `mid`

d) `mid + 1` e) `mid - 1`