October 2012

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 process	Indicates a one-way selection process. If the conditional expression is `true`, then the process is executed.
	 conditional expression process 1 process 2	Indicates a two-way selection process. If the conditional expression is `true`, then the process 1 is executed. If it is `false`, then the process 2 is executed.
	 conditional expression process	Indicates a pre-test iteration process. While the conditional expression is `true`, the process is executed repeatedly.
	 process conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is `true`, the process is executed repeatedly.
	 variable: init, cond, incr process	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is `true`, the process is executed repeatedly. The increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true, false`

(continued on next page)

[Operators and their priorities]

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

Note: With division of integers, 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 concerning a logic circuit, and then answer Subquestions 1 and 2.

Figure 1 shows a logic circuit that is used as a control system for an electronic device.
The logic circuit has three inputs (X, Y and Z) and one output (F). As shown in Figure 1, the logic circuit has a feed back line, by which the last output F is fed back and is used as the current input to the NAND gate. Therefore, the current output F is determined by three current inputs X, Y and Z and the last output F. In this control system, among three inputs X, Y and Z, at most only one input can change its value on each clock cycle.

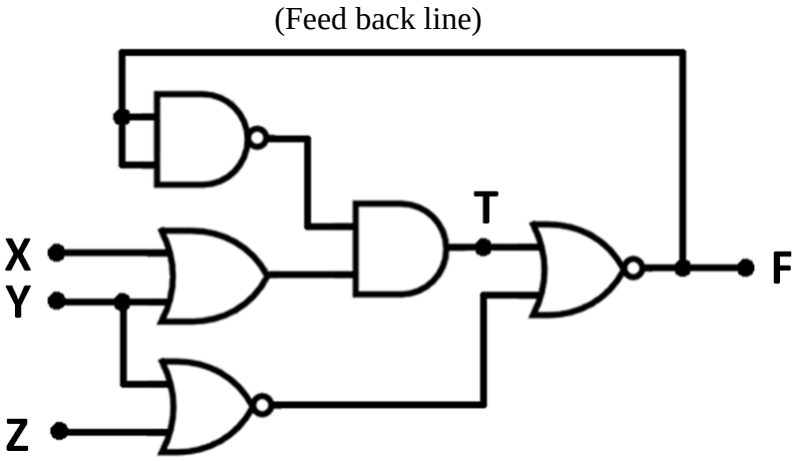


Figure 1 The logic circuit

Meanings of the logic gate symbols used in Figure 1 are as follows:

AND gate NAND gate OR gate NOR gate

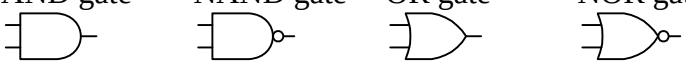


Figure 2 shows sample input/output status of the logic circuit. For example, the output F on the clock cycle *c* (value 0) is obtained from the inputs X, Y and Z on the clock cycle *c* (values 1, 0 and 1 respectively) and the output F at the clock cycle *c* - 1 (value 0).

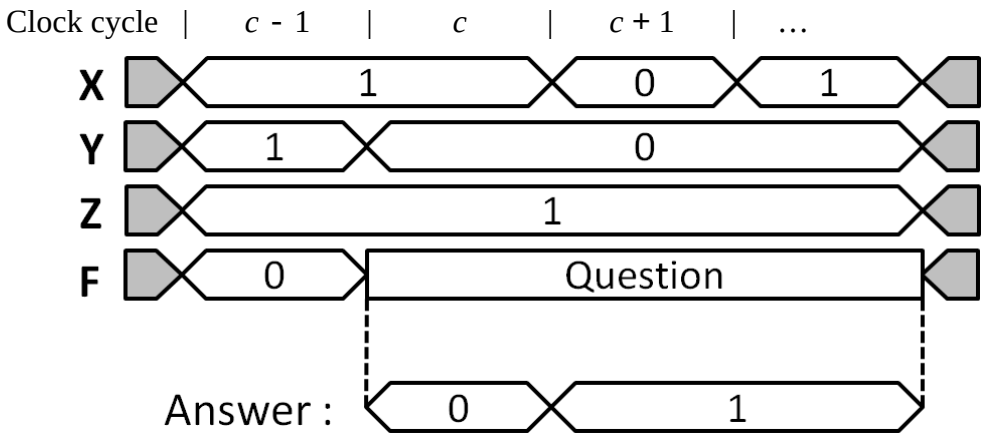
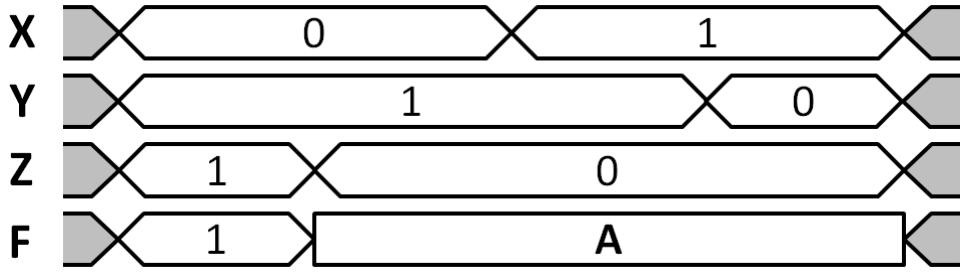
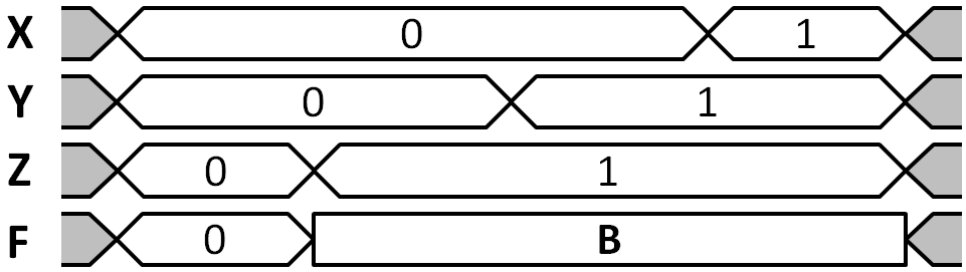


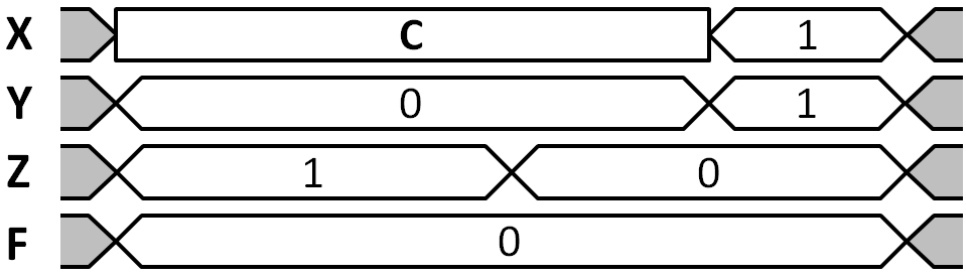
Figure 2 Sample input/output status of the logic circuit

Subquestion 1

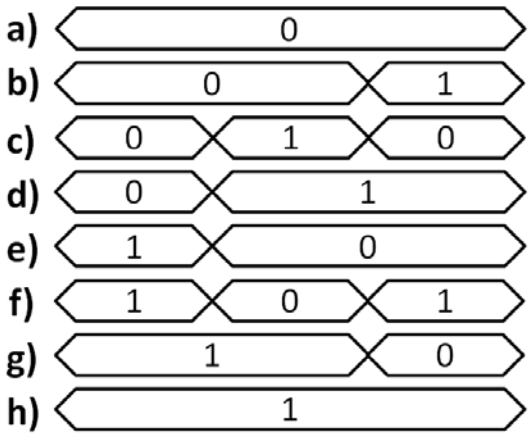
From the answer group below, select the correct answer to be inserted into each blank ☐ in the following three figures. If necessary, select the same answer twice or more.







Answer group



Subquestion 2

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

When the value 1 is obtained on the clock cycle *c* at the intermediate point T in Figure 1, it is always true that [D].

Answer group

a) the output F on the clock cycle *c* - 1 was 0
b) the output F on the clock cycle *c* - 1 was 1
c) both of inputs X and Y on the clock cycle *c* are 0
d) both of inputs X and Y on the clock cycle *c* are 1

Q2. Read the following description concerning software requirements definition, and then answer Subquestions 1 and 2.

The ABC bookstore offers a Web system that allows its customers to search, reserve, buy and rent books online. Any customer can visit the bookstore's official Web site to search for book details, but only the registered customers (hereinafter, members) can reserve, buy and rent books online. The required information for a membership is the customer's detail information such as name, address, a unique social ID, phone number, e-mail address, and credit card number. Once the registration is completed, the system will send the confirmation notification to the customer's e-mail address.

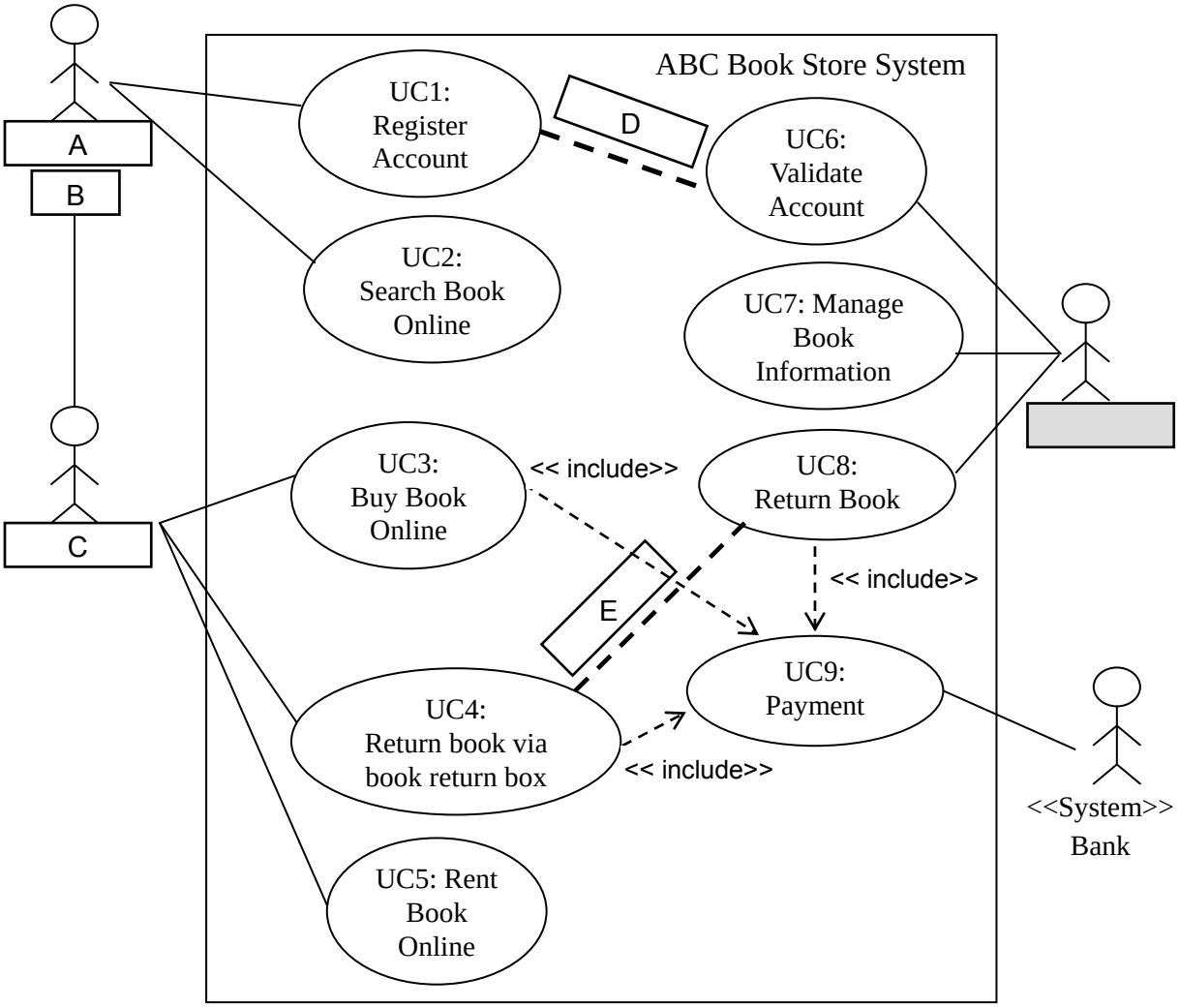
The ABC bookstore offers services for both hard-copied books and electronic books. The rental service is applied for hard-copied books only.

For hard-copied books, a customer can search for book information and reserve the books that he/she wants to buy or rent. The reserved time is 1 day. After the customer made the reservation, he/she must go to the bookstore to pick up the books within the reserved time. Otherwise, the reserved books will be released to other customers. To pick up the books, the customer must present the reservation number to the bookstore's staff and pay for the books.

The rental period is 7 days for each book. To return the books, the customer can either go to the bookstore during its office hours (8:00 to 22:00) to return them to the staff at the counter service, or use an automated self-service book return box that is located in front of the bookstore and is available for 24 hours. The book return box has a barcode reader attached to it. The customer has to use the bar code reader to scan a barcode of each book and then put the book into the book return box. If the customer returns the books later than the return date, he/she will be charged at $10 per day per book via his/her credit card. The late-returned fee will be shown in his/her membership account.

In case of buying books online (both hard-copied books and electronic books), the customer must make payment using his/her credit card and also provide a shipping address for the books delivery.

Figure 1 shows a use case diagram for the ABC bookstore system.



Note: Shaded part [ ] is not shown

Figure 1 A use case diagram for the ABC bookstore system

Subquestion 1

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

Answer group for A and C

a) Customer b) Member c) Staff

Answer group for B

a) ↑ b) △ c) ◇

Answer group for D and E

a) <<extend>> ----------> b) <---------- <<extend>>

c) <<include>> ----------> d) <---------- <<include>>

Subquestion 2

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

Figure 2 shows a high-level class diagram that implements the ABC bookstore system.

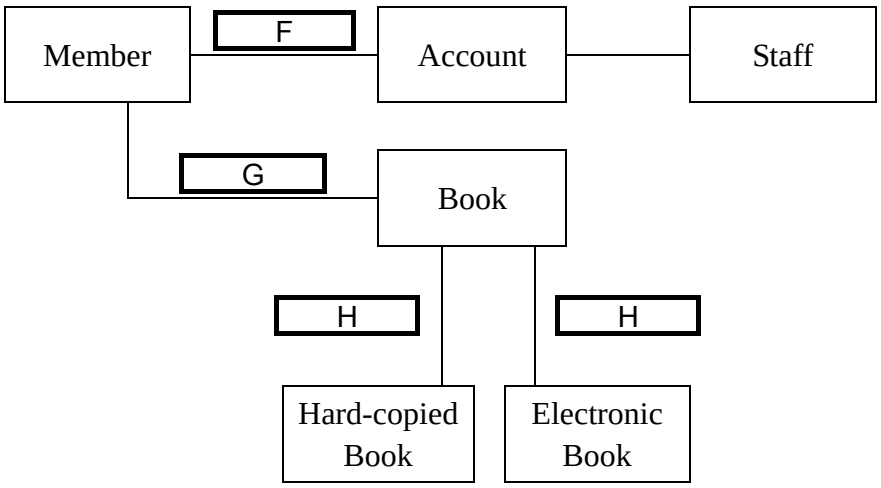


Figure 2 A high-level class diagram for the ABC bookstore system

Answer group

a) Aggregation relationship
b) Composition relationship
c) Generalization relationship
d) Many-to-many association relationship
e) One-to-many association relationship
f) One-to-one association relationship

Q3. Read the following description concerning a database design, and then answer Subquestions 1 through 3.

A restaurant inventory system makes use of the following tables.
Ingredients or items received are stored and recorded in DELIVERY table. Each delivery from a supplier is assigned a unique LOT_NO with one or several items in it. The item is identified by an ITEM_ID which serves as the primary key of ITEM table.

DELIVERY Table

LOT_NO	
ITEM_ID	Item identifier (primary key of ITEM table)
SUPPLIER	Code of the supplier
ARRIVED	Delivery date of the item/stock
EXPIRY	Expiry date of the item/stock
QTY_DLV	The quantity delivered

A request form is issued by the kitchen and recorded into REQUEST table. It has a unique REQUEST_NO and one or several items identified by ITEM_ID. All items conform to the same unit of measure used in DELIVERY table. Once the quantity needed and quantity released is equal, the request for the item has been fulfilled.

REQUEST Table

REQUEST_NO	
ITEM_ID	Item identifier (primary key of ITEM table)
QTY_REQ	The quantity needed
QTY_REL	The quantity released by the stock room to fulfill the request

A person takes the request form and releases the stocks needed from the stock room. Each item released passes a bar code scanner, and the system generates a unique RELEASE_ID and adds a new release record into RELEASE table. The person indicates the request being filled up. A request item can come from different lots. At the end of each day, expired items or those with poor quality are released and disposed of.

RELEASE Table

RELEASE_ID	Primary key of RELEASE table
LOT_NO	Lot number of the item being released
ITEM_ID	Item identifier (primary key of ITEM table)
REQUEST_NO	Corresponding request being fulfilled (NUL if item is expired or rejected)
QTY_REL	The quantity released
RELEASE_BY	Employee-id of the person who released the stock item
RELEASED	Release date of the item
STATUS	0 – Fill request; 1 – Item expired; 2 – Item rejected

ITEM table contains the list of items and its current stock-on-hand.

ITEM Table

ITEM_ID	Primary key of ITEM table
NAME	Name of the item
UNIT	Standard unit of measure
QTY_AT_HAND	The quantity at hand of the given item

The internal date format used by the system is YYYYNNN, where YYYY corresponds to the year while NNN corresponds on the n-th day of the year (1-366).

Subquestion 1

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

The following list shows the relationship between two tables when the left-side table (Table <from>) refers to the right-side table (Table <to>).

Table <from>	Relationship	Table <to>
DELIVERY	one-to-one	ITEM
ITEM	one-to-many	DELIVERY
RELEASE	[A]	DELIVERY
RELEASE	[A]	REQUEST
RELEASE	one-to-one	ITEM
REQUEST	[B]	RELEASE
DELIVERY	[B]	RELEASE
DELIVERY	[C]	REQUEST

The primary key(s) of DELIVERY table is(are) [D]. RELEASE table has several foreign keys. It has ITEM_ID as foreign key for ITEM table; [D] as foreign key(s) for DELIVERY table; ITEM_ID and REQUEST_NO as foreign keys for REQUEST table.

Answer group for A, B and C

a) one-to-one
b) one-to-many
c) many-to-many
d) no relationship

Answer group for D

a) ITEM_ID and LOT_NO
b) ITEM_ID and REQUEST_NO
c) ITEM_ID, LOT_NO and RELEASE_ID
d) ITEM_ID, RELEASE_ID and REQUEST_NO
e) LOT_NO
f) RELEASE_ID
g) REQUEST_NO

Subquestion 2

The quantity fields in ITEM and REQUEST tables should be properly maintained in a multi-user environment. To ensure correct updating, a record/field lock is issued at the start of the transaction then committed or rolled-back once the full transaction is completed. From the answer group below, select the statement which best describes the database operations of a full transaction to maintain the quantity fields.

Answer group

a) An INSERT to DELIVERY table should include an UPDATE to ITEM table to increase the QTY_AT_HAND in ITEM table.
b) An INSERT to RELEASE table should include an UPDATE to REQUEST table to increase the QTY_REL in REQUEST table.
c) An INSERT to REQUEST table should include an UPDATE to ITEM table to reduce the QTY_AT_HAND in ITEM table.
d) An UPDATE to DELIVERY table should always include an UPDATE to ITEM table.
e) An UPDATE to RELEASE table should always include an UPDATE to ITEM table and REQUEST table.

Subquestion 3

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

In order to avoid spoilage, the person releasing the items should prioritize the items with an earlier expiry date. DELIVERY table was modified to include a new field QTY_REL to determine how much of the item has already been released from the stock. The following SQL statement displays all the lot numbers and corresponding quantity remaining for a specific item. ITEM_ID of the item being selected is in the variable ITEM_LIST. The list displays the lots with an earlier expiry date first. Lots where the item was already released will no longer appear on the list.

```
SELECT LOT_NO, [ E ] AS QTY
   FROM DELIVERY
   WHERE ITEM_ID = #ITEM_LIST
   AND [ E ] > 0
   [ F ] EXPIRY [ G ]
```

Answer group for E

a) ARRIVED
b) ITEM_ID
c) QTY_DLV
d) QTY_REL
e) QTY_DLV – QTY_REL

Answer group for F and G

a) ASC
b) DESC
c) EARLIEST
d) ORDER BY
e) SORTED BY

Q4. Read the following description concerning network security, and then answer Subquestion.

Company A has built its corporate network shown in the Figure. Currently, there is a plan to enhance the security controls over the network in order to prevent the occurrence of security incidents and to resolve some network problems.

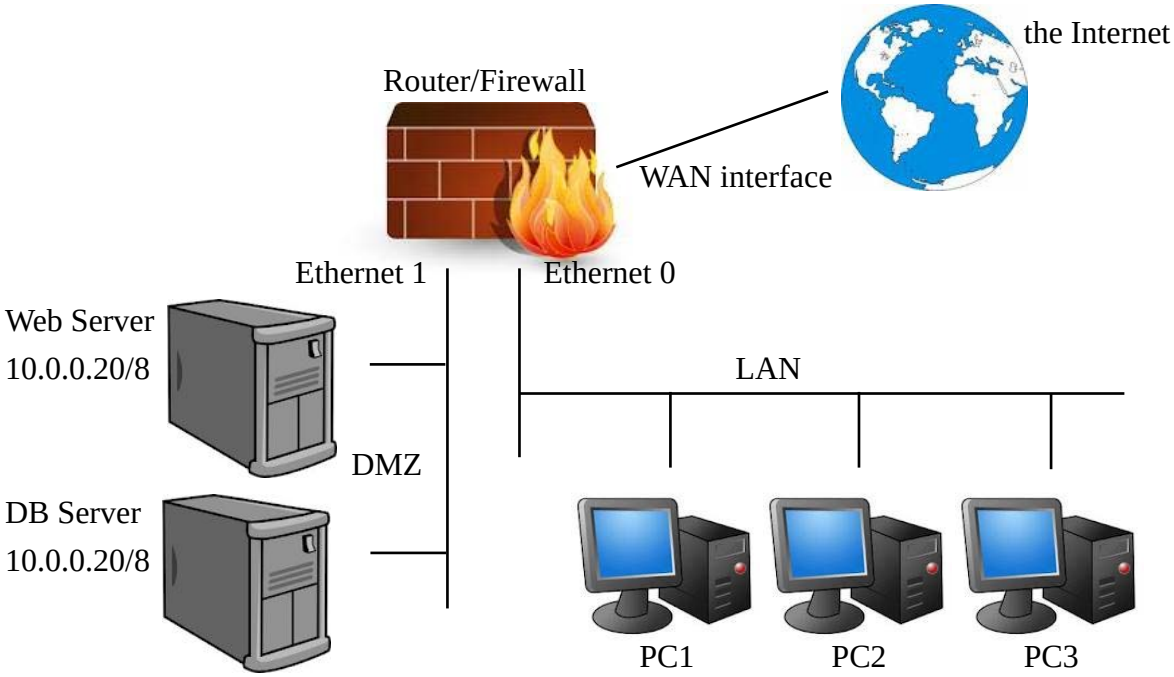


Figure Company A's network configuration

The table shows the TCP/IP addresses and subnet masks for PC1, PC2 and PC3.

Table IP addresses for PCs

PC name	TCP/IP address	Subnet mask
PC1	192.168.99.1	255.255.255.0
PC2	192.168.99.2	255.255.255.0
PC3	192.168.99.3	255.255.255.0

PCs on the LAN and also authenticated users over the Internet are able to access the Web Server and DB Server in DMZ. HTTP and HTTPS are used to access the Web Server.

The new security policy of Company A is activated. One requirement for improving internal network security is to monitor all unwanted and non-trusted actions from LAN to DMZ as well as packets and services offered from DMZ to LAN. This requirement enables the management system of Company A to issue appropriate alerts depending on the situations when malicious activities occur. The solution of [A] will meet this requirement.

Currently, telnet accesses from the Internet to the DB Server are blocked by configuring the firewall rules. Now, Company A wishes to block telnet accesses from the LAN to the DB Server too. For that, adding a rule entry of [B], which denies telnet accesses from the LAN to the DB Server, will meet this requirement.

Company A further wants to implement a measure to prevent DoS and DDoS attacks from the Internet. It has 3 public TCP/IP addresses that can be used to connect to the Internet, but only one of them is used at WAN interface in Router/Firewall. Two other addresses are for backup purpose.
First of all, Company A adopts a measure to prevent attackers from running any sort of "[C]" in DMZ, in order to avoid discovering the internal structure of publicly accessible areas of Company A's network. Then, if Company A's network is really under such an attack, the solution of [D] and replacing the TCP/IP address at WAN interface in Router/Firewall with the backup one will recover the network from the attack.

Subquestion

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

Answer group for A

a) setting the IDS sensor between Firewall and LAN
b) setting the IDS sensor between Firewall and the Internet
c) setting the IDS/IPS sensor between Firewall and LAN
d) setting the IDS/IPS sensor between Firewall and the Internet

Answer group for B

	Source	Destination	Port	Action	Interface
a)	10.0.0.0/8	192.168.99.0/24	23	Deny	Ethernet1 (In)
b)	10.0.0.20/8	192.168.99.0/24	23	Deny	Ethernet1 (Out)
c)	192.168.99.0/22	10.0.0.20/8	23	Deny	Ethernet0(Out)
d)	192.168.99.0/24	10.0.0.0/8	23	Deny	Ethernet0 (In)
e)	192.168.99.0/24	10.0.0.20/8	23	Deny	Ethernet0 (In)

Answer group for C

a) file transfer
b) port scan
c) SQL injection
d) traceroute

Answer group for D

a) blocking HTTP spoofing
b) blocking ICMP at the Firewall
c) blocking SSL authentication TCP/IP request at DMZ which use HTTP/ HTTPS
d) blocking TCP at the Firewall

Q5. Read the following description concerning a teaching permit renewal system, and then answer Subquestions 1 and 2.

In a certain country, the ministry of higher education is responsible for the maintenance of teaching permits among the lectures in the country.

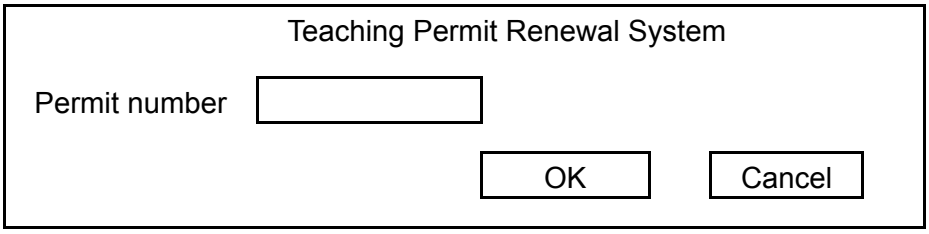
The lecturer has to register for the permit. The name, address, phone number and registration date are recorded, and the teaching permit is registered. The permit is not issued until the applicant has proof of passing the post-graduate examination (Masters or PhD). Registration and issue of the permit is free.

The permit number identifies each permit holder. When the permit is issued, the issue date is recorded. There are different types of permit class. The permit class represents the type of tutorial classes the holder can teach. Higher-level permit class automatically allows the holder to teach lower tutorial classes.

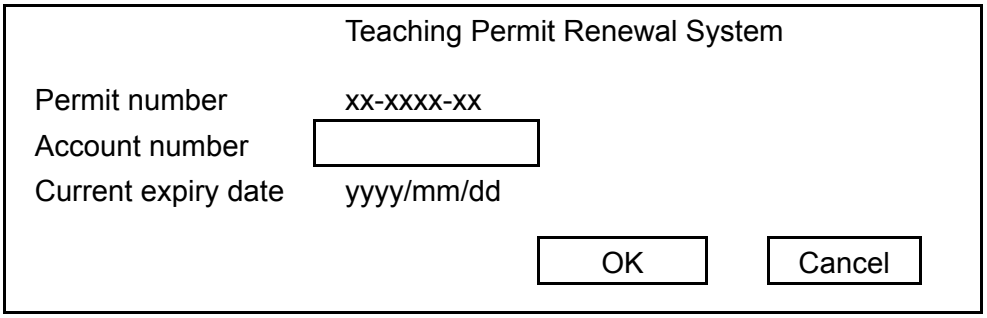
Each teaching permit has a 1-year valid period effective from the issue date. The current expiry date is recorded in the permit. The permit holder has 30 days before the current expiry date to renew his/her permit. When he/she renews the permit, the renewal date is recorded, and the current expiry date is copied to the expiry date. Then, the current expiry date is extended by 1 year. The list of expired permits is printed out each day.

A record of each renewal identified by the renewal code is kept. The permit holder has to pay for each renewal. The permit holder can pay for the renewal through an Internet payment system or a credit card. The card number, issuing bank, issue date, card expiry date and the renewal code are recorded for credit card payments. The permit holder needs to give an account number if he/she pays by the Internet payment system.

The screen of the teaching permit renewal system is shown below. The permit holder enters his/her permit number, and then clicks the OK button.



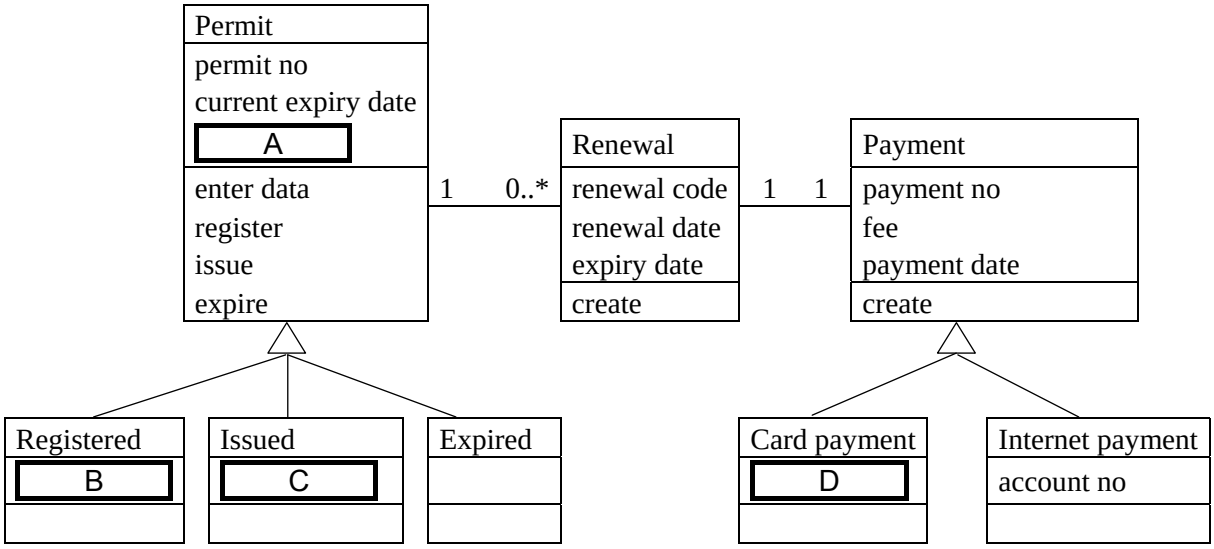
When he/she selects the Internet payment system from the menu for his/her payment, the following screen is shown.



The permit number and current expiry date are shown on the screen. When the permit holder enters the account number and clicks the OK button, a confirmation dialog is shown where he/she can continue with the processing or cancel the transaction.

Subquestion 1

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

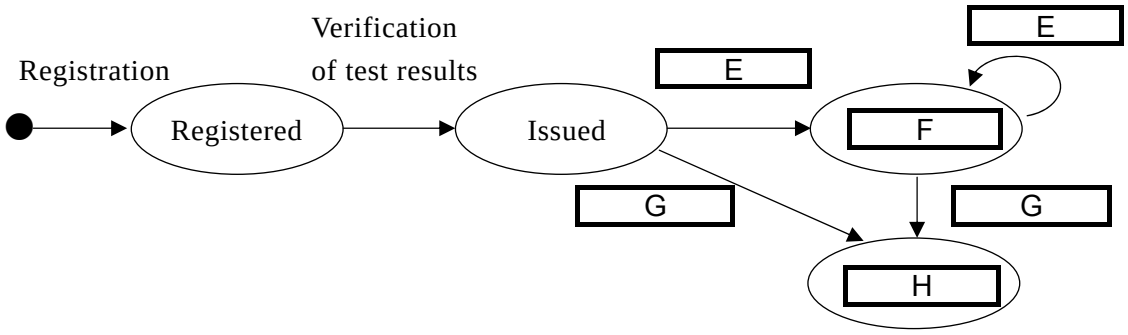


Answer group

a) card number, issuing bank, issue date
b) card number, issuing bank, issue date, card expiry date
c) issue date
d) name, address, phone number
e) name, address, phone number, permit class
f) permit class
g) registration date

Subquestion 2

From the answer group below, select the correct answer to be inserted into each blank ☐ in the following state transition diagram.



Answer group

a) Checking of expired permits
b) Entering account number
c) Entering permit number
d) Expired
e) Register permit
f) Renew permit
g) Renewed

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

The subprogram `LRU()` simulates the popular page replacement operation by using LRU algorithm that is used by OS for virtual storage controls.

In virtual storage controls, in order to achieve system processing efficiently, pages with a high application frequency are permanently loaded in a main storage unit, while pages with a low application frequency are stored in an external storage unit and are loaded into the main storage unit only when they are needed.

LRU stands for "Least Recently Used". In this method, when a required page is not found in the main storage unit (i.e. a "page fault" occurs), the page for which the time elapsed since it was used the last time is the longest among the pages in the main storage unit is sent out to the external storage unit, and then the required page is loaded into the main storage unit.

[Program Description]

1) The test data used to simulate the page replacement operation are provided as initial values for global integer array `PageReq[]` and global integer variables `NumPageReq` and `NumFrame`.
2) `PageReq[]` contains a list of required page numbers. Page numbers are picked up one by one from the top. `NumPageReq` contains the total number of pages in `PageReq[]`.
3) `NumFrame` contains the total number of page frames.
4) `Frame[]` contains the page numbers that are currently loaded in the main storage unit. Initial value `-1` means that the frame is empty. `Count[]` is parallel with `Frame[]`, and it contains the values which represent the time elapsed since the corresponding page number contained in `Frame[]` was used the last time. For example, when `Frame[2]=4` and `Count[2]=3`, it indicates that the page number currently loaded in `Frame[2]` is `4`, and the time elapsed since this page was used the last time is `3`.
5) Indexes of arrays start at 1.
6) `Fault` counts the occurrence of page faults.
7) The subprogram `print()` displays the contents of arguments in a new line. Arguments can be variables, arrays or constants.

The following list shows the execution result of the program.

```
i NextPage Fault  Frame[ ]  Count[ ]
1     1      0     1 -1 -1   1  0  0
2     2      0     1  2 -1   2  1  0
3     3      0     1  2  3   3  2  1
4     4      1     4  2  3   1  3  2
5     2      1     4  2  3   2  1  3
6     3      1     4  2  3   3  2  1
7     4      1     4  2  3   1  3  2
8     5      2     4  5  3   2  1  3
```

[Program]

```
O Global Integer: NumFrame ← 3, Frame[3], Count[3]
O Global Integer: NumPageReq ← 8, PageReq[8] ← { 1,2,3,4,2,3,4,5 }

O Subprogram LRU( )
  O Integer: i, j, Fault, Found, NextPage

  ■ i: 1, i ≤ NumFrame, 1     /* Initialize Frame[] and Count[] */
    • Frame[i] ← -1
    • Count[i] ←  0
  ■
  • print("  i NextPage Fault  Frame[ ]  Count[ ]")
  • Fault ← 0
  ■ i: 1, i ≤ NumPageReq, 1
    • NextPage ← PageReq[i]
    • Found ← 0
    ■ j: 1, j ≤ NumFrame and Found = 0, 1
      ▲ Frame[j] = NextPage or Frame[j] = -1
        • Found ← j
    ■
    ▲ Found = 0
      • [ A ]
      • Found ← 1
      ■ j: 2, j ≤ NumFrame, 1
        ▲ [ B ]
          • Found ← j
      ■
    •
    ■ j: 1, j ≤ NumFrame, 1
      ▲ [ D ]
        • Count[j] ← 1
      ▲ Frame[j] ≠ -1
        • Count[j] ← Count[j] + 1
    ■
    • print(i, NextPage, Fault, Frame[], Count[])
  ■
/* End of Subprogram LRU */
```

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 and C

a) `Count[Found] ← 1`
b) `Count[i] ← Count[i] + 1`
c) `Fault ← 1`
d) `Fault ← Fault + 1`
e) `Frame[Found] ← NextPage`
f) `Frame[i] ← NextPage`

Answer group for B and D

a) `Count[j] < Count[Found]`
b) `Count[j] > Count[Found]`
c) `Frame[Found] = -1`
d) `Frame[Found] ≠ -1`
e) `j = Found`
f) `j ≠ Found`

Subquestion 2

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

In order to simulate another case, the initial values of integer array `PageReq[]` are changed from `{ 1,2,3,4,2,3,4,5 }` to `{ 1,2,3,4,1,1,2,3 }`, and then, the subprogram `LRU()` is executed. The following list shows the execution result.

i	NextPage	Fault	Frame[ ]	Count[ ]
1	1	0	1 -1 -1	1 0 0
2	2	0	1 2 -1	2 1 0
3	3	0	1 2 3	3 2 1
4	4	1	4 2 3	1 3 2
5	1		4 1 3	2 1 3
6	1		F	
7	2	E	G	
8	3			

Note: Shaded parts are not shown.

Answer group for E

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

Answer group for F and G

a) 4 1 1 3 2 1
b) 4 1 2 3 2 1
c) 4 1 2 4 2 1
d) 4 1 3 2 1 3
e) 4 1 3 3 1 4
f) 4 2 1 3 1 2
g) 4 2 1 4 1 3

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark Ⓢ 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]

The program reads a sequence of English words from an input file, and prints out each word with the number of vowels, consonants, digits and others contained in the word. At the end, it prints out the total number of words.

(1) Words are separated by a space character, a new-line character, or an end-of-file marker. In this question, these three codes are called separators.
(2) Vowels are alphabetical characters `A`, `a`, `E`, `e`, `I`, `i`, `O`, `o`, `U` and `u`.
(3) Consonants are alphabetical characters other than vowels.
(4) Digits are numerical characters `0` through `9`.
(5) Others are any characters other than alphabetical characters, numerical characters, and separators.
(6) The input file is a sequential file named `FILE.TXT`.
(7) It is assumed that the length of each word is less than or equal to 15.
(8) For example, when the input file contains the words "`ITPEC FE-Exam Oct. 2012`", the program prints out the following list.

```
Word           Vowel Consonant Digit Other
ITPEC              2         3     0     0
FE-Exam            3         3     0     1
Oct.               1         2     0     1
2012               0         0     4     0

Number of word(s): 4
```

[Program] Note: Reference markers ← (1), ← (2), … are not part of the program.

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

enum v_c_d_o {vowel, consonant, digit, other};

typedef enum v_c_d_o v_c_d_o;

void cnt_char(v_c_d_o type,
              int *ptr_v, int *ptr_c, int *ptr_d, int *ptr_o);

void main()
{
  int  ch;                              /* current character */
  char *filename = "FILE.TXT";          /* name of the input file */
  FILE *fpt;                            /* file pointer */
  int  cnt_word;                        /* word counter */
  int  cnt_v, cnt_c, cnt_d, cnt_o;      /* character counters */
  int  word_pos;
  char word[16];

  fpt = fopen(filename,"r");
  if (!fpt)
  { printf("\n%s ... FILE NOT FOUND.", filename);
    exit(0);
  }

  printf("\nWord           Vowel Consonant Digit Other\n");
  cnt_word = 0;
  cnt_v = cnt_c = cnt_d = cnt_o = 0;
  word_pos = 0;
  strcpy(word, "               ");
  ch = ' ';                                          ← (1)

  while (      A       )
  { ch = fgetc(fpt);
    if (       B       )
    { if (word_pos > 0)                              ← (2)
      { printf("%15s%4d%10d%6d%6d\n",
               word, cnt_v, cnt_c, cnt_d, cnt_o);
        cnt_word = cnt_word + 1;
        cnt_v = cnt_c = cnt_d = cnt_o = 0;
        word_pos = 0;
        strcpy(word, "               ");          ← (3)
      }
    }
```

```
    else
    { word[      C       ] = ch;
      if (isalpha(ch))                                    ← (4)
      { switch (      D       )
        { case 'a': cnt_char(vowel, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
          case 'e': cnt_char(vowel, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
          case 'i': cnt_char(vowel, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
          case 'o': cnt_char(vowel, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
          case 'u': cnt_char(vowel, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
          default:  cnt_char(consonant,
                                    &cnt_v, &cnt_c, &cnt_d, &cnt_o);
                    break;
        }
      }
      else
        if (      E       )
          cnt_char(digit, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
        else
          cnt_char(other, &cnt_v, &cnt_c, &cnt_d, &cnt_o);
    }
  }
  printf("\nNumber of word(s): %d\n", cnt_word);
  fclose(fpt);
}

void cnt_char(v_c_d_o type,
              int *ptr_v, int *ptr_c, int *ptr_d, int *ptr_o)
{
  switch (type)
  { case vowel:     ++*ptr_v;
                    break;
    case consonant: ++*ptr_c;
                    break;
    case digit:     ++*ptr_d;
                    break;
    case other:     ++*ptr_o;
                    break;
    default:        printf("\nPROGRAM ERROR.\n");
                    exit(0);
  }
}
```

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 and B

a) `ch != ' ' && ch != '\n'`
b) `ch == ' ' || ch == '\n'`
c) `ch != EOF`
d) `ch == EOF`
e) `ch != ' ' && ch != '\n' && ch != EOF`
f) `ch == ' ' || ch == '\n' || ch == EOF`

Answer group for C

a) `++cnt_word`
b) `++word_pos`
c) `cnt_word++`
d) `word_pos++`

Answer group for D and E

a) `ch`
b) `isdigit(ch)`
c) `islower(ch)`
d) `isupper(ch)`
e) `tolower(ch)`
f) `toupper(ch)`

Subquestion 2

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

Among the statements a) through f) in the answer group below about the behavior of the above program, the correct statements are [F] and [G].

Answer group

a) On line ← (1), when "`ch = ' ';`" is changed to "`ch = 'A';`", the constant `'A'` is processed as the first character of the first word in the input file.
b) On line ← (1), when "`ch = ' ';`" is changed to "`ch = 'A';`", the program works correctly.
c) On line ← (2), when "`if (word_pos > 0)`" is changed to "`if (word_pos > 1)`", the program works correctly.
d) On line ← (3), when "`strcpy(word, "");`" is changed to "`word[0] = ' '; word[1] = '\0';`", the program works correctly.
e) On line ← (4), when "`if (isalpha(ch))`" is changed to "`if ('a' <= ch && ch <= 'z')`", the program works correctly.
f) When the words in the input file are separated by consecutive space characters, for example "`ITPEC  FE-Exam  Oct.  2012`", the program works correctly.

Q8. Read the following description of a Java program and the program itself, and then answer Subquestion.

[Program Description]

There are an interface `ListI` and a class `LinkedList`. The class `LinkedList` implements the interface `ListI`. Here is double linked list implemented.
The class `LinkedList` requires an auxiliary class `Node`, which represents the nodes in double linked list. The class `Node` is a nested class of the class `LinkedList`. The class `Node` is to be designed so that any subclass of `LinkedList` is able to access to it.
To test the class `LinkedList`, a class `LinkedListTest` is implemented. When this program is executed, the following list will be printed out.
The concurrency (thread-safety) issue is not considered in the program.

```
First time
[0] = 2
[1] = 1
[2] = 3
[3] = 7
[4] = 6
[5] = 5
[6] = 4

Second time
[0] = 1
[1] = 3
[2] = 6
[3] = 5

Cloned list
[0] = 1
[1] = 3
[2] = 6
[3] = 5

Original list
[0] = 1
[1] = 3
[2] = 6
[3] = 5

Cloned list
[0] = 3
[1] = 6
[2] = 5
```

[Program]

```
/* Interfaces for implementation of the linked list. */
public interface ListI {
   public int getSize();       /* returns the number of elements in the list */

   public boolean isEmpty();   /* returns true if and only if the list is empty */

   public Object getElementIndexed(int i);   /* returns the i-th element
                                                in the list. */
   public Object getFirstElement();   /* returns the first element in the list */

   public Object getLastElement();    /* returns the last element in the list */

   public void insertItemAt(Object item, int i);   /* inserts a new element
                                                      at the i-th position of the list */
   public void insertFirst(Object item);    /* inserts a new element
                                               at the first of the list */
   public void insertLast(Object item);     /* inserts a new element
                                               at the end of the list */
   public Object removeElementAt(int i);    /* remove the element
                                                at the i-th position of the list */
   public Object removeFirst();   /* remove the element at the first of the list */

   public Object removeLast();    /* remove the element at the end of the list */
}

/* The double linked list class "LinkedList" implements the ListI and Cloneable interfaces of
   a List, which is an ordered collection of elements that can be accessed through an index.
   The size of a LinkedList can grow as needed. */

public class LinkedList implements ListI, Cloneable {
   public LinkedList() {
      refFirst = refLast = null;
      count = 0;
   }
   public int getSize() {
      return count;
   }
   public boolean isEmpty() {
      return (      A      );
   }
   public Object getElementIndexed(int i) {
      Node node = refFirst;
      for (int j = 0;       B       ; j++) {
         node = node.refNext;
      }
      return node.refItem;
   }
```

```
public Object getFirstElement() {
   Object result = (refFirst != null ? refFirst.refItem : null);
   return result;
}
public Object getLastElement() {
   Object result = (refLast != null ? refLast.refItem : null);
   return result;
}

/* The result of the toString() method should include all the fields of the object. */
public String toString() {
   StringBuffer s = new StringBuffer();
   int i = 0;
   for (Node node = refFirst;
        node != null; node = node.refNext, i++) {
      s.append("[" + i + "] = " + node.refItem + "\n");
   }
   return s.toString();
}

/* Two lists are considered as equal if their lengths are the same and
   the elements at the same position of the two lists are pair wise equal. */
public boolean equals(Object other) {
   if (other != null && other C LinkedList) {
      LinkedList otherlist = (LinkedList) other;
      if (this.getSize() == otherlist.getSize()) {
         Node thisnode = this.refFirst;
         Node othernode = otherlist.refFirst;
         while (thisnode != null && othernode != null) {
            if (!thisnode.refItem.equals(othernode.refItem))
               return false;
            thisnode = thisnode.refNext;
            othernode = othernode.refNext;
         }
         return true;
      }
   }
   return false;
}
```

```
/* The clone() method creates a deep copy of the list. A deep copy means
   that the objects referenced by the fields of reference types are cloned. */
public [ D ] clone() throws CloneNotSupportedException {
   LinkedList list = (LinkedList) super.clone();
   list.refFirst = list.refLast = null;
   list.count = 0;
   for (Node node = refFirst; node != null; node = node.refNext) {
      if (node.refItem != null) {
         list.insertLast(node.refItem);
      }
   }
   return list;
}

public void insertItemAt(Object item, int i) {
   if (i <= 0) {
      insertFirst(item);
   } else if (i >= count) {
            insertLast(item);
         } else {
            Node n = refFirst;
            for (int j = 0; n != null && j < i - 1; j++) {
               n = n.refNext;
            }
            Node node = new Node();
            node.refItem = item;
            [ E ];
            node.refPrev = n;
            node.refNext.refPrev = node;
            n.refNext = node;
            count++;
         }
}

public void insertFirst(Object item) {
   Node node = new Node();
   node.refItem = item;
   node.refNext = refFirst;
   node.refPrev = null;
   if (refFirst == null)
      refLast = node;
   else
      refFirst.refPrev = node;
   refFirst = node;
   count++;
}
```

```
public void insertLast(Object item) {
   Node node = new Node();
   node.refItem = item;
   node.refNext = null;
   node.refPrev = refLast;
   if (refLast != null)
      refLast.refNext = node;
   else
      refFirst = refLast = node;
   refLast = node;
   count++;
}

public Object removeElementAt(int i) {
   Object result = null;
   if (i <= 0) {
      result = removeFirst();
   } else if (i >= count) {
              result = removeLast();
          } else {
            Node n = refFirst;
            for (int j = 0; n != null && j < i - 1; j++)
               n = n.refNext;
            result = n.refNext.refItem;
            n.refNext = n.refNext.refNext;
            n.refNext.refPrev = n;
            count--;
         }
   return result;
}

public Object removeFirst() {
   Object result = null;
   if (refFirst != null) {
      result = refFirst;
      refFirst = refFirst.refNext;
      if (refFirst != null)
         refFirst.refPrev = null;
      else
         refLast = null;
      count--;
   }
   return result;
}
```

```
   public Object removeLast() {
      Object result = null;
      if (refLast != null) {
         result = refLast;
         refLast = refLast.refPrev;
         if (refLast != null)
            refLast.refNext = null;
         else
            refFirst = null;
         count--;
      }
      return result;
   }

   protected Node refFirst, refLast;
   protected int count;

   [ F ] class Node {
      Object refItem;
      [ G ] refNext;
      [ G ] refPrev;
   }
}

public class LinkedListTest {

   public static void main(String args[])
          throws CloneNotSupportedException {
      LinkedList l = new LinkedList();
      l.insertFirst(new Integer(1));
      l.insertFirst(new Integer(2));
      l.insertLast(new Integer(3));
      l.insertLast(new Integer(4));
      l.insertItemAt(new Integer(5), 3);
      l.insertItemAt(new Integer(6), 3);
      l.insertItemAt(new Integer(7), 3);

      System.out.println("First time");
      System.out.println(l);

      l.removeFirst();
      l.removeLast();
      l.removeElementAt(2);

      System.out.println("Second time");
      System.out.println(l);

      LinkedList l2 = (LinkedList) l.clone();
      System.out.println("Cloned list");
      System.out.println(l2);
```

```
        l2.removeFirst();
        System.out.println("Original list");
        System.out.println(l);
        System.out.println("Cloned list");
        System.out.println(l2);
    }
}
```

Subquestion

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

Answer group for A and B

a) `count`
b) `count < 0`
c) `count == 0`
d) `count > 0`
e) `node != null && j < i`
f) `node != null && j > i`
g) `node != null || j < i`
h) `node != null || j > i`

Answer group for C and D

a) `(instanceOf)`
b) `elementOf`
c) `Instance`
d) `instanceOf`
e) `instanceof`
f) `LinkedList`
g) `Node`

Answer group for E

a) `n.refNext = n.refNext`
b) `node = n.refNext`
c) `node.refNext = n.refNext`
d) `node.refPrev = n.refNext`

Answer group for F and G

a) `List`
b) `ListElement`
c) `Node`
d) `Object`
e) `private`
f) `protected`