Q2 2021

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 sample question below.

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

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

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

   [Sample Answer]

Sample	ⓐ ⓑ ⓒ ● ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ

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

Notations used in the pseudo-language

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

[Declaration, comment, and process]

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

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

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

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

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

[Logical constants]

true, false

[Operators and their precedence]

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

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

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

Q1. Read the following description of risks in information security, and then answer Subquestions 1 and 2.

Company S decides to define the assessment criteria and conduct a risk assessment of the company's information assets to manage risks in the company's information security by quantifying them. As a part of this decision, Mr. A is in charge of the risk assessment of server X and server Y.

[Calculation of values of risks]

For each aspect, there are multiple threats and corresponding vulnerabilities. The risk values of information assets are calculated by the following formula:

Risk value = (Importance of information asset)
× (Value of threat) × (Value of vulnerability)

[Criteria for determining the importance of information assets]

Company S defines the evaluation criteria and values of the importance of information assets based on the risk assessment. Tables 1, 2 and 3 show the defined evaluation criteria and the values in terms of confidentiality, integrity, and availability, respectively.

Table 1 Evaluation criteria and values of confidentiality

Evaluation criteria	Value
Can be disclosed externally	1
Can only be disclosed internally	2
Can only be disclosed within the department	3
Can only be disclosed to a minimum number of restricted entites	4

Table 2 Evaluation criteria and values of integrity

Evaluation criteria		Value
Loss of information integrity:	does not affect operations	1
	has a slight impact on operations	2
	has a significant impact on operations	3

Table 3 Evaluation criteria and values of availability

Evaluation criteria		Value
Suspension of use by non-regular maintenance is permitted up to:	24 hours / year	1
	5 hours / year	2
	1 hour / year	3
	10 minutes / year	4
	1 minute / year	5

[Criteria for determining threats and vulnerabilities]

Company S also defines the evaluation criteria and values of threats and vulnerabilities as shown in Tables 4 and 5 respectively.

Table 4 Evaluation criteria and values of threats

Evaluation criteria	Value
Low probability of occurrence	1
Moderate probability of occurrence	2
High probability of occurrence	3

Table 5 Evaluation criteria and values of vulnerabilities

Evaluation criteria	Value
Appropriate controls and measures are in place	1
Some controls and measures are in place	2
Inadequate controls and measures	3

[Server X and Server Y]

Server X has a database of general supplier information. The general supplier information includes the supplier code, official name, abbreviation, address, and telephone number of the supplier.

Server Y has a database of business transaction information. The business transaction information includes the supplier code, purchased product code, unit price, purchase history.

To evaluate the importance of servers X and Y in terms of confidentiality, integrity, and availability, Mr. A interviewed the related departments regarding the general supplier information and business transaction information, and summarized their responses.

[Responses of interview from the related departments]

(1) General supplier information

(i) General supplier information pertains to that published in telephone directories and on each supplier's website. However, this information cannot be disclosed externally because Company S does not want its business relationship with some of the suppliers to be known by its competitors.

(ii) The general supplier information is not used in EDI (Electronic Data Interchange) with the suppliers; hence, the impact on procurement operations is slight even if the information contains errors.

(iii) Employees use general supplier information for checking telephone numbers or printing address labels for business letters. When server X is not available, the information can be obtained by alternate means.

(2) Business transaction information

(i) Business transaction information must not be known to competitors and competitive suppliers of Company S. Furthermore, the information cannot be disclosed to other departments even internally.

(ii) Business transaction information errors have a considerable impact on operations including purchasing and payment.

(iii) Business transaction information is used for online entries in procurement during business hours and night time batch processing. The system that processes this information cannot have a downtime of 4 hours or more annually except for maintenance.

[Status of threats and vulnerabilities / Acceptable risk level]

Mr. A investigates the threats to servers and the vulnerabilities of Company S against these threats. He assesses these threats and vulnerabilities based on the criteria as shown in Table 4 and Table 5. The major threats and vulnerabilities are shown in Table 6.

Company S defines acceptable risk levels as shown in Table 7. If the value of each risk of the information assets is less than or equal to the value shown in the table, the risk is retained; otherwise, measures to resolve the risk are implemented.

Table 6 Values of major threats and vulnerabilities of servers X and Y

Threat		Vulnerability	
Type	Value	Type	Value
Virus infection	3	Anti-virus software not installed	3
Unauthorized access	3	Inadequate access control	2
Failure	2	Insufficient maintenance	3
Spoofing	2	Inadequate password management	2
Wiretapping	2	Latest recommended encryption not used	1

Table 7 Acceptable risk levels

Confidentiality	13
Integrity	15
Availability	10

Table 8 Risk assessment of servers X and Y (excerpt)

Information assets			Threat		Vulnerability		Risk
Title	Importance		Description	Value	Description	Value	Value
	Category	Value					
Server X	Confidentiality	▨	⋮				
			Spoofing	▨	B	▨	D
			⋮				
	Integrity	▨	Virus infection	▨	Antivirus software not installed	▨	18
			Unauthorized access	▨	C	▨	12
			Spoofing	▨	B	▨	8
			⋮				
	Availability	▨	⋮				
Server Y	Confidentiality	▨	⋮				
			Unauthorized access	▨	C	▨	18
			⋮				
	Integrity	▨	Virus infection	▨	Antivirus software not installed	▨	27
			⋮				
	Availability	A	⋮				

Note: The shaded parts ▨ are not shown. “...” indicates omission.

[Risk assessment of server X and server Y]

Mr. A conducts a risk assessment of servers X and Y according to the criteria defined in Tables 1 to 5, the interview responses, and the status of servers X and Y. For the risk assessment in company S, the smallest value from the applicable criteria is selected from the assessment criteria defined in Tables 1 to 3.

The partial assessment results are shown in Table 8.

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in Table 8.

Answer group for A

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

Answer group for B and C

a) Data loss due to data replication (backup) error
b) Inadequate access control
c) Inadequate capacity management (Availability)
d) Inadequate password management
e) Insufficient maintenance
f) IT system error in batch processing (Integrity)

Answer group for D

a) 4 b) 6 c) 8 d) 9 e) 12

Subquestion 2

Concerning the area indicated by the dashed lines ⬚ in the integrity section for server X in Table 8, from the answer group below, select the appropriate risk measure based on the acceptable risk levels of Company S.

Answer group

a) Increasing the frequency of periodic maintenance
b) Installing antivirus software
c) Installing IDS (Intrusion Detection System)
d) Strengthening password management
e) Using public key cryptography

Q2. Read the following description of CPU scheduling, and then answer Subquestions 1 and 2.

Modern multiprogramming-based operating systems allow more than one process to be loaded into the memory for execution. The loaded processes share the CPU via time scheduling. CPU scheduling handles the problem of deciding which processes in the ready queue are to be executed at each specific time unit. One of the basic methods uses the Shortest-Job-First (SJF) algorithm. In SJF, the process manager selects the process with the shortest estimated CPU burst time among the waiting processes. Here, scheduling is performed by examining the length of the CPU burst time of each process.

Several criteria are applied to evaluate the performance of scheduling algorithms.
Table 1 shows the definitions of CPU burst time, waiting time, response time, arrival time, and turnaround time.

Table 1 Definitions of criteria used to evaluate scheduling algorithms

CPU burst time	The amount of time required by a process for CPU execution.
Waiting time	The amount of time that a process spends waiting in the ready queue.
Response time	The amount of time from the submission of a request until the first response is produced.
Arrival time	The time at which a process enters the ready queue.
Turnaround time	The interval from the time of process submission to the time of process completion (i.e., the total of CPU burst time and waiting time).

Subquestion 1

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

There are four processes: P1, P2, P3, and P4. Table 2 shows the CPU burst time of each process. Assume that all processes enter the ready queue simultaneously at time 0, and that they are ready for execution.

Table 2 CPU burst time of each process

Process	CPU burst time
P1	6
P2	8
P3	7
P4	3

Figure 1 shows the process executed on each time unit using the SJF algorithm. The process manager selects P4 as the process to start execution at time 0 because it has the shortest CPU burst time among the ready processes. Each time a process is completed, the process manager selects another process with the next shortest CPU burst time.

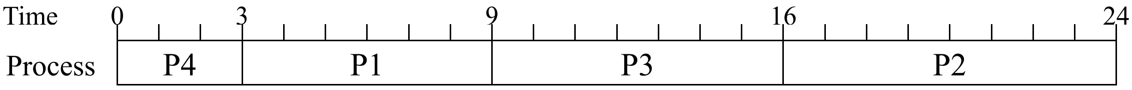


Figure 1 Process executed on each time unit using the SJF algorithm

Table 3 shows the execution results of the processes of Table 2 in SJF algorithm.

Table 3 Execution results of the processes in SJF algorithm

Process	Waiting time	Response time	Turnaround time
P1	3	3	9
P2	A		
P3	9	9	16
P4	0	0	3

Answer group for A

a)

6	6	8

b)

16	8	24

c)

16	16	8

d)

16	16	24

Subquestion 2

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

If the arrival time varies for each process, the process manager has a chance to reconsider whether or not it should stop the current executing process and switch to the arriving one. A preemptive SJF algorithm preempts the current executing process when another process arrives and has shorter CPU burst time.

There are four other processes (P5, P6, P7, and P8). Table 4 shows the arrival time and CPU burst time of each process.

Table 4 Arrival time and CPU burst time of each process

Process	Arrival time	CPU burst time
P5	0	8
P6	1	4
P7	2	9
P8	3	5

Figure 2 shows the process executed on each time unit using the preemptive SJF algorithm. It denotes that P5 will be provided a CPU burst time first and P7 will be the last. Because P6 has the shortest CPU burst time at its arrival time, it preempts the current process and is allocated the CPU until its end of execution.

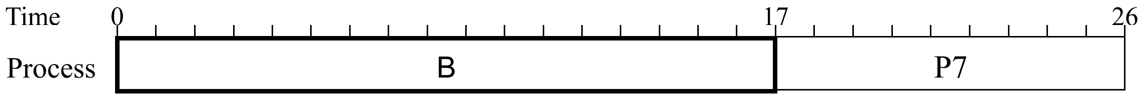


Figure 2 Process executed on each time unit in preemptive SJF algorithm

Table 5 shows the execution results of the processes of Table 4 using the preemptive SJF algorithm.

Table 5 Execution results of the processes using the preemptive SJF algorithm

Process	Waiting time	Response time	Turnaround time
P5	9	0	17
P6	0	0	4
P7	15	15	24
P8	C		

The non-preemptive SJF algorithm will allow the currently running process to finish its CPU burst time. Subsequently, the process manager selects the shortest process in the queue for the next execution. Figure 3 shows the process executed on each time unit using the non-preemptive SJF algorithm. Unlike the preemptive SJF algorithm, the non-preemptive SJF algorithm allows the current process to continue, even if another shorter process arrives during its execution.

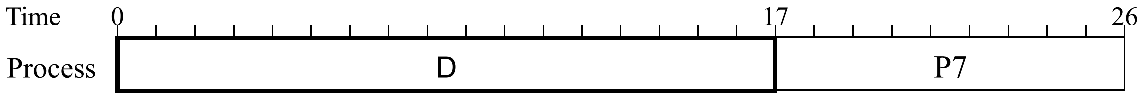


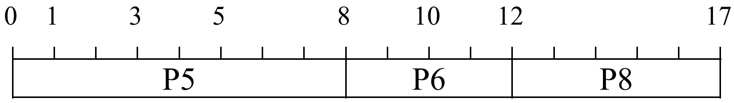
Figure 3 Process executed on each time unit using the non-preemptive SJF algorithm

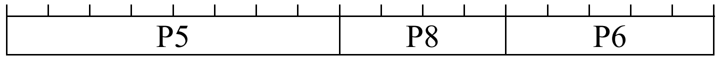
Table 6 shows the execution results of the processes of Table 4 using the non-preemptive SJF algorithm.

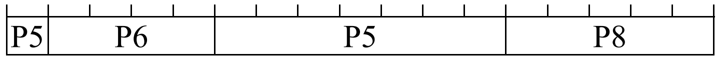
Table 6 Execution results of the processes using the non-preemptive SJF algorithm

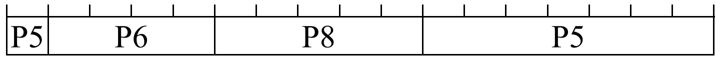
Process	Waiting time	Response time	Turnaround time
P5	0	0	8
P6	7	7	11
P7	15	15	24
P8	E		

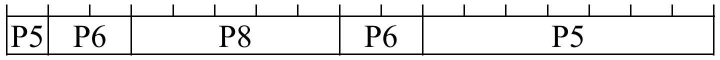
Answer group for B and D

a)


b)


c)


d)


e)


Answer group for C and E

a)

0	0	5

b)

2	2	7

c)

5	5	9

d)

9	9	14

Q3. Read the following description of a database for order management, and then answer Subquestions 1 and 2.

Factory U, a furniture manufacturer, uses an order management database to manage the order status of materials, which are required for its production.
When factory U orders materials, they usually arrive together as a single delivery. However, some arrive in two or more delivery batches. For example, if factory U orders 100 rubber cushions, 80 of these arrive earlier, and the remaining 20 arrive later.

The database consists of the following four tables; the table structure and sample data are shown for each table. The underline ____ indicates the primary key, and the dotted underline .... indicates the foreign key.

(1) Material Table

The Material table contains the information regarding the materials that factory U can order.

MaterialNo	MaterialName	UnitPrice
1	Plastic sheet	10
2	Rubber cushion	15
3	Wood board	12

(2) Order Table

The Order table contains the information regarding each order date of materials.

OrderNo	OrderDate
1	2021-04-02
2	2021-04-05
3	2021-04-06

(3) OrderDetail Table

The OrderDetail table contains the information regarding materials and their quantities that are included in each order.

OrderDetailNo	OrderNo	MaterialNo	Quantity
1	1	1	40
2	1	2	20
3	2	1	10
4	2	3	30
5	3	1	10

(4) Arrival Table

The Arrival table contains the information regarding the arrival status of ordered materials.

ArrivalNo	ArrivalDate	ArrivalQuantity	OrderDetailNo	OrderNo
1	2021-04-05	20	1	1
2	2021-04-06	10	2	1
3	2021-04-07	30	4	2
4	2021-04-08	20	1	1
5	2021-04-09	10	3	2

From the sample data shown above, the current arrival status by OrderDetailNo is as follows. A value in parentheses indicates the ordered quantity and the quantity that arrives.

OrderDetailNo	Arrival status
1 (40)	Completed by ArrivalNo 1 (20) and ArrivalNo 4 (20).
2 (20)	Partially completed by ArrivalNo 2 (10). Not arrived (10).
3 (10)	Completed by ArrivalNo 5 (10).
4 (30)	Completed by ArrivalNo 3 (30).
5 (10)	Not arrived (10).

Subquestion 1

The following SQL statement SQL1 displays the total quantity of materials that are recorded in the OrderDetail table by material.

```
-- SQL1 --
SELECT m.MaterialNo, m.MaterialName, d.TotalQuantity
FROM (SELECT MaterialNo, MaterialName
      FROM Material) m,
     (SELECT MaterialNo, SUM(Quantity) as TotalQuantity
      FROM OrderDetail
      GROUP BY MaterialNo) d
WHERE m.MaterialNo = d.MaterialNo
ORDER BY d.TotalQuantity
```

From the answer group below, select the correct output of SQL1, when SQL1 is executed using the sample data shown above.

Answer group

a)

MaterialNo	MaterialName	TotalQuantity
1	Plastic sheet	60

b)

MaterialNo	MaterialName	TotalQuantity
1	Plastic sheet	60
2	Rubber cushion	20
3	Wood board	30

c)

MaterialNo	MaterialName	TotalQuantity
1	Plastic sheet	60
3	Wood board	30
2	Rubber cushion	20

d)

MaterialNo	MaterialName	TotalQuantity
2	Rubber cushion	20
3	Wood board	30
1	Plastic sheet	60

e)

MaterialNo	MaterialName	TotalQuantity
3	Wood board	30
2	Rubber cushion	20
1	Plastic sheet	60

Subquestion 2

From the answer groups below, select the correct answer to be inserted in each blank □ in the following description and SQL statement SQL2.

Note:

- `LEFT JOIN` returns all rows from the left table with matching rows in the right table.
- In this question, the function `ISNULL(`*exp*`, `*val*`)` returns the value of *exp* if *exp* is not NULL; otherwise, it returns the value of *val*.

The following SQL statement SQL2 displays the arrival status of ordered materials by order number.

```
-- SQL2 --
SELECT d.OrderNo, d.TotalQuantity,
       ISNULL(      A      ) AS ArrivalQuantity,
       (d.TotalQuantity - ISNULL(      A      )) AS NeedToArrival
FROM      (SELECT OrderNo,      B      
           FROM OrderDetail
           GROUP BY OrderNo) d
LEFT JOIN (SELECT OrderNo,      C      
           FROM Arrival
           GROUP BY OrderNo) a
ON d.OrderNo = a.OrderNo
```

From the sample data on the previous pages, SQL2 creates the following output. During this process, when two tables (d and a) are LEFT JOIN-ed, the joined table has one row containing NULL fields. The d.OrderNo of that row is D.

OrderNo	TotalQuantity	ArrivalQuantity	NeedToArrival
1	60	50	10
2	40	40	0
3	10	0	10

Answer group for A

a) a.ArrivalQuantity, 0

b) a.ArrivalQuantity, d.TotalQuantity

c) d.TotalQuantity, 0

d) d.TotalQuantity, a.ArrivalQuantity

Answer group for B and C

a) ArrivalQuantity

b) COUNT(ArrivalQuantity) AS ArrivalQuantity

c) COUNT(Quantity) AS TotalQuantity

d) Quantity AS TotalQuantity

e) SUM(ArrivalQuantity) AS ArrivalQuantity

f) SUM(Quantity) AS TotalQuantity

Answer group for D

a) 1   b) 2   c) 3

Q4. Read the following description of a network installation for company V, and then answer Subquestions 1 through 3.

Company V designs the internal network infrastructure with variable length subnet mask network address scheme (different segments of networks) using private IP addresses (10.1.1.0/24 and 172.16.100.0/24). These network segments are connected to each other using Core Router (CR), Router 1 (R1), and Router 2 (R2). An administrator planned the IP address scheme of company V according to Table 1. The Network Address Translation function is placed at CR to access the Internet from the internal network.
The design of company V's internal network is shown in Figure 1.

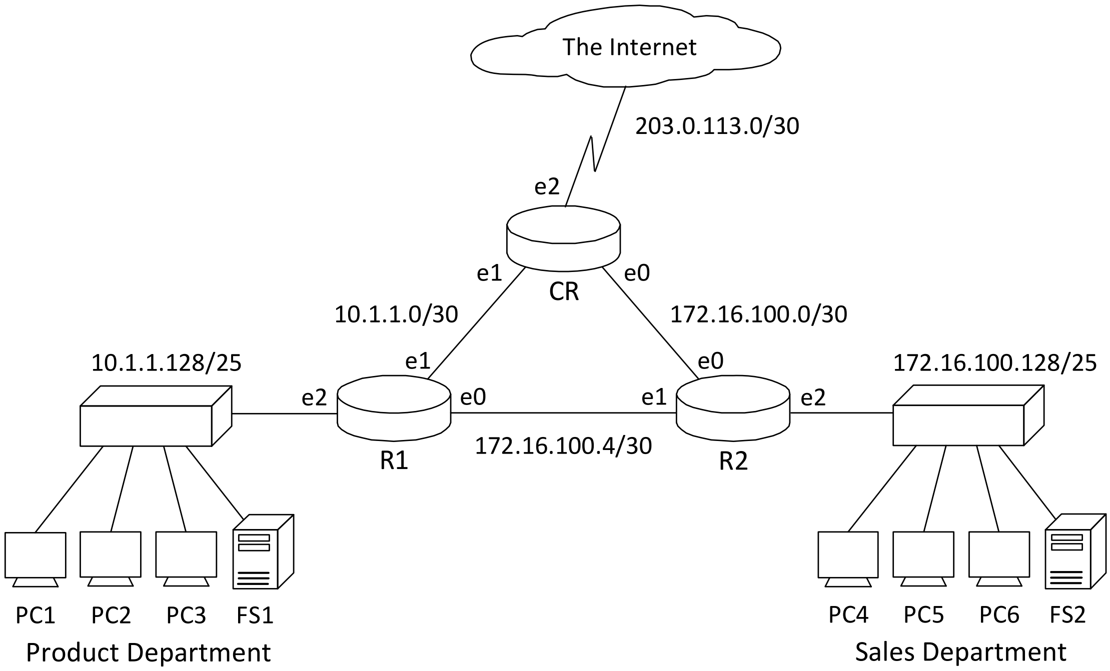


Figure 1 Company V's internal network

Table1 Addressing scheme of company V

Device	Interface	IP Address	Subnet Mask
Core Router	e0	172.16.100.1	255.255.255.252
	e1	10.1.1.1	255.255.255.252
	e2	203.0.113.2	255.255.255.252
Router 1	e0	172.16.100.5	255.255.255.252
	e1	10.1.1.2	255.255.255.252
	e2	10.1.1.129	255.255.255.128
Router 2	e0	172.16.100.2	255.255.255.252
	e1	172.16.100.6	255.255.255.252
	e2	172.16.100.129	255.255.255.128

Subquestion 1

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

The administrator assigned the IP addresses and configured static routing protocol on each router to get connection between R1's LAN, R2's LAN and the Internet. After completing the network configuration, the administrator starts to test the connection among the three routers. He used "ping" commands at the command prompt on PC1, PC2, PC4 and PC5 to ping to an address on the Internet; all the "ping" commands are successful. Tables 2, 3 and 4 show the routing information of the routers. Note that "any/any" for "Destination Network" means a default gateway.

Table 2 Routing Information of CR

Destination Network	Interface	Next hop/GW
172.16.100.0/30	e0	A
172.16.100.128/25	e0	172.16.100.2
10.1.1.0/30	e1	directly connected
10.1.1.128/25	e1	10.1.1.2
203.0.113.0/30	e2	directly connected
any/any	e2	203.0.113.1

Table 3 Routing Information of R1

Destination Network	Interface	Next hop/GW
172.16.100.4/30	e0	directly connected
172.16.100.128/25	e0	B
any/any	e1	10.1.1.1
10.1.1.0/30	e1	directly connected
10.1.1.128/25	e2	directly connected

Table 4 Routing Information of R2

Destination Network	Interface	Next _hop/GW
172.16.100.0/30	e0	directly connected
C	e0	172.16.100.1
10.1.1.128/25	e1	172.16.100.5
172.16.100.4/30	e1	directly connected
172.16.100.128/25	e2	directly connected

Answer group for A and B

a) 10.1.1.0	b) 10.1.1.1	c) 10.1.1.2
d) 172.16.100.0	e) 172.16.100.1	f) 172.16.100.2
g) 172.16.100.5	h) 172.16.100.6	i) directly connected

Answer group for C

a) 10.1.1.0/30
b) 10.1.1.128/25
c) 172.16.100.0/30
d) 172.16.100.4/30
e) 172.16.100.128/25
f) 203.0.113.0/30
g) any/any

Subquestion 2

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

After the setup, all hosts from R1's LAN and R2's LAN can access the Internet and also share their resources with each other except for PC3. The administrator checked the status of PC3 and found that it was not able to connect to a wired network. Pinging the loopback address is successful, but the gateway cannot be reached. On the network switch, all the interface lights are on except for the interface connected to PC3. The LED on the network card of PC3 is off. The port of the switch connected to PC3 is operational when the other PCs are connected. He considers that $\boxed{\text{D}}$ is the most likely cause of this problem, and resolved this problem successfully.

Answer group for D

a) the gateway requires fixing
b) the Internet connection is down
c) the network cable is faulty
d) the network switch is faulty
e) the PC has an incorrect IP address for the DNS server

Subquestion 3

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

The administrator is configuring the firewall functions on the three routers to satisfy the following two requirements:

(1) File server FS1 in the product department should be accessible from both the product department and sales department. Meanwhile, file server FS2 in the sales department must be accessible only from the sales department to protect customer information. Each file server provides its service with SMB protocol.

(2) The administrator manages the entire network. He accesses the network from his laptop using SSH (remote login protocol). He usually works in the product department office, and occasionally works in the sales department office as well.

Tables 5, 6 and 7 show the firewall rules on routers CR, R1 and R2.

When the firewall function receives a packet, it scans the rule table in order from top to bottom. When it finds a rule that matches the conditions, it terminates the scan and applies that rule to the packet. Note that because this firewall function understands the TCP stream, the related packets that have been allowed once (e.g., packets from the same TCP session) remain allowed even if there is no explicit rule. In addition, the packets from each router itself are implicitly allowed.

After the setup, he received a complaint from one of the employees claiming that the firewall does not seem to be working. He found the misconfigured rule on the rule table of [E], and then he [F].

Table 5 Firewall rule table of CR

Rule No	Source	Destination	Destination Service	Action
1	10.1.1.128/25	any	any	Allow
2	172.16.100.128/25	any	any	Allow
3	any	any	any	Deny

Table 6 Firewall rule table of R1

Rule No	Source	Destination	Destination Service	Action
1	172.16.100.128/25	FS1	SMB/tcp	Allow
2	172.16.100.128/25	10.1.1.128/25	SSH/tcp	Allow
3	10.1.1.128/25	any	any	Allow
4	any	any	any	Deny

Table 7 Firewall rule table of R2

Rule No	Source	Destination	Destination Service	Action
1	10.1.1.128/25	FS2	SMB/tcp	Allow
2	10.1.1.128/25	172.16.100.128/25	SSH/tcp	Allow
3	172.16.100.128/25	any	any	Allow
4	any	any	any	Deny

Answer group for E

a) CR b) R1 c) R2

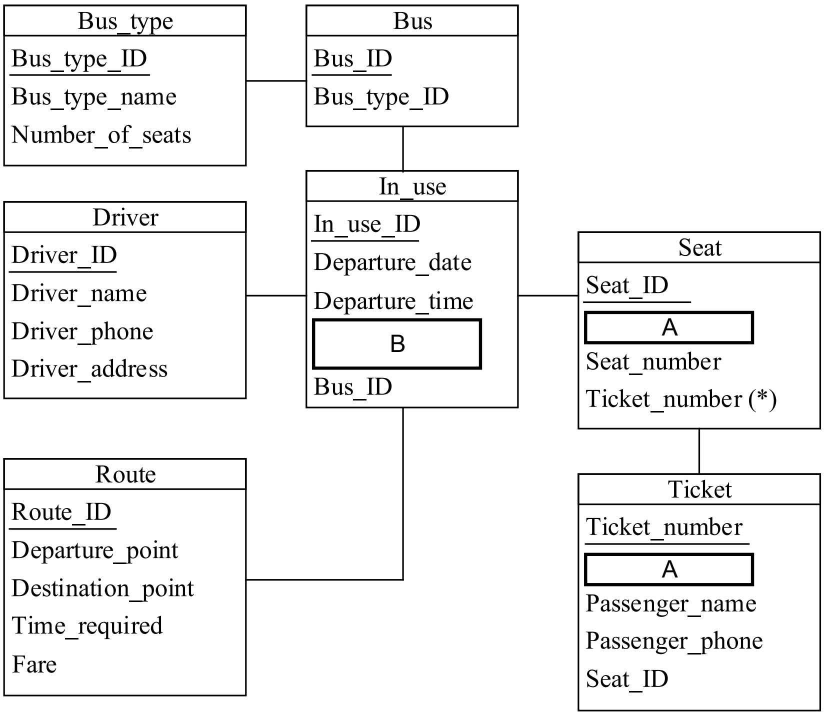
Answer group for F

a) moved it to the top of the table
b) removed it
c) swapped it for the rule just above it
d) swapped it for the rule just below it

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

Company W operates long-distance buses. Each bus has one departure point and one arrival point, and does not stop along the way. Company W is planning to develop the bus ticket reservation system.

- The system is designed for three types of users: Passenger, Manager, and Driver.
- Company W has several types of buses. The number of seats is fixed for each type.
- The departure point, arrival point, required time, and fare are determined for each route.
- Two months before the departure date, company W decides the routes to follow and buses to be used according to the schedule. At this time, all seats are registered as vacant seats.



Note: An underline indicates the primary key.
(*) indicates that the entity value can be NULL.

Figure 1 Ticket reservation system database

The system has three processes: 1) Driver assignment, 2) Driver confirmation, and 3) Ticket booking.

Here, DFD is deployed to present the data flow in the system. Figure 2 shows the context diagram of the system. Figure 3 shows the level 1 DFD of the Driver assignment and Driver confirmation processes.

1

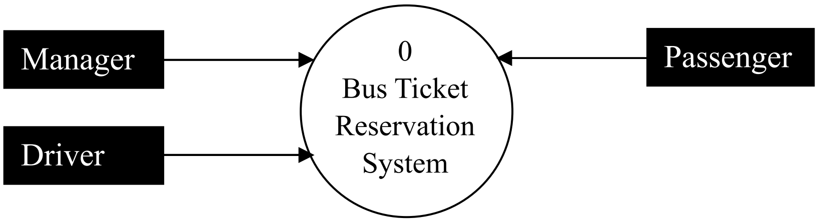


Figure 2 Context Diagram

[For the Manager and Driver]

In the system, the manager has the responsibility of allocating a bus to each route and assigning a driver to the bus. After the manager assigns the bus driver, notification is sent to the driver. The driver has to confirm the assignment to complete the assignment process.

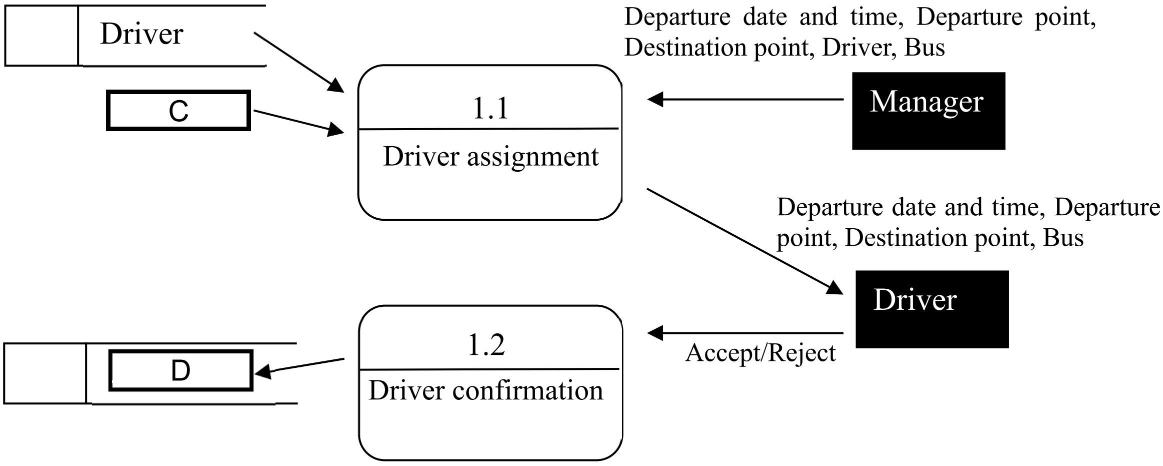

Driver
C
Departure date and time, Departure point, Destination point, Driver, Bus
Manager
1.1
Driver assignment
Departure date and time, Departure point, Destination point, Bus
Driver
1.2
Driver confirmation
Accept/Reject
D

Figure 3 Level 1 DFD of Driver assignment and Driver confirmation processes

Subquestion 1

From the answer groups below, select the correct answer to be inserted in each blank □ in Figure 1 and Figure 3.

Answer group for A

a) Bus_type_ID　　b) Driver_ID　　c) In_use_ID

d) Route_ID　　e) Seat_ID

Answer group for B

a) Driver_ID (*)

b) Driver_ID (*)
Route_ID

c) Driver_ID (*)
Route_ID
Seat_ID

d) Driver_ID (*)
Route_ID
Ticket_number

e) Seat_ID
Ticket_number

Answer group for C

a) Bus
b) Bus, Bus_type
c) Bus, Bus_type, In_use
d) Bus, Bus_type, In_use, Route, Seat
e) Bus, Bus_type, In_use, Route, Seat, Ticket
f) Bus, Bus_type, Route
g) Bus, Bus_type, Route, Seat
h) Bus, Bus_type, Route, Seat, Ticket

Answer group for D

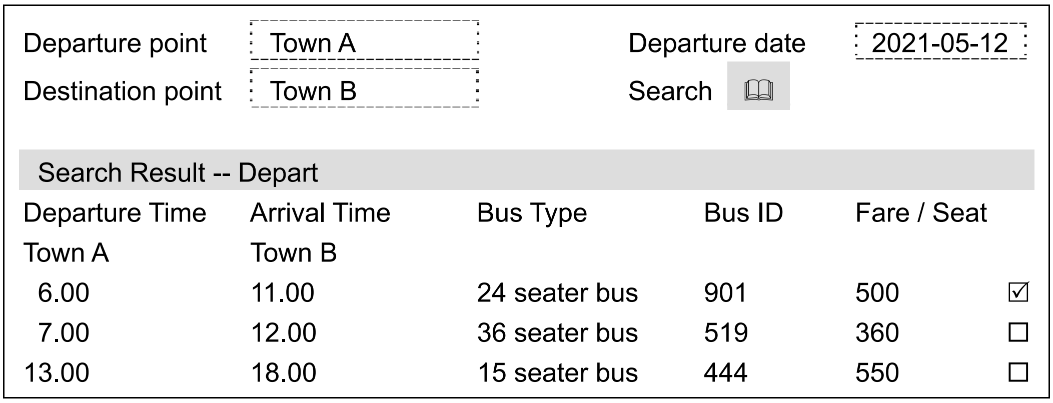
a) Bus b) Bus_type c) Driver d) In_use
e) Route f) Seat g) Ticket

Subquestion 2

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

[For the passenger]

Booking is performed by the passenger. The passenger opens the bus ticket reservation application to book a bus seat. Company W has many buses for many routes. The ticket booking can be accomplished by the passenger via a reservation application. The passenger is requested to choose the departure date and the departure and destination points. Then the system displays the bus schedule, the bus type, and the fare on that date, as shown in Figure 4. The passenger can make one reservation via the booking screen.



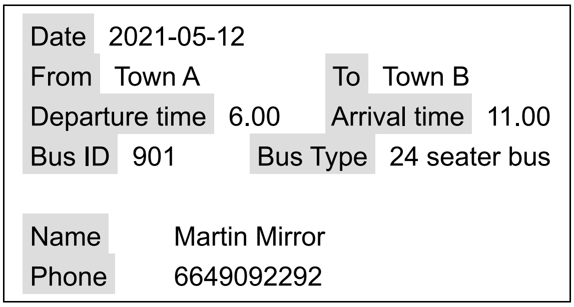
Departure point: Town A
Destination point: Town B
Departure date: 2021-05-12
Search 📖

Search Result -- Depart

Departure Time Town A	Arrival Time Town B	Bus Type	Bus ID	Fare / Seat	
6.00	11.00	24 seater bus	901	500	☑
7.00	12.00	36 seater bus	519	360	□
13.00	18.00	15 seater bus	444	550	□

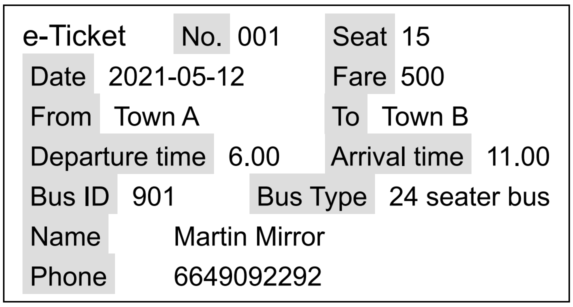
Figure 4 Screen: Booking

After the passenger has selected the preferred time and bus-type available in the booking screen, another screen appears for the passenger to enter the passenger detail(s), as shown in Figure 5. The ticket for each passenger is printed as an e-ticket, as illustrated in Figure 6.



Date	2021-05-12		
From	Town A	To	Town B
Departure time	6.00	Arrival time	11.00
Bus ID	901	Bus Type	24 seater bus
Name	Martin Mirror		
Phone	6649092292		

Figure 5 Screen: Passenger details



e-Ticket	No. 001	Seat	15
Date	2021-05-12	Fare	500
From	Town A	To	Town B
Departure time	6.00	Arrival time	11.00
Bus ID	901	Bus Type	24 seater bus
Name	Martin Mirror		
Phone	6649092292		

Figure 6 e-ticket

Figure 7 shows the level 1 DFD of the Ticket booking process.

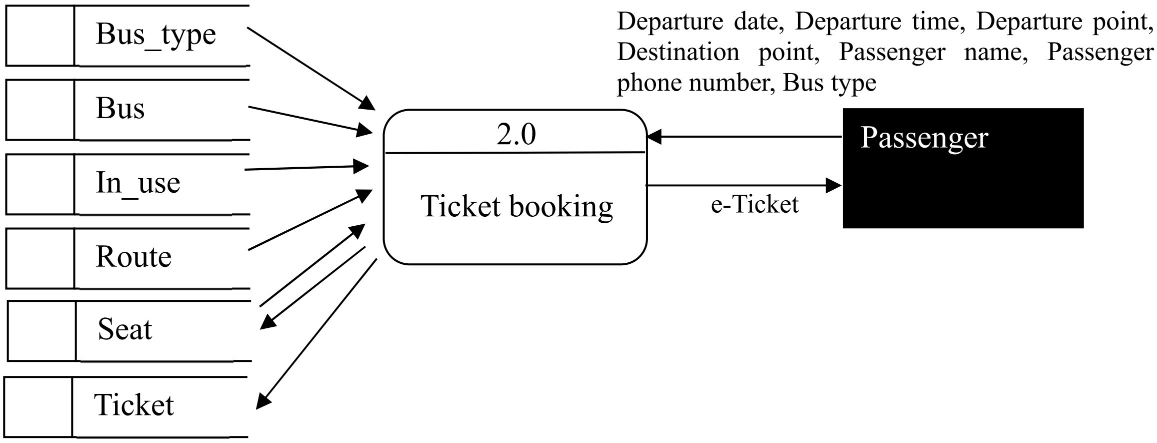

Bus_type
Bus
In_use
Route
Seat
Ticket
2.0
Ticket booking
Departure date, Departure time, Departure point, Destination point, Passenger name, Passenger phone number, Bus type
Passenger
e-Ticket

Figure 7 Level 1 DFD of Ticket booking process

The details of the Generate list process and Reservation process are as follows.

Generate list process

(1) Search In_use table using the departure date, departure point, and destination point in chronological order.

(2) Search Seat_table using the obtained key. If a vacant seat is available, add the departure time, arrival time, and bus information in the list.

(3) If the list is empty, generate an error message.

Reservation process

(1) Start exclusive control, and generate a unique number and use it as the ticket number.

(2) Search Seat_table using [E] as the key to search for a vacant seat.

(3) If a vacant seat is available, [F]; otherwise, generate an error message.

(4) Terminate exclusive control.

Figure 8 shows the activity diagram of the Ticket booking process.

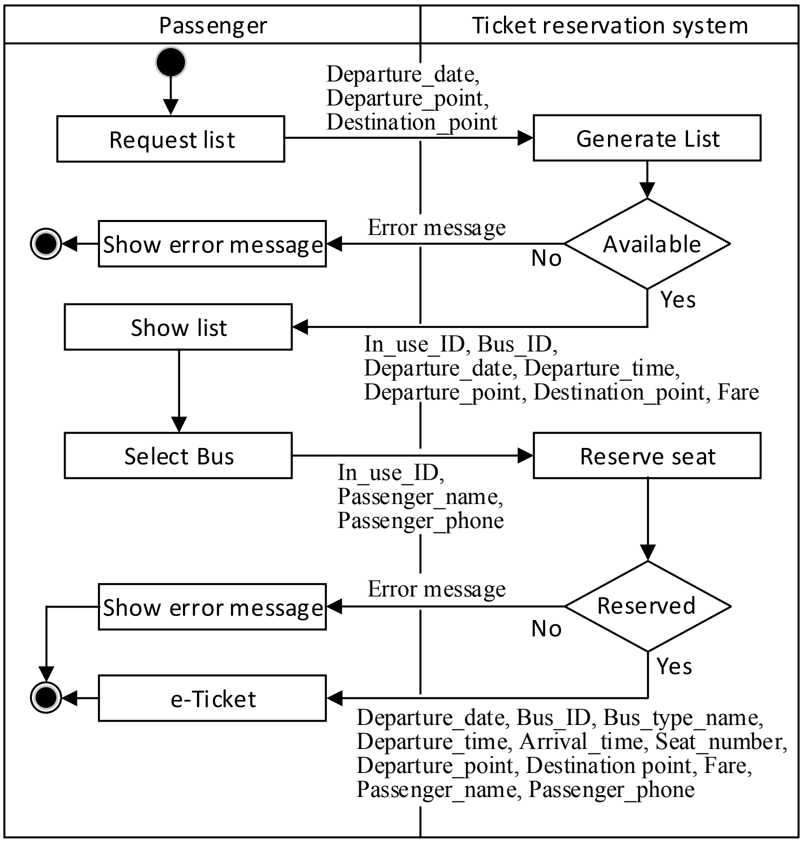


Figure 8 Activity diagram of Ticket booking process

Answer group for E

a) Bus_ID
b) Bus_type_ID
c) Driver_ID
d) In_use_ID
e) Route_ID
f) Seat_ID
g) Ticket_number

Answer group for F

a) insert new data to Ticket table.
b) insert new data to Ticket table and insert new data to Seat table.
c) insert new data to Ticket table and set Ticket_number to Ticket_number in Seat table.
d) update Ticket table.
e) update Ticket table and insert new data to Seat table.

Q6. Read the following description of social network analysis, and then answer Subquestions 1 through 3.

[Program Description]

In social media marketing, influence is one of the most important factors. The most basic measure of influence is the number of "friends". Although this can measure popularity, it does not consider "reach". One of the more complex measures of influence is Closeness Centrality, which determines the "closeness" of an influencer to all the other users in the social network. This program calculates an index of Closeness Centrality (called Closeness Centrality Index (CCI)).

(1) Closeness Centrality uses the shortest path distances from one node to all other nodes on the network. The distance between two nodes is determined by counting the number of edges or links between the two. For example, in Figure 1, from node 10, node 6 has a distance of 1, and node 4 has a distance of 2.

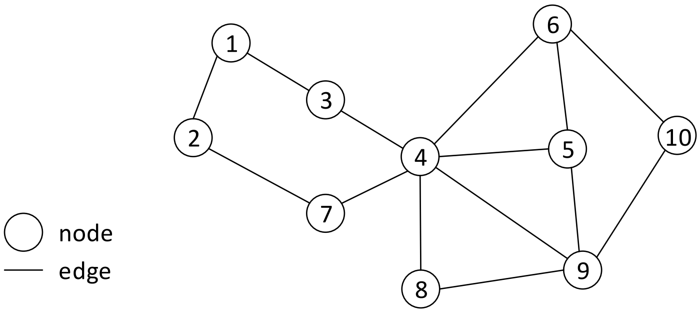


Figure 1 Social network example

(2) The CCI value is from 0 to 1, where the node with the highest value is the closest to all other nodes. For a graph with n number of nodes, the CCI value for each node is computed by the number of the other nodes divided by the sum of the distances to the other nodes.

$$CCI = \frac{n-1}{\sum(distance\ to\ other\ nodes)}$$

(3) The social network illustrated in Figure 1 produces the distance matrix shown in Table 1. For example, the last row of Table 1 indicates that node 10 has distances 4, 4, 3, 2, 2, 1, 3, 2, and 1 to nodes 1 to 9, and the CCI value of node 10 is 9 / 22 = 0.41. Here, CCI is rounded to two decimal places.

Table 1 Distance matrix of social network illustrated in Figure 1

Node	1	2	3	4	5	6	7	8	9	10	CCI
1	0	1	1	2	3	3	2	3	3	4	0.41
2	1	0	2	2	3	3	1	3	3	4	B
3	1	2	0	1	2	2	2	2	2	3	0.53
4	2	2	1	0	1	1	1	1	1	2	0.75
5	3	3	2	1	0	1	2	2	1	2	0.53
6	3	3	2	1	1	A		2	2	1	0.53
7	2	1	2	1	2			2	2	3	0.53
8	3	3	2	1	2	2	2	0	1	2	0.50
9	3	3	2	1	1	2	2	1	0	1	0.56
10	4	4	3	2	2	1	3	2	1	0	0.41

Subquestion 1

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

Answer group for A

a)

0	2
2	0

b)

0	3
2	0

c)

0	2
3	0

d)

0	3
3	0

Answer group for B

a) 0.36　　b) 0.41　　c) 0.45　　d) 0.50

Subquestion 2

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

[Program Description]

(1) The program uses breadth-first search/traversal algorithm to build the distance matrix. The algorithm is as follows.

- o Set the distance of the start node.
- o Put the start node into the queue.
- o While the queue is not empty, repeat the following:
  - o Get a node v from the queue.
  - o For each node w that is a neighbor of node v, perform the following:
    - o If node w has not yet been visited, calculate the distance between the start node and node w. Thereafter, add node w to the queue.

(2) The class `Graph` represents a graph network data structure with nodes and edges and has the following methods. Graph `G` is defined in the function `main` (not shown in the program). Here, graph `G` is undirected and connected (there is a path between every pair of nodes), and the number of nodes ≥ 2.

Method	Description
`number_of_nodes()`	Returns the number of nodes
`neighbor_nodes(x)`	Returns an array of the neighbor nodes of node `x`

(3) The class `Queue` represents a queue data structure, and has the following methods. Queue `Q` is defined in the function `bfs`.

Method	Description
`empty()`	Returns `true` if the queue is empty; otherwise, returns `false`
`enqueue(x)`	Enqueues `x` to the queue
`dequeue()`	Removes and returns the head element from the queue

(4) The program has the following functions.

Function	Description
`bfs(G, s)`	Performs breath-first search/traversal from node `s` on graph `G`, and returns an array of the distances from node `s` to the other nodes on graph `G`
`distanceMatrix(G)`	Builds and returns a distance matrix for graph `G`
`computeCci(G)`	Computes CCI of the nodes on graph `G`

(5) The program uses the following external functions.

Function	Description
`uninitialized(x)`	Returns `true` if variable `x` is uninitialized; otherwise returns `false`
`length(x)`	Returns the number of elements of array `x`

(6) The indexes of all arrays used in the program start at 1.

(7) The function `main` (not shown in the program) sets up the graph data in `G` and then calls the function `computeCci`.

[Program]

```
FUNCTION: computeCci(Graph: G) {
   INT: i, j, n, sum
   n ← G.number_of_nodes();
   INT: matrix[n][n]
   FLOAT: cci[n]
   matrix ← distanceMatrix(G);
   FOR (i ← 1; i ≤ n; i ← i + 1) {
      sum ← 0;
      FOR (j ← 1; j ≤ n; j ← j + 1) {
         sum ← sum + matrix[i][j];
      }
      cci[i] ← (FLOAT)(n - 1) / (FLOAT)sum;
   }
   return(cci);
}

FUNCTION: distanceMatrix(Graph: G) {
   INT: i, j, n
   n ← G.number_of_nodes();
   INT: d[n], matrix[n][n]
   FOR (i ← 1; i ≤ n; i ← i + 1) {
      d ← bfs(G, i);
      FOR (j ← 1; j ≤ n; j ← j + 1) {
         matrix[i][j] ← d[j];
      }
   }
   return(matrix);
}

FUNCTION: bfs(Graph: G, INT: s) {
   Queue: Q
   INT: i, array_w[], n, v, w
   n ← G.number_of_nodes();
   INT: d[n]
   [      C      ];
   Q.enqueue(s);
   WHILE (not Q.empty()) {
      v ← Q.dequeue();    /* ← α */
      array_w ← G.neighbor_nodes(v);
      FOR (i ← 1; i ≤ length(array_w); i ← i + 1) {
         w ← array_w[i];    /* ← β */
         IF (uninitialized(d[w])) {
            [      D      ];
            Q.enqueue(w);    /* ← γ */
         }
      }
   }
   return(d);
}
```

Answer group for C

a) `d[1] ← 0`	b) `d[1] ← 1`	c) `d[1] ← n`
d) `d[s] ← 0`	e) `d[s] ← 1`	f) `d[s] ← n`

Answer group for D

a) `d[v] ← d[s]`	b) `d[v] ← d[s] + 1`
c) `d[v] ← d[w]`	d) `d[v] ← d[w] + 1`
e) `d[w] ← d[s]`	f) `d[w] ← d[s] + 1`
g) `d[w] ← d[v]`	h) `d[w] ← d[v] + 1`

Subquestion 3

From the answer group below, select the correct answer to be inserted in each blank [   ] in the following description. It is assumed that the correct answers are inserted in blanks [C] and [D].

When the program is executed by providing graph `G` shown in Figure 1, function `bfs` is called 10 times.

When function `bfs` is called the first time, the line pointed out by `/* ← α */` is executed [E] times, the line pointed out by `/* ← β */` is executed [F] times, and the line pointed out by `/* ← γ */` is executed [G] times.

Answer group for E through G

a) 8	b) 9	c) 10	d) 14
e) 18	f) 22	g) 28	h) 36

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

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

A polynomial with integer coefficients of degree `n` can be represented by an integer array of length `n`+1. For example, the array `{7, -1, 0, 5}` represents the polynomial $5x^3$-x+7. In this representation, the leading coefficient, which is the last element of the array, is always non-zero except for a zero polynomial represented by `{0}`.

[Program Description]

(1) The function `display_poly` prints a polynomial of degree `deg` that is represented by the array `pol[]` to the standard output. For each term of the polynomial, `display_poly` prints the coefficient, `*` (the multiplication sign), `x`, `^` (the power sign), and the degree of the term in this order, except for the following cases (if multiple cases match, the first one is applied).

(i) Print `0` for the zero polynomial.

(ii) Print nothing for a term if the coefficient of the term is 0.

(iii) Print `+` before the positive coefficient of a term except for the term with the highest degree.

(iv) If the coefficient of a term is 1 or `-`1 and the degree of the term is not 0, omit `1` and the following `*`.

(v) If the degree of a term is 1, omit `^` and the degree.

(vi) If the degree of a term is 0, do not print `*`, `x`, `^`, and the degree.

(2) Examples of input and output of the function `display_poly` is shown in Table 1.

Table 1 Examples of input and output of function `display_poly`

Input		Output
`pol[]`	`deg`	
`{0}`	0	`0`
`{1}`	0	`1`
`{-1, -1}`	1	`-x-1`
`{1, 0, 1}`	2	`x^2+1`
`{7, -1, 0, 5}`	3	`5*x^3-x+7`
`{0, -6, 12, 1, -3}`	4	`-3*x^4+x^3+12*x^2-6*x`

(3) The function iterates from the term with the highest degree to the lowest. To output one term in a polynomial in each iteration, the following three functions are called.

(i) `void print_coef(int coef, int pow, int deg);`
Prints coefficient `coef` of a term with degree `pow` of a polynomial of degree `deg`. For example, if the polynomial `-3*x^4+x^3+12*x^2-6*x` is given, it prints `-3`, `+`, `+12`, and `-6` for each term.

(ii) `void print_star(int coef, int pow);`
Prints `*` for a term with coefficient `coef` and degree `pow`. For example, it prints three `*`'s for the polynomial `-3*x^4+x^3+12*x^2-6*x`.

(iii) `void print_xpow(int pow);`
Prints `x`, `^`, and degree `pow` of a term. For example, it prints `x^4`, `x^3`, `x^2`, and `x` for each term of the polynomial `-3*x^4+x^3+12*x^2-6*x`.

[Program]

```
#include <stdio.h>

/* prints polynomial */
void display_poly(int[], int);
/* prints coefficient of term in polynomial*/
void print_coef(int, int, int);
/* prints star of term in polynomial */
void print_star(int, int);
/* prints x and power of term in polynomial*/
void print_xpow(int);

void display_poly(int pol[], int deg) {
    int i;
    if (deg == 0) {
        printf([    A    ]);
    } else {
        for (i = deg; i >= 0; i--) {
            if (pol[i] != 0) {
                print_coef([    B    ], deg);
                print_star([    B    ]);
                print_xpow(i);
            }
        }
    }
    printf("\n");
}
```

```
void print_coef(int coef, int pow, int deg) {
    if (coef == 1) {
        if (pow == 0) {
            printf([     C     ]);
        } else if (pow < deg) {
            printf("+");
        }
    } else if (coef == -1) {
        printf([     D     ]);
        if (pow == 0) {
            printf("1");
        }
    } else {
        if (coef > 1 && pow < deg) {
            printf("+");
        }
        printf("%d", coef);
    }
}

void print_star(int coef, int pow) {
    if ([     E     ] && coef != 1 && coef != -1) {
        printf("*");
    }
}

void print_xpow(int pow) {
    if (pow > 0) {
        printf("x");
    }
    if (pow > 1) {
        printf("^%d", pow);
    }
}

int main() {
    int pol[5] = {11, 1, -1, 0, -7};

    display_poly(pol, 4);
    return 0;
}
```

Subquestion 1

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

Answer group for A

a) `"%d", 0`
b) `"%d", pol[0]`
c) `"+%d", pol[deg]`
d) `"-%d", pol[deg]`

Answer group for B

a) `*pol, deg - i`
b) `pol[0], deg - i`
c) `pol[i], deg - i`
d) `pol[i], i`
e) `pol[i + 1], deg - i`
f) `pol[i + 1], i`

Answer group for C and D

a) `""`
b) `"+"`
c) `"+1"`
d) `"+x"`
e) `"-"`
f) `"-1"`
g) `"-x"`
h) `"1"`

Answer group for E

a) `pow == 0`
b) `pow > 0`
c) `pow != 1`
d) `pow > 1`

Subquestion 2

From the answer group below, select the correct answer to be inserted in each blank ☐ in the following description. It is assumed that the correct answers are inserted in all of the blanks A to E.

(1) The function `print_xpow` is modified as follows:

```
void print_xpow(int pow) {
    if (pow > 0) {
        printf("x^%d", pow);
    }
}
```

When the program is executed, the program outputs F.

(2) Inadvertently, function main is modified as follows:

```
int main() {
    int pol[3] = {0, 0, 0};

    display_poly(pol, 2);
    return 0;
}
```

This modification violates the rule that the leading coefficient, which is the last element of the array, is always non-zero except for a zero polynomial represented by {0}. In this case, when the program is executed, the program outputs [G]. Here, ignore the modification in (1) above.

Answer group for F

a) -7*x^4+0*x^3-1*x^2+1*x^1+11

b) -7*x^4+0*x^3-1*x^2+1*x^1+11*x^0

c) -7*x^4-x^2+x+11

d) -7*x^4-x^2+x^1+11

e) -7*x^4-x^2+x^1+11*x^0

Answer group for G

a) 0

b) 00

c) 0*x^2

d) 0*x^2+0

e) a blank line ('\n' only)

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

[Program Description]

This program is a prototype component for implementing the forward and backward buttons in a web browser-like application. As a user visits different web pages using the application, the *visit history*, a list of visited web pages, is maintained. The planned application functionality is that the user can go backward by clicking the back button and revisit previously visited web pages, and that the user can also sequentially return to the most recently visited web page by clicking the forward button. This is a partial implementation to assist the application to traverse the visit history.

The `WebPage` class (Program 1) denotes one visited web page at a particular time. The class has following members.

(1) The `WebPage` constructor creates a `WebPage` with the URL and timestamp parameters.
(2) The `url` field denotes the visited web page represented by a URL string.
(3) The `date` field represents the timestamp of the web page visit.
(4) The `toString` method returns the text representation of this object.

[Program 1]

```
public class WebPage {
   private final String url;
   private final String date;

   WebPage(String url, String date) {
      this.url = url;
      this.date = date;
   }

   public [   A   ] toString() {
      return "URL: " + url + "\tAccessed: " + date;
   }
}
```

The `WebVisitHistory` class (Program 2) is a partial implementation for managing the visit history. The class has the following members:

(1) `Node` is a nested class representing each point in the visit history with the following members.
   (i) The `webPage` field represents the visited web page.
   (ii) The `previous` field is a reference to the `Node` representing the previously visited web page immediately *before* the web page represented by this `Node`. If there is no such previous web page, it refers to the `oldestMarker` field (see below).

(iii) The next field is a reference to the Node representing the visited web page immediately *after* the web page represented by this Node. If there is no such next web page, it refers to the newestMarker field (see below).

(2) The oldestMarker and newestMarker fields are two sentinel constants to mark the boundaries of the linked list of the visited web pages. oldestMarker refers to the first visited web page, and newestMarker refers to the last visited one.

(3) The length field keeps count of the visited web pages, and the length method returns the field value.

(4) The WebVisitHistory constructor initializes an empty visit history with only the boundary markers being adjacent to each other.

(5) The isEmpty method returns true if there are no web pages in the visit history, or false, otherwise.

(6) The add method adds the specified WebPage to the visit history as the last web page visited.

(7) The historyIterator method returns a *history iterator* to iterate the visit history data.

(8) The HistoryIterator inner class implements the java.util.ListIterator interface that supports the bidirectional iterations of the visit history data. An iterator maintains its *position* that always lies between Nodes representing visited web pages or the sentinels. The initial position for the iteration is set to the end of the visit history (i.e., no next web page) so that it is possible to start iterating the visit history backward. The class has the following methods.

(i) The hasNext method returns true if there are any web pages in the forward direction, or false, otherwise.

(ii) The hasPrevious method returns true if there are any web pages in the backward direction, or false, otherwise.

(iii) The next method returns the next web page and advances the position. If there is no next one, a NoSuchElementException is thrown.

(iv) The previous method returns the previous web page and moves the position backward. If there is no previous one, a NoSuchElementException is thrown.

(v) The nextIndex method returns the index of the web page that would be returned by a subsequent call to the next method, or the length value if this iterator is at the end of the visit history.

(iv) The previousIndex method returns the index of the web page that would be returned by a subsequent call to the previous method, or -1 if this iterator is at the beginning of the visit history.

(vii) The remove, add, and set methods are unsupported.

[Program 2]

```
import java.util.ListIterator;
import java.util.NoSuchElementException;

public class WebVisitHistory {
   private final Node oldestMarker, newestMarker;
   private int length;

   public WebVisitHistory() {
      oldestMarker = new Node(null);
      newestMarker = new Node(null);
      oldestMarker.next = newestMarker;
      newestMarker.previous = oldestMarker;
   }

   private static class Node {
      private final WebPage webPage;
      private Node previous, next;

      Node(WebPage webPage) {
         this.webPage = webPage;
      }
   }

   public boolean isEmpty() {
      return length() == 0;
   }

   public int length() {
      return length;
   }

   public void add(WebPage webPage) {
      Node newNode = new Node(webPage);

      Node backup = newestMarker.previous;
      newestMarker.previous = [   B   ];
      backup.next = [   B   ];
      newNode.next = newestMarker;
      newNode.previous = [   C   ];
      ++length;
   }
   public ListIterator<WebPage> historyIterator() {
      return new HistoryIterator();
   }
   private class HistoryIterator implements ListIterator<WebPage> {
      private Node current = newestMarker;
      private int index = length();

      public boolean hasNext() {
         return current != newestMarker;
      }

      public boolean hasPrevious() {
         return current.previous != oldestMarker;
      }

      public WebPage next() {
         if (!hasNext()) {
            throw new NoSuchElementException();
         }
         WebPage webPage = current.webPage;
         current = [    D    ];
         index++;
         return webPage;
      }

      public WebPage previous() {
         if (!hasPrevious()) {
            throw new NoSuchElementException();
         }
         current = [    E    ];
         index--;
         return current.webPage;
      }

      public int nextIndex() {
         return [    F    ];
      }

      public int previousIndex() {
         return index - 1;
      }

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

      public void set(WebPage webPage) {
         throw new UnsupportedOperationException("unsupported");
      }

      public void add(WebPage webPage) {
         throw new UnsupportedOperationException("unsupported");
      }
   }
}
```

The `Tester` class (Program 3) tests the implementation of iterations and generates the following output:

```
Visited Pages: 4
Going backward:
index: 3 URL: https://example.com/p4.html  Accessed: 2021-01-01 00:00:30
index: 2 URL: https://example.com/p3.html  Accessed: 2021-01-01 00:00:20
index: 1 URL: https://example.com/p2.html  Accessed: 2021-01-01 00:00:10
index: 0 URL: https://example.com/p1.html  Accessed: 2021-01-01 00:00:00
Going forward:
index: 0 URL: https://example.com/p1.html  Accessed: 2021-01-01 00:00:00
index: 1 URL: https://example.com/p2.html  Accessed: 2021-01-01 00:00:10
index: 2 URL: https://example.com/p3.html  Accessed: 2021-01-01 00:00:20
index: 3 URL: https://example.com/p4.html  Accessed: 2021-01-01 00:00:30
```

[Program 3]

```
import java.util.ListIterator;

public class Tester {
   private static final String URL_TEMPLATE
                                  = "https://example.com/p%d.html";
   private static final int N_PAGES = 4;

   public static void main(String[] args) {
      WebVisitHistory history = new WebVisitHistory();
      for (int i = 0; i < N_PAGES; ++i) {
         int t = i * 10;
         history.add(new WebPage(String.format(URL_TEMPLATE, i + 1),
           String.format("2021-01-01 00:%02d:%02d", t / 60, t % 60)));
      }

      System.out.printf("Visited Pages: %d%n", history.length());
      ListIterator<WebPage> historyIterator =
                                            history.historyIterator();
      if (historyIterator.nextIndex() != history.length()) {
         throw new AssertionError();
      }
      System.out.println("Going backward:");
      while (historyIterator.hasPrevious()) {
         System.out.printf("index: %d %s%n",
              historyIterator.previousIndex(),
              historyIterator.previous());
      }
      if (historyIterator.previousIndex() != -1) {
         throw new AssertionError();
      }
      System.out.println("Going forward:");
      while (historyIterator.hasNext()) {
         System.out.printf("index: %d %s%n",
               historyIterator.nextIndex(), historyIterator.next());
      }
   }
}
```

Subquestion 1

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

Answer group for A

a) `Date` b) `Object` c) `String`

d) `URL` e) `WebPage`

Answer group for B and C

a) `backup` b) `backup.previous`

c) `new Node(webPage)` d) `newestMarker.previous`

e) `newNode` f) `oldestMarker.next`

Answer group for D and E

a) `current.next` b) `current.next.next`

c) `current.previous` d) `current.previous.previous`

e) `newestMarker.previous` f) `oldestMarker.next`

Answer group for F

a) `++index` b) `index`

c) `index + 1` d) `index == length() ? index : index + 1`

e) `index++`

Subquestion 2

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

As the next step of prototyping, new method `removeRestAfter` (Program 4) is added to the `WebVisitHistory` class. The method removes all `Node`s after the `Node` having the specified `WebPage` to allow a new visit history path to be started after the `WebPage`. If there is no specified `WebPage` found on the visit history, a `NoSuchElementException` is thrown. Removing any visit history data makes any existing history iterators (`ListIterator`s) void, and new history iterators must be obtained by calling the `historyIterator` method after the removal.

[Program 4]

```
public void removeRestAfter(WebPage webPage) {
   int count = 0;
   for (Node node = newestMarker.previous;
        node != oldestMarker; node = node.previous) {
      if (node.webPage == webPage) {
         node.next =    G    ;
         newestMarker.previous = node;
         length -= count;
         return;
      }
      count++;
   }
   throw new NoSuchElementException();
}
```

Answer group for G

a) `newestMarker`	b) `newestMarker.previous`
c) `node`	d) `node.next`
e) `node.previous`	f) `oldestMarker`
g) `oldestMarker.next`	