# October 2014

# Fundamental IT Engineer Examination (Afternoon)

**Questions must be answered in accordance with the following:**

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

**Instructions:**

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

2. Mark your examinee information and test answers in accordance with the instructions below. Your answer will not be graded if you do not mark properly. Do not mark or write on the answer sheet outside of the prescribed places.
   (1) **Examinee Number**
   Write your examinee number in the space provided, and mark the appropriate space below each digit.
   (2) **Date of Birth**
   Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.
   (3) **Question Selection**
   For **Q7** and **Q8**, mark the ⓢ of the question you select to answer in the "Selection Column" on your answer sheet.
   (4) **Answers**
   Mark your answers as shown in the following sample question.

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

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

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

   [Sample Answer]

| Sample | ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ ⓙ |
|---|---|

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

## Notations used for pseudo-language

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

[Declaration, comment, and process]

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

[Logical constants]

`true, false`

( continued on next page )

[Operators and their priorities]

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

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

---

## Notations used for E-R diagrams

In questions that use E-R diagrams, the following notations are used unless otherwise stated.

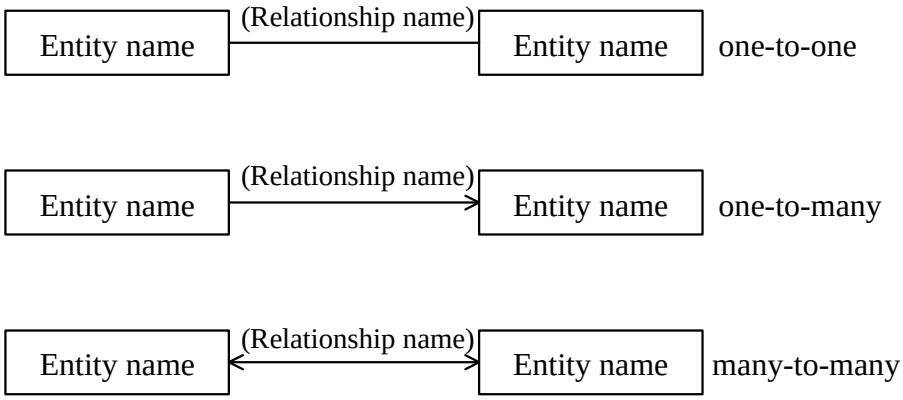



Figure Notations used for entities and relationships

1. Entities are represented by rectangles.
2. Entity names are indicated within the rectangles.
3. A relationship between entities is represented by a line.
   The relationship name is indicated at the side of the line as “(Relationship name)”.
   The relationship name can be omitted.
4. A “one-to-one” relationship is represented by a line.
   A “one-to-many” relationship is represented by a line with an arrow pointing towards the “many” side.
   A “many-to-many” relationship is represented by a line with an arrow on both ends.

---

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

**Q1.** Read the following description concerning electronic circuits, and then answer Subquestions 1 and 2.

Flip-flop circuits are used for designing computer hardware such as CPU, memory, counter etc. J-K flip-flop is one of the most commonly used such type of flip-flop circuits.
Figure 1 shows the block diagram of J-K flip-flop, and Table 1 shows the characteristic table of J-K flip-flop.

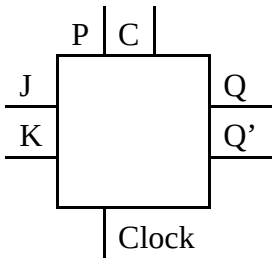



**Figure 1. Block diagram of J-K flip-flop**

**Table 1. Characteristic table of J-K flip-flop**

| Inputs | | | | | Output |
|---|---|---|---|---|---|
| P (Preset) | C (Clear) | J | K | Clock | Q |
| 1 | 1 | 0 | 0 | - | No change |
| 1 | 1 | 0 | 1 | - | Reset |
| 1 | 1 | 1 | 0 | - | Set |
| 1 | 1 | 1 | 1 | - | Toggle |
| 1 | 0 | any | any | - | 0 |
| 0 | 1 | any | any | - | 1 |

The outputs Q and Q**'** varies according to the values of J and K. Q and Q**'** are always inverse of each other. Toggle means the value is inversed, that is, if the value of Q is 0 then after toggle it becomes 1, and vice versa. Reset and Set means the value becomes 0 and 1 respectively.
Clock supplies the values 0 and 1 alternately and cyclically. When Clock rises (goes 0 to 1), the J-K flip-flop obtains the values of J and K. When Clock falls (goes 1 to 0), the J-K flip-flop outputs Q and Q' according to the last-obtained values of J and K.
If P=1 and C=0, or if P=0 and C=1, the value of Q is set as shown in Table 1, regardless of the values of J and K.

## Subquestion 1

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

(1) Assuming that, at the last clock rise, the input values were P=1, C=1, J=0 and K=1. Then, at the subsequent clock fall, the output values will be [ A ].

(2) Figure 2 shows an example of the input and output wave forms of the J-K flip-flop. In Figure 2, the wave form in shaded part (1) is [ B ], and the wave form in shaded part (2) can be either [ C ] or [ D ].
Here, P and C will have the values P=1 and C=1 throughout the given clock cycles.

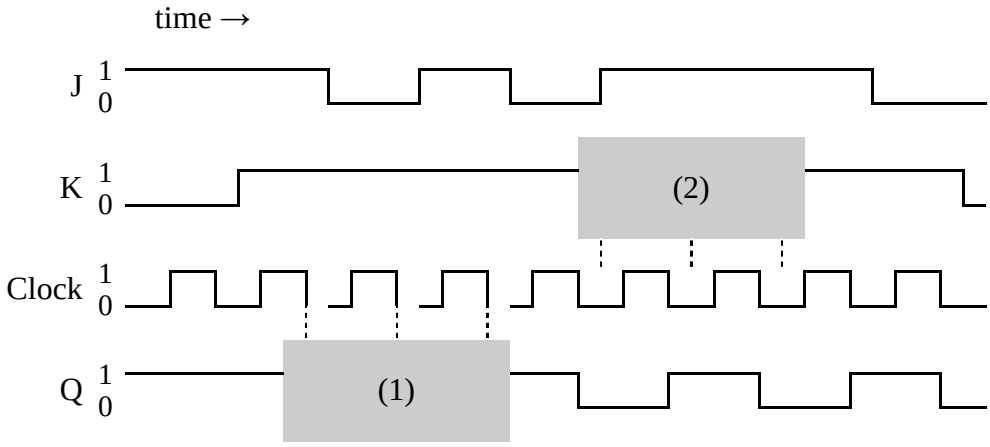
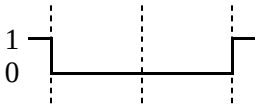


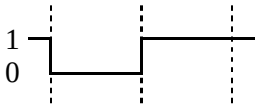

**Figure 2. Example of input and output wave forms of J-K flip-flop**

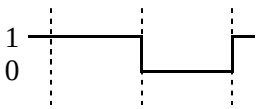
Answer group for A

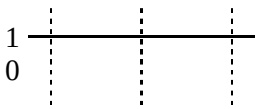
a) Q = 0 and Q' = 0
b) Q = 0 and Q' = 1
c) Q = 1 and Q' = 0
d) Q = 1 and Q' = 1

Answer group for B through D

a) 


b) 


c) 


d) 

## Subquestion 2

From the answer groups below, select the correct answer to be inserted into the blank [ ] in Table 2.

In designing electronic devices, J-K flip-flops can be used to realize a counter.
Figure 3 shows a 4-bit counter that uses 4 J-K flip-flops. All the inputs of P, C, J and K are 1s. On Clock input line on FF1, the cyclic clock pulse is supplied. However, on each Clock input line on FF2, FF3 and FF4, instead of the cyclic clock pulse, the output from the previous J-K flip-flop is supplied.

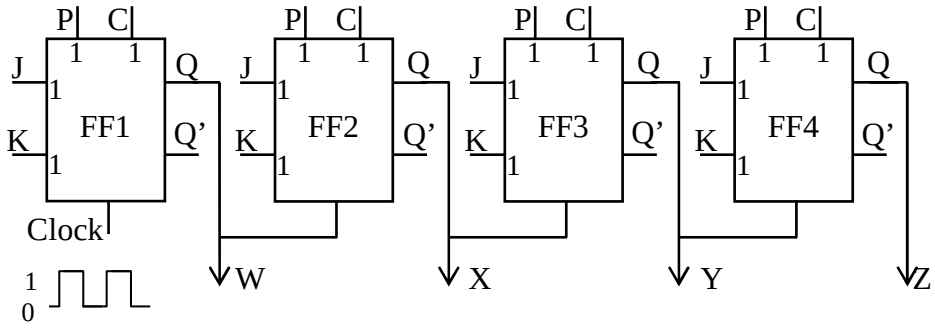



**Figure 3. 4-bit counter that uses 4 J-K flip-flops**

Assuming that, at the clock cycle *c*, the initial values of W, X, Y and Z are 0, 1, 0 and 0 respectively. Table 2 shows how the output values change on and after the clock cycle *c*.

**Table 2. Output values on and after the clock cycle *c***

| Clock Cycle | W | X | Y | Z |
|---|---|---|---|---|
| *c* | 1 | 0 | 0 | 0 |
| *c* + 1 | 0 | 1 | 0 | 0 |
| *c* + 2 | | | | |
| *c* + 3 | E | | | |

(Note) Shaded part is not shown

Answer group

a) 0 0 0 1

b) 0 0 1 0

c) 0 1 0 1

d) 1 0 1 0

**Q2.** Read the following description concerning a virtual memory system, and then answer Subquestion. In this question, "…h" denotes a hexadecimal value. For example, 100h denotes a hexadecimal value 100 (decimal value 256).

The basic idea behind virtual memory is to run a program whose size may exceed the physical memory size available for it. The operating system keeps those parts of the program currently in use in the physical memory, and the rest on the disk, by swapping pieces of the program between the physical memory and the disk as needed.

In a typical virtual memory system, a virtual address is split into a virtual page number (high-order bits) and a byte offset (low-order bits). Figure 1 shows an example of a simple virtual memory system. In this example, a 16-bit virtual address and a 4k-byte page size is used. Therefore, the virtual addresses range from 0 to [ A ], and the high-order 4 bits could specify one of the [ B ] virtual pages, and the low-order 12 bits would then specify the byte offset (0 to FFFh) within the virtual page.

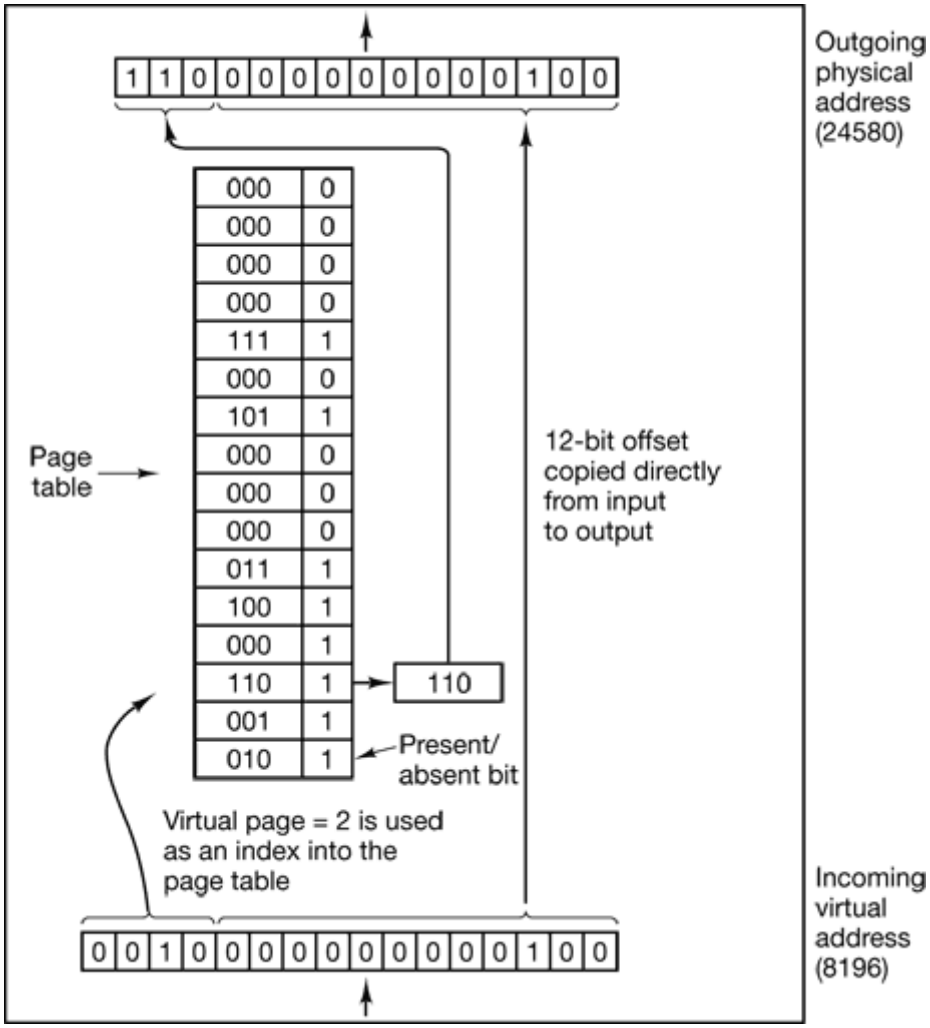



**Figure 1. Example of a simple virtual memory system**

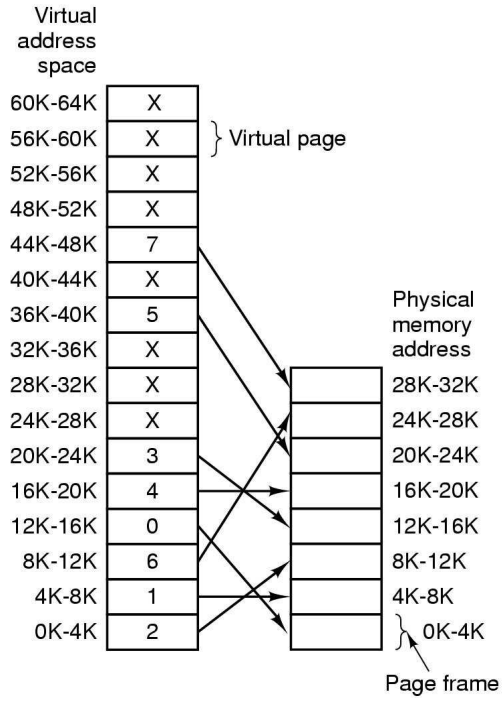



**Figure 2. Snapshot of the page table**

The relation between virtual addresses and physical addresses is given by the page table. Assuming a case in which program *A* is running on the virtual memory system shown in Figure 1. The size of program *A* is C000h bytes. A snapshot of the page table during the program execution is shown in Figure 2. At this point, 8 virtual pages are loaded into the physical memory, and the virtual address B300h corresponds to the physical address [ C ].

There are two major page replacement algorithms used for paging:

(1) FIFO (First In First Out)

When a page fault occurs, page-out the page whose page-in time was the oldest, and page-in the required page.

(2) LRU (Least Recently Used)

When a page fault occurs, page-out the page that has been unused for the longest time, and page-in the required page.

Consider the case of program *A* again.

(1) Before the program starts, no virtual pages are loaded into the physical memory.

(2) After the program starts, the virtual pages are accessed in the following sequence:

$$3 \rightarrow 1 \rightarrow 0 \rightarrow 5 \rightarrow 4 \rightarrow 9 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 11 \rightarrow 6$$
$$(\blacktriangle)$$

Figure 2 shows the snapshot of the page table at the point (▲) in the above sequence.

(3) After the point (▲), when the virtual page 6 is accessed, a page fault occurs. If the virtual memory system uses FIFO algorithm, the page [ D ] is paged-out, and the virtual page 6 is paged-in. If the virtual memory system uses LRU algorithm, the page [ E ] is paged-out, and the virtual page 6 is paged-in.

## Subquestion

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

Answer group for A

a) FFFh b) 1FFFh c) 7FFFh d) FFFFh

Answer group for B

a) 8 b) 16 c) 128 d) 256

Answer group for C

a) 0300h b) 1300h c) 2300h d) 3300h
e) 4300h f) 5300h g) 6300h h) 7300h

Answer group for D and E

a) 0 b) 1 c) 2 d) 3
e) 4 f) 5 g) 9 h) 11

**Q3.** Read the following description concerning a relational database, and then answer Subquestions 1 and 2.

An IT faculty of a university is going to develop the system to manage all materials of different subjects. Ms. A is in charge of getting requirements for this system. She identifies some main requirements:

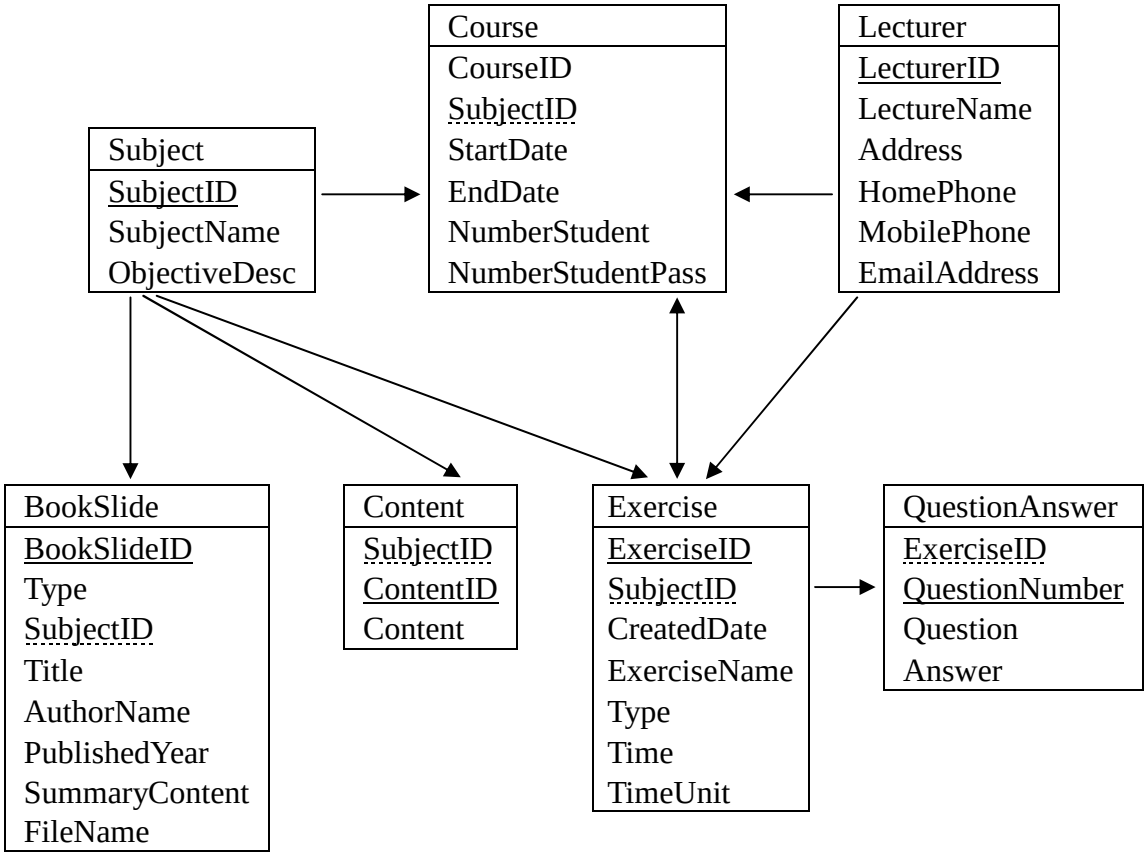
(1) The system records each subject with the subject name, text description of objectives, list of text books and lecture slides, and the list of contents. Each subject has many exercises to evaluate the students' competence.
(2) The system records information about each lecturer, such as the name, home address, home phone number, mobile phone number, e-mail address.
(3) Each text book and lecture slide has the title, author name, published year, the summary of content, and file name and its path where the content is stored.
(4) The system records information about each exercise, such as the lecturer who created it, created date, time for doing (in hours, days, or weeks), type of exercise (individual homework, individual test, team assignment, team project), list of questions and answers. Each exercise corresponds to one subject.
(5) Each subject has different courses. The system records information about each course, such as the start date, end date, the number of students who attended the course, and the number of passed students.
(6) During the course, the main lecturer can make the new exercises or use the old ones. The history of exercises used for each course is recorded. In order to evaluate the difficulty level of each exercise, the average grade of students of each course are also recorded.

### Subquestion 1

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

Based on the above requirements, Ms. A is planning to create a database for the system. She first created the E-R Diagram of the database (incomplete) as shown in Figure 1.

To complete the E-R Diagram, the attribute LecturerID should be added to the entity [ A ] and entity [ B ], as a foreign key.




(Note) A solid underline _____ indicates an attribute of a primary key.
A dashed underline ..... indicates an attribute of a foreign key.

**Figure 1. The E-R Diagram of the database (incomplete)**

Answer group

a) BookSlide
b) Content
c) Course
d) Exercise
e) QuestionAnswer
f) Subject

### Subquestion 2

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

Based on the E-R Diagram completed in Subquestion 1, a database is created for the material management system.

Recently, the lecturers requested of Ms. A to provide an SQL statement that lists all exercises used for a specified subject, as shown below. The list includes the exercise name, average grades of students, and use count (the number of courses which use that exercise).

| Exercise Name | Average Grades | Use Count |
|---|---|---|
| Class Diagram | 65 | 3 |
| E-R Diagram | 79 | 2 |
| SQL coding | 84 | 5 |

Ms. A first creates a new table CourseExercise that contains the average grades of students by course and exercise, as shown below.

| CourseExercise |
|---|
| CourseID<br>ExerciseID<br>AverageGrade |

Ms. A then creates the following SQL statement. Here, SubjectID of the specified subject is given by the host variable `:SubjectID`.

```
SELECT Exercise.ExerciseName,
       AVG(      C      ) AS AverageGrades,
       COUNT(*) AS UseCount
FROM Exercise, CourseExercise
     WHERE   D
     AND     E
GROUP BY Exercise.ExerciseID, ExerciseName
```

Answer group for C

a) `Course.CourseID`
b) `Course.SubjectID`
c) `CourseExercise.AverageGrade`
d) `CourseExercise.CourseID`
e) `Exercise.ExerciseID`
f) `Exercise.SubjectID`

Answer group for D and E

a) `Course.SubjectID = Exercise.SubjectID`
b) `CourseExercise.ExerciseID = Exercise.ExerciseID`
c) `CourseExercise.CourseID = Course.CourseID`
d) `Exercise.SubjectID = :SubjectID`
e) `Subject.SubjectID = :SubjectID`

**Q4.** Read the following description concerning a network for a small office, and then answer Subquestion.

A simple IP network using private IP addresses (192.168.1.0/24) is setup for a small office as shown in Figure 1. Some computers in the office IP network access the Internet through an ADSL router with Network Address Translation (NAT) function and Packet Filtering Firewall enabled. The IP Routing table, NAT table, and Packet Filtering Firewall table of ADSL router are correctly configured according to the setup guideline provided by the Internet Service Provider (ISP).

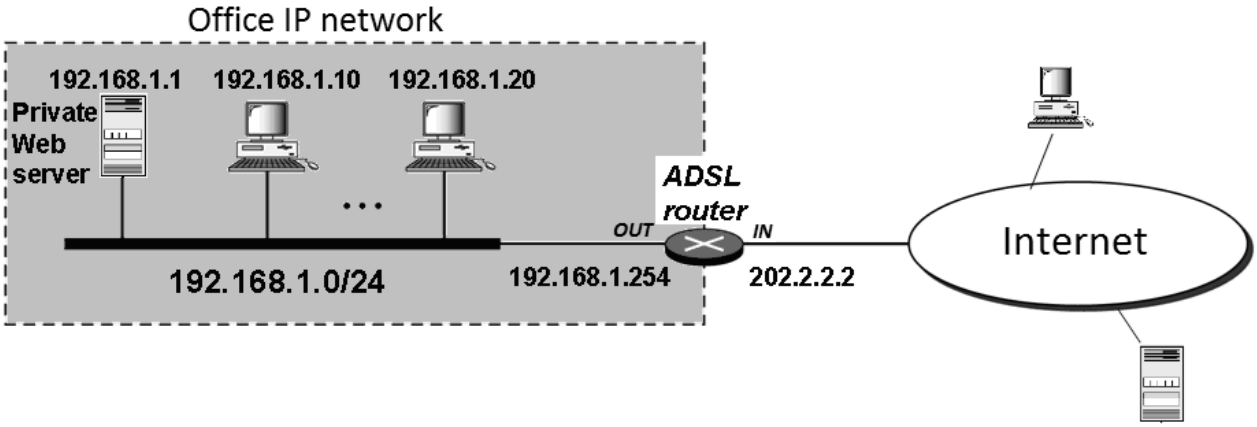



**Figure 1. Configuration of office IP network**

The network settings are checked according to the following three steps.

The first step is to verify the routing table entry of each computer in the office IP network. Table 1 shows a part of the routing table entries. In Table 1, the entry “Local access” is for direct communication within the office IP network, and the entry “Internet access” is for communication with computers on the Internet.

**Table 1. Routing table entries of each computer**

| | Destination Address | Netmask | Gateway | … |
|---|---|---|---|---|
| Local access | 192.168.1.0 | A | | … |
| Internet access | 0.0.0.0 | 0.0.0.0 | B | … |

(Note) Shaded part is not shown

The second step is to monitor the Internet usage whether it complies with the office Internet usage policy or not. The information in the NAT table of ADSL router can be used to simply track current Internet usage. A snapshot of the NAT table of ADSL router is shown in Table 2.

**Table 2. Information in the NAT table of ADSL router**

| Private address | Private port | External address | External port | NAT port | Protocol |
|---|---|---|---|---|---|
| 192.168.1.12 | 12345 | 205.5.5.1 | 53 | 1003 | UDP |
| 192.168.1.12 | 62345 | 209.85.231.104 | 80 | 1005 | TCP |
| 192.168.1.1 | 80 | 60.1.1.13 | 31245 | 1006 | TCP |
| 192.168.1.10 | 21392 | 205.5.5.1 | 53 | 1004 | UDP |
| 192.168.1.10 | 20001 | 203.151.20.61 | 80 | 1005 | UDP |
| 192.168.1.20 | 12345 | 210.12.1.1 | 25 | 1007 | TCP |

From Table 2, it is found that the number of computers that is accessing external HTTP Web servers is [ C ], because the default protocol and port number of HTTP Web server are TCP and port 80 respectively.
In case of the access to the external server 203.151.20.61 from the computer 192.168.1.10, the source port, destination port, source IP, and destination IP seen by the external server are 1005, 80, [ D ], and [ E ] respectively.

The last step is to set firewall rules in ADSL router to block or allow Internet usage traffic according to the office Internet usage policy. Table 3 shows the structure of firewall table of ADSL router.

**Table 3. Structure of firewall table of ADSL router**

| Arrive on Interface | Source | | Destination | | Protocol | Action |
|---|---|---|---|---|---|---|
| | IP Address | Port | IP Address | Port | | |

- "Arrive on Interface" field should specify either "IN" or "OUT". "IN" is the interface connecting to the Internet, and "OUT" is the interface connecting to the office IP network.
- "Protocol" field should specify the transport (or other) protocol the rule is applied to (if relevant).
- "Action" field should specify either "Drop" or "Accept".
- For any field in the table, "*" can be used to indicate "any value". For example, if the rule is independent of the protocol, specify "*" in Protocol field.

The last record in Table 2 reveals that the direct access to external email server running at port 25 is currently opened. Thus any virus-infected program within the office IP network can easily send spam mail through this security hole. Since all email accesses in the office are done via Web-based email services using HTTP port 80, it is not necessary to open the direct email server access at port 25. To block such outbound traffic from any computer within the office IP network, a packet filtering rule should be added to the firewall table. The rule should contain the field values [ F ] (each value corresponds to each field in Table 3, from left to right).

## Subquestion

From the answer groups below, select the correct answer to be inserted into each blank [ ] in the above description and Table 1.

Answer group for A, B, D and E

a) 0.0.0.0 b) 192.168.1.0 c) 192.168.1.254 d) 202.2.2.2
e) 203.151.20.61 f) 255.255.255.0 g) 255.255.255.192

Answer group for C

a) 0 b) 1 c) 2 d) 3

Answer group for F

a) IN, *, *, *, 25, TCP, Accept
b) IN, *, *, *, 25, TCP, Drop
c) IN, *, 25, *, *, TCP, Accept
d) IN, *, 25, *, *, TCP, Drop
e) OUT, *, *, *, 25, TCP, Accept
f) OUT, *, *, *, 25, TCP, Drop
g) OUT, *, 25, *, *, TCP, Accept
h) OUT, *, 25, *, *, TCP, Drop

**Q5.** Read the following description concerning wireless network security, and then answer Subquestion.

Company *X* is an e-commerce company selling various products online. With only 10 employees, the company is relatively small. Its e-commerce system consists of a Web server and a mail server. Recently, the company was renovated and its servers were outsourced to an external entity to reduce the hardware and infrastructure cost, leaving only client computers in the network.

After the renovation, the employees requested that they want to use their own portable computers and smart devices at work. Mr. *A*, the owner of Company *X*, agreed with the employees and installed a wireless access point (hereinafter, AP) in the company instead of upgrading the client computers. The current network configuration of Company *X* is shown in Figure 1.

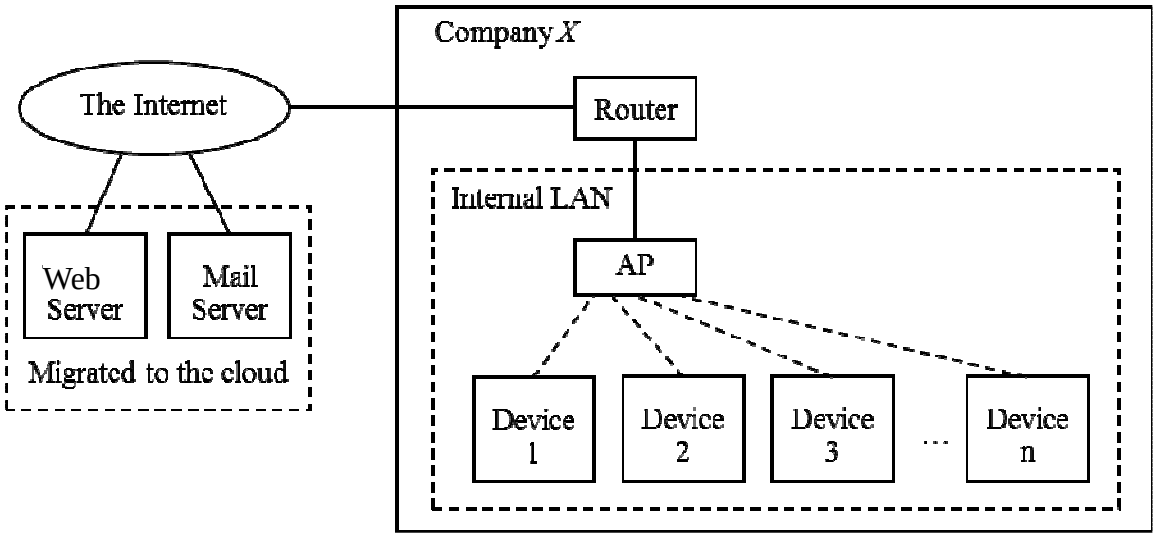



**Figure 1. The current network configuration of Company *X***

Initially, [ A ] filtering was deployed as the primary security measure. It is an access control mechanism available to allow only pre-registered devices to connect to the network. Unlike an IP address where the user can manually change easily, [ A ] is embedded in the network interface card makes it more difficult to change without proper utilities.

Furthermore, Mr. *A* decided to hide the name of the network from the public. Normally, an AP sends beacon frames to provide information for other devices to join the network. This process is called beaconing.

An important information necessary to initiate the connection is [ B ], which represents the network name. If the [ B ] beaconing is turned off, the users are required to enter the network name manually and will be able to connect to the network only if it is matched with the one previously entered on the AP.

Since these mechanisms do not provide encryption, it is possible that the communication may be intercepted and interpreted easily by attackers. To maintain confidentiality, Mr. *A* decided to enable [ C ] on the Web server to secure the HTTP connections between the Web server and the browsers. When used in conjunction with plain HTTP, the secure version of the protocol called HTTPS, which uses port 443 instead of port 80, is available.

Despite the implemented security measures, it turned out that unauthorized users have gained access to the network and use the network as a free Wi-Fi. Although the network name is not broadcasted by the AP and is invisible to the public, it is revealed whenever a legitimate user trying to associate with the AP. Figure 2 shows parts of the information captured from the wireless networks near Company *X*.

| AP | Encryption | Network Name |
|---|---|---|
| xx:xx:xx:C3:5F:AC | WPA | |
| xx:xx:xx:2C:46:80 | Open | |
| xx:xx:xx:1B:83:60 | Open | Public |
| xx:xx:xx:8E:1D:50 | WEP | Warehouse |

| AP | Client Station | Probe |
|---|---|---|
| (not associated) | xx:xx:xx:B2:62:3C | CAFE |
| (not associated) | xx:xx:xx:1C:BB:79 | WLAN |
| xx:xx:xx:C3:5F:AC | xx:xx:xx:E1:19:A1 | Office |
| xx:xx:xx:2C:46:80 | xx:xx:xx:F0:37:DB | |
| xx:xx:xx:2C:46:80 | xx:xx:xx:A4:26:F0 | WLAN |
| xx:xx:xx:8E:1D:50 | xx:xx:xx:DD:8F:44 | |

**Figure 2. Parts of the information captured near Company *X***

As shown in Figure 2, there are 4 APs available near Company *X*. Two of them have hidden network names, and those names are revealed as “Office” and “WLAN” with the [ A ] of xx:xx:xx:C3:5F:AC and xx:xx:xx:2C:46:80 respectively.
Since the wireless network of Company *X* has no encryption enabled, it is obvious that the AP with the network name [ D ] belongs to Company *X*.

Furthermore, from Figure 2, it is most likely that an unauthorized user will be able to connect to Company *X*'s network later on, by inappropriately changing his/her computer's [ A ] to [ E ], which is actually allowed to connect to the AP.

Mr. *A* realized that encryption should be enabled in order to prevent unauthorized access. Among the APs in Figure 2, the network name of the most secure AP is [ F ]. Mr. *A* is now studying the encryption method WPA2, that is better than the ones used in Figure 2, and he may choose it at a later time.

## Subquestion

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

Answer group for A through C

a) DHCP b) MAC address c) Passphrase d) SNMP
e) SSID f) SSH g) SSL

Answer group for D and F

a) CAFE b) Office c) Public
d) Warehouse e) WLAN

Answer group for E

a) xx:xx:xx:1B:83:60 or xx:xx:xx:2C:46:80
b) xx:xx:xx:1C:BB:79
c) xx:xx:xx:2C:46:80
d) xx:xx:xx:A4:26:F0 or xx:xx:xx:F0:37:DB
e) xx:xx:xx:C3:5F:AC
f) xx:xx:xx:E1:19:A1

**Q6.** Read the following description of a merge sort algorithm and the flowcharts themselves, and then answer Subquestion.

[Program Description]

Conceptually, a merge sort works as follows:

(1) If the list is of length 0 or 1, then it is already sorted.
(2) Otherwise, divide the unsorted list into two sub-arrays of about half the size.
(3) Sort each sub-array recursively by re-applying merge sort.
(4) Merge the two sub-arrays back into one sorted list.

Merge sort incorporates two main ideas to improve its runtime:

(1) A small list will take fewer steps to sort than a large list.
(2) Fewer steps are required to construct a sorted list from two sorted lists than two unsorted lists. For example, it is enough to traverse each list only once if they're already sorted (see the merge function below for an example implementation).

Figure 1 shows an example of merge sort to sort a list of integers contained in an array. It shows how the elements [38, 27, 43, 3, 9, 82, 10] are sorted in ascending order to get the result [3, 9, 10, 27, 38, 43, 82] using merge sort algorithm.

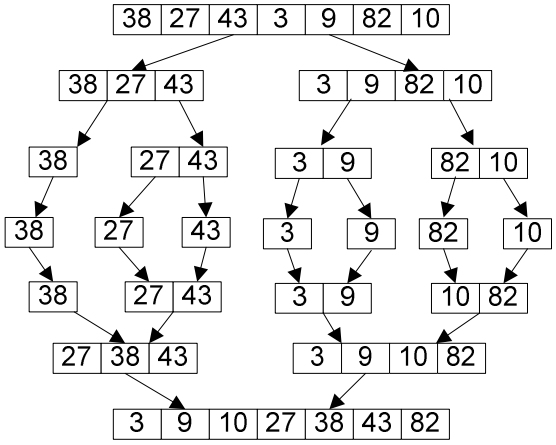



**Figure 1. Merge sort example**

Generally, when an array *A* with *n* elements (ranging from *A*[1] to *A*[*n*]) is to be sorted, apply merge sort to 2 sub-arrays (A[*l*]: *l* = 1, 2, ..., *c*) and (A[*r*]: *r* = *c*+1, *c*+2, ..., *n*), where *c* is the integer part of *n* ÷ 2. When the 2 sub-arrays are returned, they will have been sorted. They can now be merged together to form a sorted array.

Figure 2 shows the flowchart of the function Merge_sort(m). The function sorts the data in array m in ascending order.

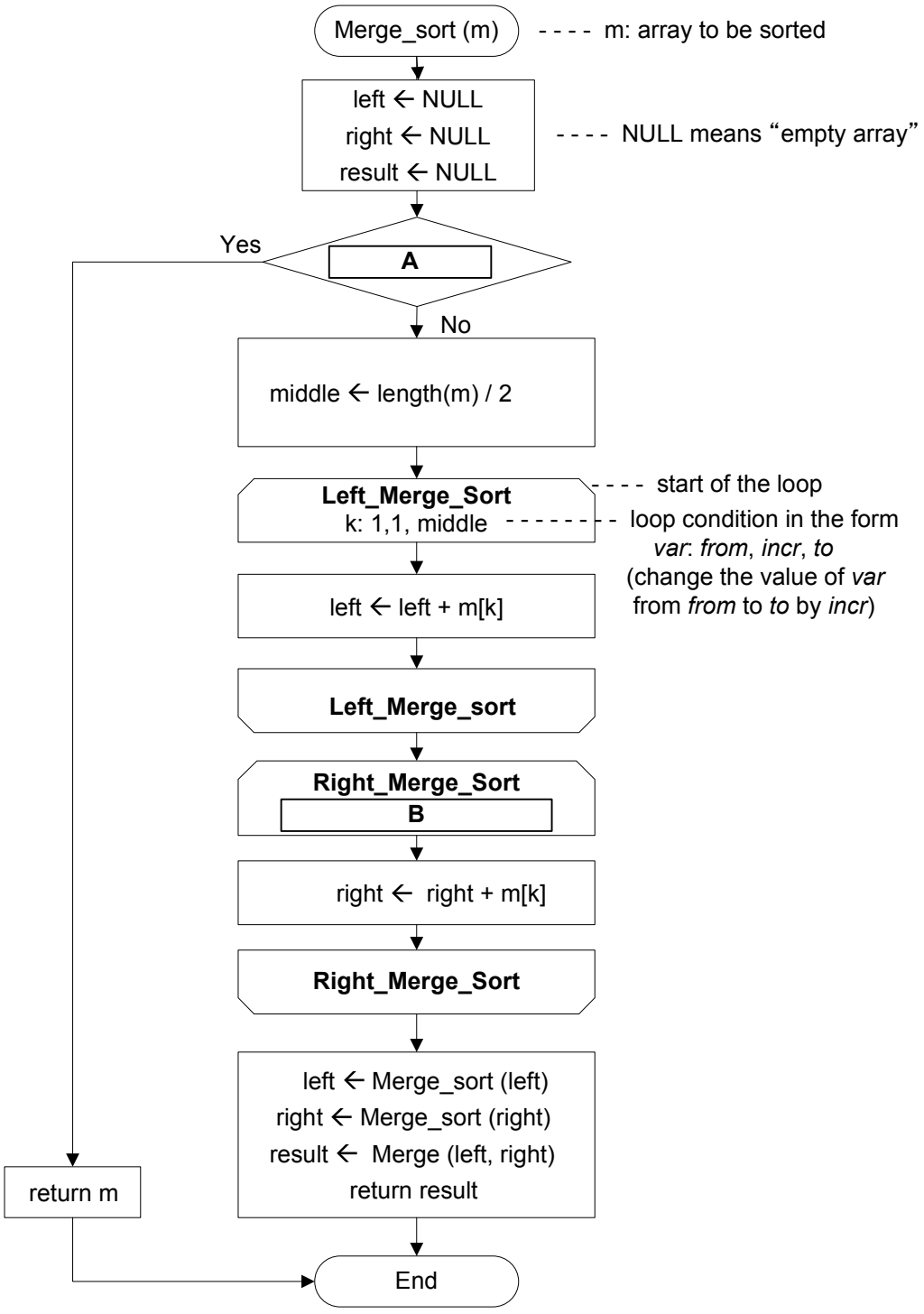



**Figure 2. Function Merge_sort(m)**

Explanation:

- All variables in Figure 2 are integer type.
- k is a temporary variable.
- left, right and result are temporary arrays to keep the elements during sorting process. They are initialized as empty arrays at the beginning.
- left ← left + m[k] denotes that array m[k] is appended to left array.
- right ← right + m[k] denotes that array m[k] is appended to right array.
- Function length(m) returns the number of elements of array m (returns 0 if m is NULL).

Figure 3 shows the flowchart of the function Merge(left, right) called from the function Merge_sort(m). This function merges 2 sub-arrays left and right.

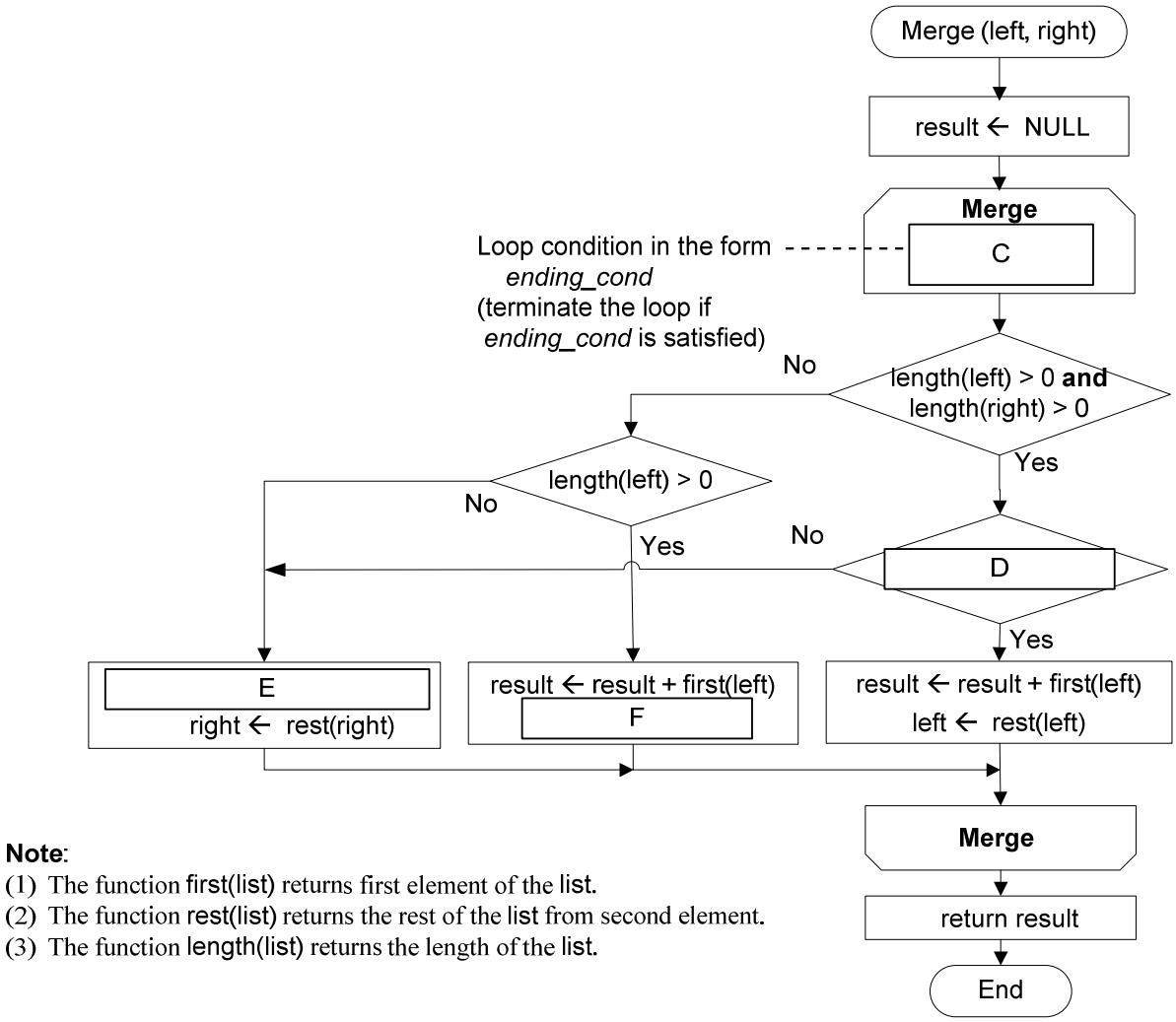



**Figure 3. Function Merge(left, right)**

## Subquestion

From the answer groups below, select the correct answer to be inserted into each blank [ ] in Figure 2 and Figure 3.

Answer group for A

a) length(m) < 1
b) length(m) ≤ 1
c) length(m) = 1
d) length(m) > 1
e) length(m) ≥1

Answer group for B

a) k: middle+1, length(m), 1
b) k: middle+1, length(m), -1
c) k: middle, length(m), 1
d) k: middle, length(m), -1

Answer group for C

a) length(left) ≠ 0 and length(right) ≠ 0
b) length(left) = 0 and length(right) = 0
c) length(left) = 0 or length(right) = 0
d) length(left) > 0 and length(right) > 0
e) length(left) ≥ 0 and length(right) ≥ 0

Answer group for D

a) first(left) ≤ first(right)
b) first(left) = first(right)
c) first(left) = first(right) + 1
d) first(left) = first(right) - 1
e) first(left) > first(right)

Answer group for E and F

a) left ← rest (left)
b) left ← rest (right)
c) result ← result + first (left)
d) result ← result + first (right)
e) right ← rest (left)
f) right ← rest (right)

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

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

[Program Description]

A word or a character sequence is called a “palindrome”, if it remains the same when written backwards (right to left). For example, the word “MADAM” is a palindrome, because it is the same when it is written backwards.

A word is called a “mirrored word”, if each character of the word has its mirror character and the word written backwards gives the original word. For example, the word “2TOTS” is a mirrored word, because “T” and “O” are mirror character of their own, and “2” and “S” are the mirror character of each other.

Some words are palindrome as well as mirrored. Such kind of words are called “mirrored palindrome”. The word “XOYOX” is a mirrored palindrome, because it is a palindrome as well as a mirrored word. Here, “X”, “O” and “Y” are all their own mirror character.

Table 1 shows the acceptable characters and their mirror characters. Acceptable characters consist of upper case alphabets, digits and space. ‘\0’ (null) represents there is no mirror character for that character. Here, “0” (zero) and “O” (oh) will be considered as the same character, and hence only the character “O” is given in Table 1.

**Table 1. Acceptable characters and their mirror characters**

| Character | Mirror Character |
|---|---|
| A | A |
| B | \0 |
| C | \0 |
| D | \0 |
| E | 3 |
| F | \0 |
| G | \0 |
| H | H |
| I | I |
| J | L |
| K | \0 |
| L | J |

| Character | Mirror Character |
|---|---|
| M | M |
| N | \0 |
| O | O |
| P | \0 |
| Q | \0 |
| R | \0 |
| S | 2 |
| T | T |
| U | U |
| V | V |
| W | W |
| X | X |

| Character | Mirror Character |
|---|---|
| Y | Y |
| Z | 5 |
| 1 | 1 |
| 2 | S |
| 3 | E |
| 4 | \0 |
| 5 | Z |
| 6 | \0 |
| 7 | \0 |
| 8 | 8 |
| 9 | \0 |

Input character string contains words to check and each separated by exactly one space. The length of input character string should not exceed 100. The last character in input character string must not be a space.

The function `gets()` reads input character string, and adds '\0' after the last character.

For each word in the input character string, the program prints one of the following message according to the conditions.

| Message | Condition |
|---|---|
| **XXX** is an ordinary word. | If **XXX** is neither a palindrome nor a mirrored word |
| **XXX** is an ordinary palindrome. | If **XXX** is a palindrome and is not a mirrored word |
| **XXX** is a mirrored word. | If **XXX** is not a palindrome and is a mirrored word |
| **XXX** is a mirrored palindrome. | If **XXX** is a palindrome as well as a mirrored word |

Example of an input character string and its output messages are as follows:

(Input character string)

```
ITPEC MADAM 2TOTS XOYOX
```

(Output messages)

```
ITPEC is an ordinary word.
MADAM is an ordinary palindrome.
2TOTS is a mirrored word.
XOYOX is a mirrored palindrome.
```

The program has 2 user defined functions:

(1) `int check(char str[], int i, int j)`

This function checks the word, and returns 3 for mirrored palindrome, 2 for ordinary palindrome, 1 for mirrored word, and 0 for ordinary word. Here, `str[]` is the input character string, `i` is the start index, and `j` is the end index of the word.

(2) `int getEndIndex(char str[], int i)`

This function returns the end index of the word in the input character string `str[]` whose start index is `i`.

[Program]

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

char reverse[128];

int check(char str[], int i, int j) {
   int val = 3;

   for(;  A  ; i++, j--) {
      if (str[i] != str[j]) {
         B ;
      }
      if (str[i] != reverse[str[j]]) {
         C ;
      }
   }
   return val;
}

int getEndIndex(char str[], int i) {
   while (  D  ) {
      i++;
   }
   return i - 1;
}
```

```
void main() {
   int i, j, sIndex = 0, eIndex;
   char str[101], s[101];

   for (i = 0; i < 128; i++) {
      reverse[i] = '\0';
   }

   reverse['A'] = 'A'; reverse['E'] = '3'; reverse['H'] = 'H';
   reverse['I'] = 'I'; reverse['J'] = 'L'; reverse['L'] = 'J';
   reverse['M'] = 'M'; reverse['O'] = 'O'; reverse['S'] = '2';
   reverse['T'] = 'T'; reverse['U'] = 'U'; reverse['V'] = 'V';
   reverse['W'] = 'W'; reverse['X'] = 'X'; reverse['Y'] = 'Y';
   reverse['Z'] = '5'; reverse['1'] = '1'; reverse['2'] = 'S';
   reverse['3'] = 'E'; reverse['5'] = 'Z'; reverse['8'] = '8';

   gets(str);

   while (str[sIndex] != '\0') {
      while (str[sIndex] == ' ') {
         sIndex++;
      }
      eIndex = getEndIndex(str, sIndex);
      for (i = 0, j = sIndex; i <= eIndex - sIndex; i++, j++) {
         s[i] = str[j];
      }
      s[i] = '\0';

      switch (check(str, sIndex, eIndex)) {
         case 3:
            printf("%s is a mirrored palindrome.\n", s);
            break;
         case 2:
            printf("%s is an ordinary palindrome.\n", s);
            break;
         case 1:
            printf("%s is a mirrored word.\n", s);
            break;
         default:
            printf("%s is an ordinary word.\n", s);
      }
      [      E      ];
   }
}
```

### Subquestion 1

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

Answer group for A

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

Answer group for B and C

a) `val &= 1`  b) `val &= 2`  c) `val += 1`
d) `val += 2`  e) `val |= 1`  f) `val |= 2`

Answer group for D

a) `str[i] != ' ' && str[i] != '\0'`
b) `str[i] != ' ' || str[i] != '\0'`
c) `str[i] == ' ' && str[i] == '\0'`
d) `str[i] == ' ' || str[i] == '\0'`

Answer group for E

a) `eIndex = sIndex + 1`  b) `eIndex = sIndex - 1`
c) `sIndex = eIndex + 1`  d) `sIndex = eIndex - 1`

### Subquestion 2

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

The function `getEndIndex(char str[], int i)` will return the value [ ], if the value of `str[]` is "`MADAM ADAM PALINDROME`", and the value of `i` is 6.

Answer group

a) 6  b) 7  c) 8
d) 9  e) 10

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

[Program Description]

The programs implement the operation of electronic device for measuring blood pressure.

(1) The device automatically measures systolic and diastolic blood pressure reading in mmHg, and displays the measurement results. Table 1 is used to determine the measurement results.

**Table 1. Interpretation of blood pressure measurement results**

| Category of Blood Pressure | Systolic BP (mmHg) | Diastolic BP (mmHg) | Advice on Results | Color Indicator |
|---|---|---|---|---|
| Hypotension | < 100 | < 60 | Practice a healthy lifestyle. Consult your doctor only if suffering symptoms of low BP (e.g. fainting). | WHITE |
| Normal | < 140 | < 90 | Practice a healthy lifestyle. | GREEN |
| PreHypertension | 140–159 | 90–99 | Practice a healthy lifestyle. Re-measure BP monthly over next 3 months. If high levels (≥ 140/90) persist (e.g. 2 high readings on 2 separate occasions), consult doctor. | YELLOW |
| Moderate Hypertension | 160–179 | 100–109 | Practice a healthy lifestyle. Re-measure BP monthly over next 4 months. If high levels (≥ 160/100) persist (e.g. 2 high readings on 2 separate occasions), consult doctor. | ORANGE |
| Severe Hypertension | ≥ 180 | ≥ 110 | Re-measure BP in a few days. If BP ≥ 180/110, consult doctor. | RED |

(2) If systolic BP and diastolic BP fall into different categories, the higher value should be taken for classification. For example, 160/92 should be classified as “Moderate Hypertension” and 180/92 should be classified as “Severe Hypertension”.

(3) The programs are composed of 5 classes; `BloodPressureTest`, `Tension`, `Pressure`, `Device`, and `InvalidPressuresException`.

[Program 1]

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class BloodPressureTest {
   static Device device;

   public static void main(String[] args) {
      device = new Device();    // Create a new device.
      Scanner input = new Scanner(System.in);
      System.out.print("Enter the High and Low blood pressures: ");
      try {
         int h = input.nextInt();
         int l = input.nextInt();
         if (l >= h) {
            throw new InvalidPressuresException
               ("The Low value must be less than the High value");
         } else {
            device.turnOn(l, h);    // a user launches the device
            System.out.println(device);
         }
      } catch (InvalidPressuresException e) {
         System.out.println(e.getMessage());
      } catch (InputMismatchException e) {
         System.out.println("You must input only integers");
      }
   }
}
```

[Program 2]

```
import java.util.EnumSet;
import java.util.HashMap;
import java.util.Map;

public enum Tension {
   HYPOTENSION, NORMAL, PREHYPERTENSION,
   MODERATE_HYPERTENSION, SEVERE_HYPERTENSION;
   protected static final Map<      A      >
      lookup = new HashMap<      A      >();
   static {
      int ordinal = 0;
      for (Tension suit : EnumSet.allOf(Tension.class)) {
         lookup.put(ordinal, suit);
         ordinal += 1;
      }
   }
```

```
    public static Tension fromOrdinal(int ordinal) {
       return lookup.get(ordinal);
    }
  }
```

[Program 3]

```
  public class Pressure {
    private static String newline = "\n";
    static String advices[] = {
       "Practice a healthy lifestyle." + newline
          + "Consult your doctor only" + newline
          + "if suffering symptoms of low BP (e.g. fainting).",

       "Practice a healthy lifestyle.",

       "Practice a healthy lifestyle." + newline
          + "Re-measure BP monthly over next 3 months." + newline
          + "If high levels (>=140/90) persist, consult doctor.",

       "Practice a healthy lifestyle." + newline
          + "Re-measure BP monthly over next 4 months." + newline
          + "If high levels (>=160/100) persist, consult doctor.",

       "Re-measure BP in a few days." + newline
          + "If BP>=180/110, consult doctor."
    };

    private [ B ] status;
    private int sbp;
    private int dbp;

    public Pressure() {
       this.sbp = 0;
       this.dbp = 0;
    }

    public Pressure(int low, int high) {
       this.sbp = high;
       this.dbp = low;
    }

    public void setStatus([ B ] state) {
       this.status = state;
    }

    public [ B ] getStatus() {
       return this.status;
    }
```

```
    public int getHValue() {
       return this.sbp;
    }

    public int getLValue() {
       return this.dbp;
    }

    public String toString(int i) {
       return advices[i];
    }
  }
```

[Program 4]

```
  public class Device {
    enum Indicators {
       WHITE,
       GREEN,
       YELLOW,
       ORANGE,
       RED
    };

    private static final int high[] = {100, 140, 160, 180};
    private static final int low[] = {60, 90, 100, 110};
    static String newline = "\n";

    Pressure pressures;
    Indicators indication;
    private int sbp;    //pressure's high value
    private int dbp;    //pressure's low value

    public void turnOn(int l, int h) {
       pressures = new Pressure(l, h);
       pressures.setStatus(diagnose());
    }

    private [ B ] diagnose() {
       Tension state = null;
       sbp = pressures.getHValue();
       dbp = pressures.getLValue();
       int i = 0;
       while((i < high.length) && ([ C ])){
          i++;
       }
       state = Tension.fromOrdinal(i);
       return state;
    }
```

```
    private Indicators getIndicator() {
       switch (pressures.getStatus().ordinal()) {
          case 0:
             indication = Indicators.WHITE;
             break;
          case 1:
             indication = Indicators.GREEN;
             break;
          case 2:
             indication = Indicators.YELLOW;
             break;
          case 3:
             indication = Indicators.ORANGE;
             break;
          case 4:
             indication = Indicators.RED;
             break;
       }
       return indication;
    }

    public String toString() {
       return "BLOOD PRESSURE MONITOR" + newline
          + "------------------------" + newline + "SBP: "
          + pressures.getHValue() + " | " + "DBP: "
          + pressures.getLValue() + newline + "Your blood pressure is "
          + pressures.getStatus() + newline
          + pressures.toString(pressures.getStatus().ordinal())
          + newline + "------------------------" + newline;
    }
 }
```

[Program 5]

```
 class InvalidPressuresException extends Exception {
    public InvalidPressuresException (String msg) {
       super(msg);
    }
 }
```

### Subquestion 1

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

Answer group for A

a) `Integer, Pressure`
b) `Integer, Tension`
c) `int, Pressure`
d) `int, Tension`

Answer group for B

a) `Pressure`
b) `Tension`
c) `Tension.HYPOTENSION`
d) `enum Tension`

Answer group for C

a) `sbp < high[i] || dbp < low[i]`
b) `sbp <= high[i] || dbp <= low[i]`
c) `sbp > high[i] || dbp > low[i]`
d) `sbp >= high[i] || dbp >= low[i]`

### Subquestion 2

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

In the program testing phase, various test cases are prepared and tested. Each test case has a pair of high (systolic) and low (diastolic) BP values.

Table 2 shows the test cases (1), (2) and (3), with their test data and output results.

**Table 2. Test cases and output results**

| Test case | Test data | | Output result |
|---|---|---|---|
| | High | Low | |
| Test case (1) | 88.0 | 138 | D |
| Test case (2) | 138 | 90 | E |
| Test case (3) | 160 | 98 | F |

Answer group (Note: shaded parts [redacted] are not shown.)

a)
```
BLOOD PRESSURE MONITOR
-----------------------
SBP: [redacted] | DBP: [redacted]
Your blood pressure is MODERATE_HYPERTENSION
Practice a healthy lifestyle.
Re-measure BP monthly over next 4 months.
If high levels (>=160/100) persist, consult doctor.
-----------------------
```

b)
```
BLOOD PRESSURE MONITOR
-----------------------
SBP: [redacted] | DBP: [redacted]
Your blood pressure is NORMAL
Practice a healthy lifestyle
-----------------------
```

c)
```
BLOOD PRESSURE MONITOR
-----------------------
SBP: [redacted] | DBP: [redacted]
Your blood pressure is PREHYPERTENSION
Practice a healthy lifestyle.
Re-measure BP monthly over next 3 months.
If high levels (>=140/90) persist, consult doctor.
-----------------------
```

d)
```
The Low value must be less than the High value
```

e)
```
You must input only integers
```