2003　VITEC

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1-Q5	Q6-Q9	Q10-Q13
Question Selection	**Compulsory**	**Select 1 of 4**	**Select 1 of 4**
Examination Time	**13:00 ~ 15:30 (150 minutes)**		

Instructions:

1. Use an HB 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 test 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（Q6-Q9 and Q10-Q13）

   Mark the ⓢ of the question you select to answer in the “Selection Column” on your answer sheet.

   (4) Answers

   Mark your answers as shown in the following sample question.

   [Sample Question]

   In which month is this Fundamental IT Engineer Examination conducted?

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

   Since the correct answer is “b” (October), mark your answer sheet as follows:

   [Sample Reply]

SQ1	ⓐ	●	ⓒ	ⓓ

3. “Assembly Language specifications” are provided as a reference at the end of this booklet.

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

Company names and product names appearing in the test questions are trademarks or registered trademarks of their respective companies. Note that the ® and ™ symbols are not used within.

Questions 1 through 5 are compulsory. Answer every question.

Q.1 Read the following description of an IP(Internet Protocol) address and answer the Subquestion.

Thirty-two-bit (32-bit) addresses called IP addresses are used in the Internet protocol. IP addresses are assigned to each host so that they can be uniquely identified in the network to which they are to be connected.

An IP address is composed of the network part and the host part, as shown in Figure 1, and is categorized into three classes, A, B and C, depending on the size of the network.

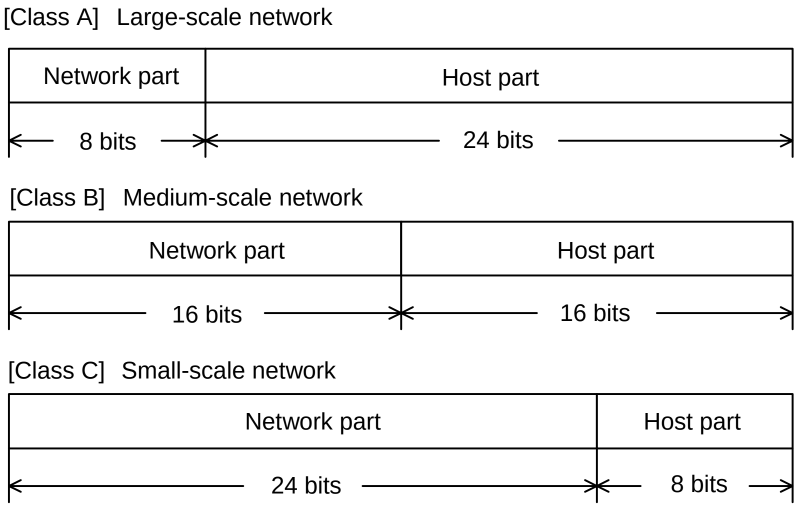


Figure 1: Constitution of an IP address

A university has decided to establish its own intra-campus network. This network uses Class B IP addresses. The expected number of hosts to be connected to this network is shown in the following table.

Table: Expected number of hosts to be connected to the intra-campus network

Department	Number of hosts
Science	2,100
Engineering	1,600
Economics	1,400
Business Administration	1,000
Law	800
Education	600
Literature	600
Total	8,100

Subquestion

Select from the answer group below the correct answers to insert in the blanks $\boxed{}$ in the following description.

(1) The minimum required number of bits to uniquely identify each host in this network is $\boxed{\text{a}}$.

(2) A portion of the host part is used as a subnet address. The use of the subnet address allows a network to be regarded as a collection of subdivided networks. Figure 2 shows the configuration of the subnet.

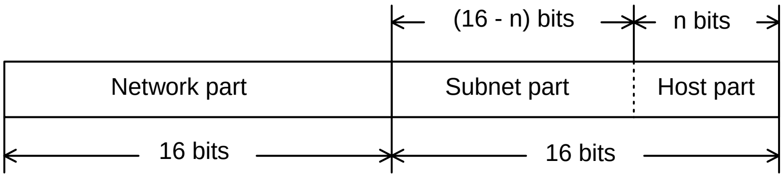


Figure 2: Configuration of Subnet

Note that each portion shown in Figure 2 will not be used as an address if all bits are zero or all bits are 1.

(3) In this network, departments are indicated in the subnet part. The host part is to consist of the minimum number of bits to indicate the number of hosts owned by the department that has the longest number of hosts. This number of bits in this part is $\boxed{\text{b}}$.

(4) Assigning the number of bits obtained in (3) to the host part, the upper limit of the number of departments that can be indicated by the subnet part is $\boxed{\text{c}}$.

Answer group for a through c:

a) 4 b) 8 c) 10 d) 11 e) 12

f) 13 g) 14 h) 15 i) 16 j) 32

Q.2 Read the following description of inventory management, then answer Subquestions 1 and 2.

The plant where Mr. Y works has recently decided to improve its inventory management systems, which previously depended on the experience and hunches of the manager in charge of inventory. Mr. Y has been asked to give his recommendations concerning the raw materials ordering system.

Subquestion 1.

There are 100 different raw materials used at the plant where Mr. Y works. Mr. Y decided to select the most vital items and propose a review of inventory management for those items. After studying various documents, it became clear that the best approach would be to closely manage (as vital items) those materials for which the plant's annual expenditures (unit cost of the raw material x amount consumed yearly) were high, while expending little on inventory management for those materials for which annual expenditures were low. After researching the annual expenditures for each raw material, Mr. Y produced the following table:

No.	Item code	Unit cost	Annual consumption	Annual expenditure
1	AA01	30	56,380	1,691,400
2	AA07	200	1,500	300,000
3	AC01	1,500	23,400	35,100,000
⋮	⋮	⋮	⋮	⋮
100	ZQ80	10	2,875	28,750
Total annual expenditure				231,730,960

Next, in order to select from the table as vital items those materials that accounted for the largest percentage of the annual expenditures, Mr. Y analyzed the data according to the following procedure:

① He sorted all of the materials in the ascending order of the annual expenditures.
② He determined the cumulative annual expenditure from the top to each place (row) in the table.
③ He drew a bar graph with the place of each material on the x axis, and the annual expenditure on the y axis.
④ He plotted a graph, indicating the cumulative annual expenditure from place to place.

From the answer group below, select the correct name for this type of chart or graph.

Answer group:

a) Arrow diagram b) Gantt chart c) Control chart
d) Pareto diagram e) Portfolio graph

Subquestion 2.

Typical inventory management systems include the "fixed interval ordering system" and the "fixed size ordering system." In order to consider which of these systems might be best suited for the list of vital items he created in Subquestion 1, Mr. Y summarized the characteristics of each system. From the answer group below, select the correct answers to insert in the blanks □ in the following description.

In order to select an ordering method, it is necessary to consider numerous factors, including the unit costs of the raw materials, the amount of raw materials consumed during a given period, the size of fluctuations in consumption and the ease with which such fluctuations can be forecast, the length of the procurement lead time (the time from ordering a material to receiving it into the stock), costs associated with ordering (such as the transportation costs and administrative costs incurred for one order), and the cost of maintaining inventory.

Because the "fixed interval ordering system" requires determining a regular interval for ordering based on the raw material procurement lead time, forecasting the consumption of the raw material over the period following each regular interval for ordering, and then determining $\boxed{\text{a}}$ on the basis of that information, it becomes possible to manage inventory very closely. This method carries little risk of materials' being out of stock even when there are large fluctuations in consumption. Also, this method is effective if there are large changes in demand in the market for a product, resulting in frequent changes to the production plan. It is effective,too, for managing inventories of raw materials whose unit costs are high.

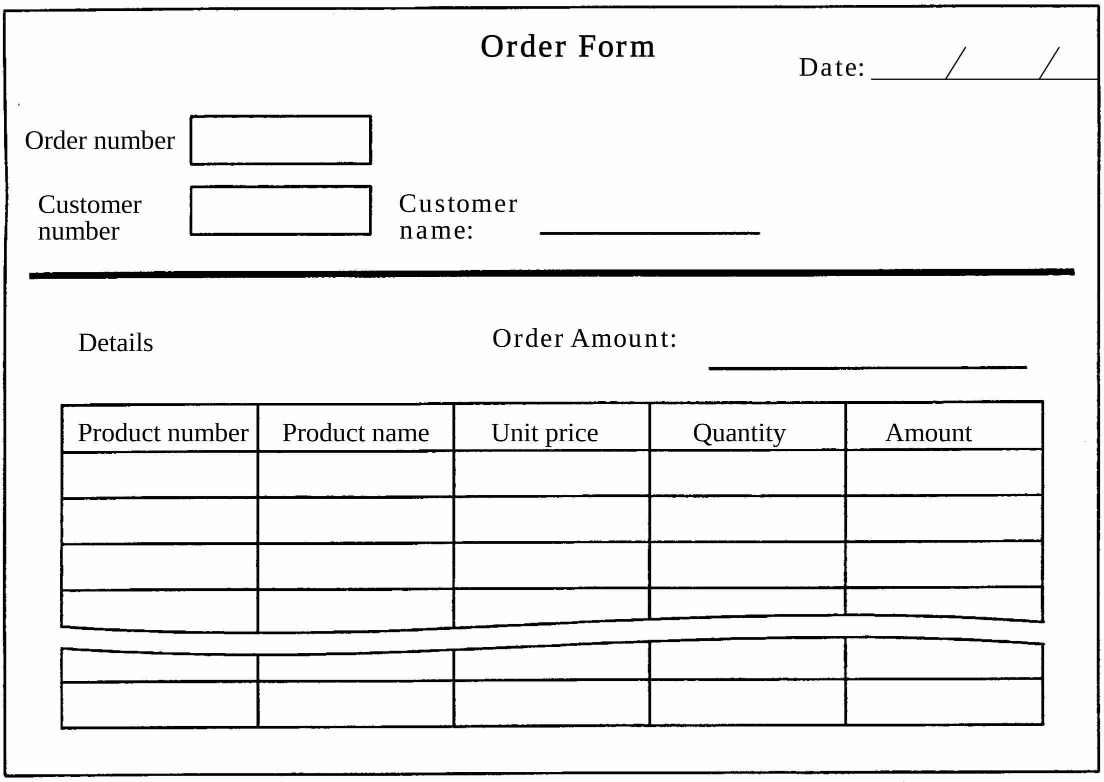
The "fixed size ordering system" orders quantities (predetermined on the basis of the procurement lead time) of materials that have fallen to $\boxed{\text{b}}$ Compared to the "fixed interval ordering system," this method is simpler. However, there is a risk of raw materials' being out of stock if there is a large fluctuation in consumption. This method is suitable for the management of materials when product demand and the raw material procurement lead time are stable, and when it is possible to accurately determine the current inventory level at any given time.

Answer group for a and b:

a) the safety stock   b) 50% of the safety stock   c) the ordering timing
d) the order point   e) the order quantity   f) the average inventory level

Q.3 Read the following text regarding a relational database, then answer Subquestions 1 and 2.

Assume a table forming a relational database is created based on the following order form, and data searches are performed using SQL.



Order Form

Date: ____/____/____

Order number □

Customer number □ Customer name: __________

Details

Order Amount: __________

Product number	Product name	Unit price	Quantity	Amount

Note: A single order is recorded on a single order form and a unique order number is assigned to each order form.

Subquestion 1.

From the answer group below, select the correct answers to be inserted in the blanks □ below.

Order_table

Order_number	$\boxed{\text{a}}$	Date

Order_details_table

Order_number	Product_number	$\boxed{\text{b}}$

Customer_table

Customer_number	Customer_name

Product_table

Product_number	Product_name	Unit_price

Answer group for a and b:

a) Amount	b) Customer_number	c) Customer_name
d) Quantity	e) Product_number	f) Product_name

Subquestion 2.

An order amount table is defined as a view table for checking the order amount of each order form. This was used to create SQL statements to search for the following information. From the answer group below, select the correct answers to be inserted into the blanks □ in the SQL statements.

"The customer numbers and the average order amounts of the customers each of whose average amount is greater than the order amount average"

Order_amount_table

Order_number	Date	Customer_number	Order_amount

Order amount: Order amount per order number

$\boxed{\text{c}}$ **Customer_number,** AVG **(Order_amount)** FROM **Order_amount _table**
GROUP BY **Customer_number**
$\boxed{\text{d}}$ AVG **(Order_amount) >** $\boxed{\text{e}}$

Answer group for c through e:

a) CREATIVE VIEW b) EXISTS c) HAVING

d) SELECT e) SUM (AVG (Order_amount))

f) SUM (Order _amount)) g) WHERE

h) (SELECT AVG (Order_amount) FROM Order_amount_table)

Q.4 Read the following description of a program, the explanation of the pseudo-language syntax, and the program itself, then answer Subquestions 1 and 2.

[Program description]

This program merges two product files into one product file. The program merges File A (Figure 1) with File B (Figure 2) to create File C (Figure 3). Assume that records which share the same product code do not exist in File A and File B.

	Product code	Product name
Record 1	100	Toothbrush
Record 2	200	Rinse

Figure 1: File A

	Product code	Product name
Record 1	150	Towel
Record 2	250	Cup
Record 3	350	Pen

Figure 2: File B

	Product code	Product name
Record 1	100	Toothbrush
Record 2	150	Towel
Record 3	200	Hair Rinse
Record 4	250	Cup
Record 5	350	Pen

Figure 3: File C

[Explanation of pseudo-language syntax]

Syntax	Explanation
⎡ Declaration part ⎣	This area is where procedures, variables, names, types, etc., are declared.
○	Declares names of procedures and variables, etc., and types, etc.
⎡ Process part ⎣	This area is where processes are described.
● Variable ← expression	Assigns the value of an expression to a variable.
● Procedure name (argument, ⋯)	Calls a procedure.
▲Conditional expression Yes Process 1 No Process 2 ▼	Executes Process 1 when the condition is true. Executes Process 2 when the condition is false.
■ Repetition conditional expression Process ■	Repeatedly executes the process while the condition is true.
⎡ ⎣	Indicates a block of processes.
=, <, >	Comparison operators that are used in conditional expressions and repetition conditional expressions.
{ }	Comments.

[Program]

```
Declaration part
○ Program name: File Merge
○ Integer: stsA, stsB, endf
○ File: fileA, fileB, fileC {sequential access files}
○ Record: rcdA, rcdB
○ Procedure: Recordinput (file, record, status)
  {This procedure reads one record from the file specified by "file", and stores the
  data in the area specified by "record". When a record is inputted, a "0" is stored in
  the variable specified by "status"; when no record is inputted, a "1" is stored in the
  variable specified by "status".}
○ Procedure: Recordoutput (file, record)
  {This procedure writes the contents of the area specified by "record" onto the file
  specified by "file".}

Processing part
● Recordinput (fileA, rcdA, stsA)
● Recordinput (fileB, rcdB, stsB)
● endf ← 0
■ endf = 0 {Repeat while endf = 0.}
      stsA = 1
α ⇒   Yes       stsB = 1
                Yes  ● endf ← 1
                No   ● Record output (fileC, rcdB)
                     ● Record input (fileB, rcdB, stsB)
      No        stsB = 1
                Yes  ● Recordoutput (fileC, rcdA)
                     ● Recordinput (fileA, rcdA, stsA)
β ⇒             No       rcdA product_code < rcdB product_code
                         Yes  ● Recordoutput (fileC, rcdA)
                              ● Recordinput (fileA, rcdA, stsA)
                         No   ● Recordoutput (fileC, rcdB)
                              ● Recordinput (fileB, rcdB, stsB)
■
```

Subquestion 1

How many times will the "**Yes**" portion of the program (indicated by "α ⇒ ") be executed? Select the correct answer from the answer group below.

Answer group:

a) 1　　b) 2　　c) 3　　d) 4　　e) 5

Subquestion 2.

Select from the answer group below the correct answers to insert in the blanks □ in the following description.

(1) Assume that the conditional expression indicated by "β ⇒" in the program was incorrectly written as follows:
rcdA product_code > **rcdB** product_code
The contents of Fiel C would then be a.

(2) Assume that, in the original program, the file shown in Figure 4 was inputted as File A instead of the file shown in Figure 1. The contents of File C would then be b.

Product code	Product name
200	Hair Rinse
100	Toothbrush

Figure 4: File A

Answer group for a and b:

a)

Product code	Product name
100	Toothbrush
150	Towel
200	Hair Rinse
250	Cup
350	Pen

b)

Product code	Product name
100	Toothbrush
150	Towel
200	Hair Rinse
350	Pen
250	Cup

c)

Product code	Product name
100	Toothbrush
200	Hair Rinse
150	Towel
350	Pen
250	Cup

d)

Product code	Product name
150	Towel
100	Toothbrush
200	Hair Rinse
350	Pen
250	Cup

e)

Product code	Product name
150	Towel
200	Hair Rinse
100	Toothbrush
250	Cup
350	Pen

f)

Product code	Product name
150	Towel
250	Cup
350	Pen
100	Toothbrush
200	Hair Rinse

g)

Product code	Product name
200	Hair Rinse
150	Towel
250	Cup
350	Pen
100	Toothbrush

h)

Product code	Product name
350	Pen
250	Cup
200	Hair Rinse
150	Towel
100	Toothbrush

Q.5 Read the following description of a program design, then answer Subquestions 1 through 3.

Company N has decided to develop a new online system for the management of product inventory. Mr. C is in charge of developing a module for handling inquiries on shipping.

The required functions, configuration and inputs/outputs of this system are as follows:

(1) The conceptual diagram showing the input-output relationship between each module and each file is illustrated in Figure 1.

(2) Both the inventory file and the stocking schedule file have one record per product code. The record format for these files is as shown in Figure 2.

(3) Interfaces between modules are as shown in Table 1.

(4) The inventory file is read. If the unallocated quantity of stock (quantity of stock - allocated quantity of stock) exceeds the desired quantity, the status is made "S1" and Messages A and B are made blank.

(5) If the unallocated quantity of stock in the inventory file is smaller than the desired quantity, the stocking schedule file is read to carry out the following processes:

① If there occurs, by the desired shipping date, a scheduled quantity of stock for storage (scheduled quantity of stock for storage - scheduled quantity of unallocated stock for storage) exceeding the stock shortage (desired quantity - unallocated quantity of stock), the status is made "S1," the quantity to be allocated from the stock scheduled for storage (that is, stock shortage in this case) is transcribed to Message A, and the scheduled storage date is transcribed to Message B.

② If the scheduled quantity of unallocated stock for storage is more than the stock shortage (desired quantity - unallocated quantity of stock) but the scheduled storage date is later than the desired shipping date, the status is made "S2," the quantity to be allocated from the stock scheduled for storage (that is, stock shortage in this case) is transcribed to Message A, and the scheduled storage date is transcribed to Message B.

③ If the scheduled quantity of unallocated stock for storage is less than the stock shortage (desired quantity - unallocated quantity of stock), the status is made "S3," the quantity that can be allocated from the stock scheduled for storage (or the scheduled quantity of unallocated stock for storage) is transcribed to Message A, and the scheduled storage date is transcribed to Message B.

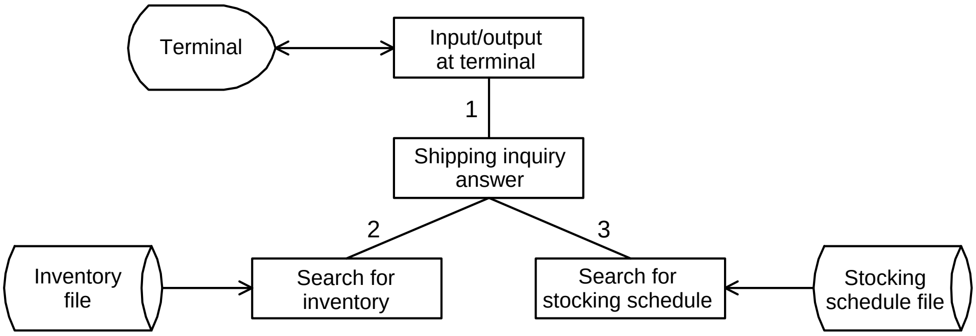


Figure 1: Module Structure and Conceptual Diagram of Input/Output Relationship

Product code	Quantity of stock	Allocated quantity of stock

Record format of inventory file

Product code	Scheduled storage date	Scheduled quantity of stock for storage	Scheduled quantity of allocated stock for storage

Record format of stocking schedule file

Figure 2: Record Format

Table 1: Interfaces between Modules

Connecting line number	Input from higher-level node	Output to higher-level module
1	Product code, desired quantity, desired shipping date	Status, Message A, and Message B
2	Product code	Unallocated quantity of stock
3	Product code	□ and scheduled quantity of unallocated stock for storage

Note: Connecting line number corresponds with a number in Figure 1.

Subquestion 1

Select from the following answer group the correct answer to insert in the blank □ in Table 1 above.

Answer group:

a) Desired quantity b) Desired shipping date c) Status

d) Scheduled storage date e) Message A f) Message B

Subquestion 2

Mr. C decided to prepare a decision table, as in Table 2 below, for the part in which the relations of processing and conditions of the shipping inquiry answer module are complicated. Select from the answer groups below the correct answers to insert in the blanks in Table 2.

Table 2: Decision Table Prepared by Mr. C

Stock shortage ≤ 0	Y	N	N	N
Scheduled quantity of unallocated stock for storage ≥ stock shortage	-	Y	b	N
Desired shipping date < scheduled storage date	-	N		-
Set status as "S1"	X	X		
Set status as "S2"				
Set status as "S3"				X
Transcribe blank to Messages A and B	X			
Transcribe stock shortage quantity to Message A		X		
Transcribe the scheduled quantity of unallocated stock for storage to Message A				X
a		X	X	X

Note: When an item in the above table is given a Y (yes), it means the condition in the corresponding condition description space is correct; N (no) means it is not correct; and - means the condition is neither correct nor incorrect.

When an item in the above table is given an X (eXecute), it means the process in the corresponding action space is to be executed when all conditions are met, while a blank means it is not to be executed.

Answer group for a:

a) Make status blank.

b) Transcribe the desired shipping date to Message A.

c) Transcribe the unallocated quantity of stock to Message A.

d) Transcribe the scheduled storage date to Message B.

e) Transcribe the unallocated quantity of stock to Message B.

Answer group for b:

a)	b)	c)	d)	e)	f)	g)	h)	i)	j)
N	N	N	N	N	N	N	N	N	N
-	-	N	N	N	N	Y	Y	Y	Y
N	N	Y	Y	N	N	-	-	Y	Y
	X		X				X		
X				X				X	
		X			X	X			X
			X	X		X			
							X	X	
	X	X							X
X	X	X	X	X	X	X	X	X	X

Subquestion 3

Mr. C prepared a flowchart for the shipping inquiry answer module, as shown in Figure 3, based on the descriptions of the required functions and Table 2. Select from the following answer groups the correct answers to insert in the blanks □ in Figure 3.

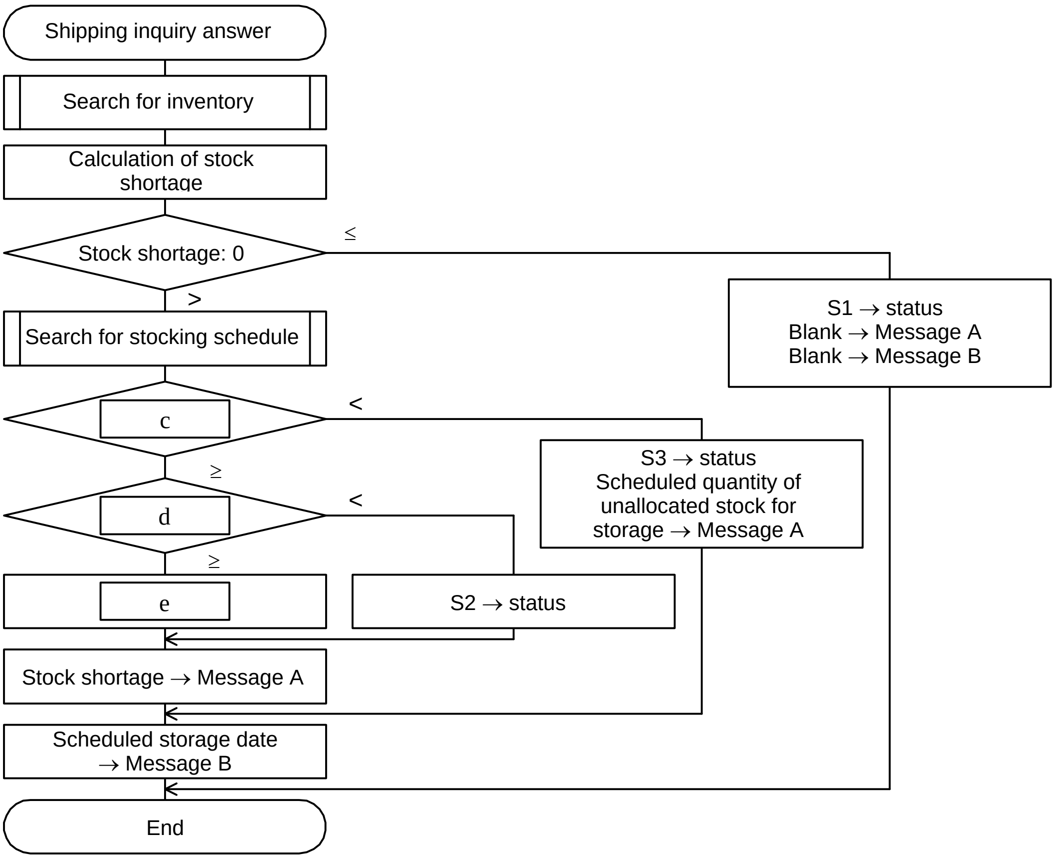

Shipping inquiry answer
Search for inventory
Calculation of stock shortage
Stock shortage: 0
≤
>
S1 → status
Blank → Message A
Blank → Message B
Search for stocking schedule
c
<
≥
S3 → status
Scheduled quantity of unallocated stock for storage → Message A
d
<
≥
e
S2 → status
Stock shortage → Message A
Scheduled storage date → Message B
End

Figure 3: Flowchart for Shipping Inquiry Answer Module

Answer group for c and d:

a) Desired quantity: stock shortage

b) Desired shipping date: scheduled storage date

c) Unallocated quantity of stock: desired quantity

d) Unallocated quantity of stock: stock shortage

e) Scheduled quantity of unallocated stock for storage: desired quantity

f) Scheduled quantity of unallocated stock for storage: stock shortage

Answer group for e:

a) "S1"→ status

b) "S2" → status

c) "S3" → status

d) Stock shortage → Message A

e) Scheduled storage date → Message B

f) Scheduled quantity of unallocated stock for storage → Message B

Select one of the following four questions (Q6, Q7, Q8, or Q9). Be sure to mark the Ⓢ in the Selection Column on your answer sheet for the question that you answered. If you select more than one question, only the first answer will be graded.

Q.6 Read the following description of a C program and the program itself, then answer the Subquestions 1 through 4.

[Program Description]

(1) The following function match let you check if a particular string matches a given pattern (p). The string consists of alphabetical characters and blanks, while pattern p consists of alphabetical characters, blanks and wildcards.

The possible wildcards are:

`?’ (question mark) can be matched by any character;

For example:

pattern “a?c” can be matched by “abc” or “aEc”...;

‘*’ (asterisk) can be matched by any number of characters;

For example:

pattern “a*c” can be matched by “ac”, “abc”, or “aBCDEc”...;

[(set of characters)] can be matched by any of characters in the specified set.

For example:

pattern “a[bcd]e” can be matched by “abe”, “ace” or “ade”.

(2) p is the pointer to a null-terminated pattern string. (A null-terminated string is a sequence of characters followed by a null character '\0').

(3) s is the pointer to a string to match by len characters. Assume that string contains no wildcard characters, such as ‘?’, ‘*’, ‘[’ or ‘]’.

(4) len is the length of string to match. For example, if s points to string “test case”, and len = 4, then the string to match is “test”.

[Program]

```
#define TRUE    1
#define FALSE   0

int match(char *p, char *s, int len)
{
    char c;
    int i, matched;

    c = *p;
    if(c == '\0')
        return (len == 0) ? TRUE : FALSE;
    if(c == '*') {
        for(i = 0; □ ; i++)
            if(match(p + 1, s + i, len - i))
                return TRUE;
        return FALSE;
    }
    if(len == 0)
        return FALSE;
    if(c == '[') {
        matched = FALSE;
        for(i = 1; p[i] != '\0'; i++) {
            if(p[i] == ']')
                return matched && match(p + i + 1, s + 1, len - 1);
            if(p[i] == *s)
                matched = TRUE;
        }
        return FALSE;
    }
    if(c == '?')
        return match(p + 1, s + 1, len - 1);
    return (c == *s) && match(p + 1, s + 1, len - 1);
}
```

Subquestion 1

Select the correct answer to insert in the blank □ from the choices provided.

Answer group:

a) `s[i] != '\0'` b) `i < len`

c) `i <= len` d) `(s[i] != '\0') && (i < len)`

Subquestion 2

Select the row with correct results from the following table.

Answer group:

	`match("", "", 0)`	`match("?", "", 0)`	`match("*", "", 0)`	`match("[]", "", 0)`
a)	FALSE	FALSE	TRUE	TRUE
b)	TRUE	FALSE	FALSE	TRUE
c)	FALSE	TRUE	TRUE	FALSE
d)	TRUE	FALSE	TRUE	FALSE

Subquestion 3

How many times will the function `match` be called when the statement `match("??*", "test case", 4);` is executed? Select the correct answer from the choices provided.

Answer group:

a) 4 b) 5 c) 6 d) 10

Subquestion 4

The following function `count_matches` counts all words that matches a given pattern `p` in a null-terminated string `s`. (A word is a sequence of one or more alphabetic characters. For example, phrase "this is a string" consists of following words: "this", "is", "a" and "string").

```
int count_matches(char *p, char *s)
{
    char c, *t;
    int in_word = FALSE, matches = 0;

    do {
        c = *s;
        if((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {
            if(!in_word) {
                in_word = TRUE;
                t = s;
            }
        }
        else {
            if(in_word) {
                in_word = FALSE;
                matches += match(p, t, □ );
            }
        }
        s++;
    } while(c);
    return matches;
}
```

Select the correct answer to insert in the blank □ from the choices provided.

Answer group:

a)	`t - s`	b)	`s - t`
c)	`strlen(s)`	d)	`strlen(t)`

Q.7 Read the following description of a COBOL program and the program itself, then answer Subquestions 1 and 2.

[Program Description]

A manufacturer selects suppliers of each of the materials required for production and manages the suppliers with a supplier-by-material file. Consider the process to delete from this file those suppliers with whom the company is no longer dealing.

(1) The format of records in the supplier-by-material file is shown below:

Material code 6 digits	Supplier count 1 digit	Supplier code		
		4 digits	…	4 digits

① The file is an indexed file that uses the material code as a key.

② The supplier count stores the number of suppliers registered in the file.

③ Up to five supplier codes (or no codes) are registered for each material code.

④ Supplier codes are registered starting from the first field; a space is stored in any field in which no supplier code is registered.

(2) The format of records in the deletion instruction file is shown below:

Material code	Supplier code
6 digits	4 digits

① The program reads from the supplier-by-material file the record which corresponds to the material code, deletes the corresponding supplier code, and reduces the supplier count by one.

② As part of the deletion processing, the program alters the supplier-by-material file so that item ④under (1) above may be satisfied.

③ Assume that all of the data in the deletion instruction file is correct.

[Program]

(Line No.)

```
 1 DATA DIVISION.
 2 FILE SECTION.
 3 FD    HATCHUSAKI.
 4 01    HATCHU-R.
 5          03    H-ZAIRYO                PIC X(6).
 6          03    H-KENSU                 PIC 9.
 7          03    H-TABLE.
 8                   05    H-HATCHU          OCCURS 5 PIC X(4).
 9 FD    SHIJI.
10 01    SHIJI-R.
11          03    S-ZAIRYO                PIC X(6).
12          03    S-HATCHU                PIC X(4).
13 WORKING-STORAGE SECTION.
14 01    END-SW                           PIC X VALUE SPACE.
15 01    CNT                              PIC 9.
16 01    W-CNT                            PIC 9.
17 PROCEDURE DIVISION.
18 HAJIME.
19          OPEN INPUT SHIJI I-O HATCHUSAKI.
20          PERFORM UNTIL END-SW = "Z"
21               READ SHIJI AT END MOVE "Z" TO END-SW END-READ
22               IF END-SW = SPACE
23                    MOVE S-ZAIRYO TO H-ZAIRYO
24                    READ HATCHUSAKI
25                         INVALID DISPLAY "READ-ERROR S-ZAIRYO = ", S-ZAIRYO
26                         NOT INVALID PERFORM KENSAKU
27                                             PERFORM D-RTN
28                    END-READ
29               END-IF
30          END-PERFORM.
31          CLOSE SHIJI HATCHUSAKI.
32          STOP RUN.
33 KENSAKU.
34          PERFORM VARYING CNT FROM 1 BY 1 UNTIL □
35               CONTINUE
36          END-PERFORM.
37 D-RTN.
38          PERFORM UNTIL CNT = H-KENSU
39               COMPUTE W-CNT = CNT + 1
40               MOVE H-HATCHU(W-CNT) TO H-HATCHU(CNT)
41               COMPUTE CNT = CNT + 1
42          END-PERFORM.
43          MOVE SPACE TO H-HATCHU(H-KENSU).
44          COMPUTE H-KENSU = H-KENSU - 1.
45          REWRITE HATCHU-R
46               INVALID DISPLAY "REWRITE-ERROR", H-ZAIRYO, S-HATCHU.
```

Subquestion 1.

From the answer group below, select the correct answer to insert in the blank □ in the program above.

Answer group:

a) `S-HATCHU = H-HATCHU(CNT)`

b) `S-HATCHU = H-HATCHU(H-KENSU)`

c) `S-ZAIRYO = H-ZAIRYO`

Subquestion 2.

Lines 38 to 42 in the program are being changed to avoid repetition. From the answer group below, select the correct answers to insert in the blanks □ in the modified code below.

Note that `W1`, `W2`, and `W3` are all defined as two-digit work areas, and `W-TABLE` is defined as a 20 (alphanumeric)-character item.

```
IF CNT NOT = H-KENSU
   MOVE H-TABLE TO W-TABLE
   COMPUTE W1 = [        a        ]
   COMPUTE W2 = [        b        ]
   COMPUTE W3 = [        c        ]
   MOVE W-TABLE(W1:W3) TO H-TABLE(W2:W3)
END-IF.
```

Answer group for a through c:

a) `CNT * 4 – 1`

b) `CNT * 4 + 1`

c) `(CNT – 1) * 4 – 1`

d) `(CNT – 1) * 4 + 1`

e) `(CNT + 1) * 4 – 1`

f) `(CNT + 1) * 4 + 1`

g) `(H-KENSU - 1) * 4 + 1`

h) `(H-KENSU - CNT) * 4 - 1`

i) `(H-KENSU - CNT) * 4`

j) `(H-KENSU - CNT) * 4 + 1`

Q.8 Read the following description of a Java program and the program itself, then answer the Subquestion.

[Program description]

The following are classes and test classes for creating a digital logic circuit simulator.

(1) The class NotGate, which represents NOT gates, and the class AndGate, which represents AND gates, are defined as subclasses of the abstract class Gate, which represents gates.

(2) All subclasses of Gate inherit the methods connectOutputTo, getInput, and getOutput; and newly include the method tick.

(3) The method connectOutputTo connects a gate's output signal line to the input signal line of a specified gate.

(4) The methods getInput and getOutput return the gate input signal line and output signal line, respectively.

(5) The method tick executes computations unique to each gate, and outputs either a true or false value to the output signal line.

(6) The class wire expresses an input signal line or output signal line.

(7) There is one output signal line for each gate.

(8) The diagram below presents an example of a circuit created using the test class LogicCircuitTest and its operations. The method main displays a computation result value in the standard output.

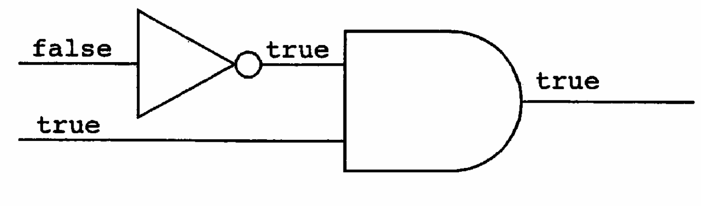


Figure: Example of Circuit Created Using LogicCircuitTest and Its Operations

[Program]

```
class Wire {
    private boolean value;
    public boolean getValue() { return value; }
    public void setValue(boolean value) { this.value = value; }
}

abstract class Gate {
    protected Wire[] input;
    protected Wire output;
    public Gate(int nInputs) {
        input = [      a      ];
        for (int i = 0; i < nInputs; i++) { input[i] = new Wire(); }
        output = new Wire();
    }
    public void connectOutputTo(Gate otherGate, int nthInput) {
        try {
             [      b      ];
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public Wire getInput(int n) { return input[n]; }
    public Wire getOutput() { return output; }
    abstract public void tick();
}

class NotGate extends Gate {
    public NotGate() { super(1); }
    public void tick() { output.setValue( [      c      ]); }
}

class AndGate extends Gate {
    public AndGate() { super(2); }
    public void tick() { output.setValue([      d      ]); }
}

public class LogicCircuitTest {
    public static void main(String[] args) {
        Gate not = new NotGate();
        Gate and = new AndGate();
        not.connectOutputTo(and, 0);
        not.getInput(0).setValue(false);
        and.getInput(1).setValue(true);
        not.tick();
        and.tick();
        System.out.println(and.getOutput().getValue());
    }
}
```

Subquestion

From the answer groups below, select the correct answers to be inserted in the blanks □ in the above program.

Answer group for a:

a) `new Wire(nInputs)`

b) `new Wire[nInputs]`

c) `new Wire()`

d) `new Wire[] {new Wire()}`

Answer group for b:

a) `input[nthInput].setValue(otherGate.output.getValue())`

b) `input[nthInput] = otherGate.output`

c) `otherGate.input[nthInput].setValue(output.getValue())`

d) `otherGate.input[nthInput] = output`

Answer group for c and d:

a) `input[0].getValue() || input[1].getValue()`

b) `!input[0].getValue() || !input[1].getValue()`

c) `input[0].getValue() && input[1].getValue()`

d) `!input[0].getValue() && !input[1].getValue()`

e) `input[0].getValue() != input[1].getValue()`

f) `input[0].getValue()`

g) `!input[0].getValue()`

h) `1 - input[0].getValue()`

Q.9 Read the following description of an assembler program and the program itself, then answer Subquestions 1 and 2.

[Program Description]

"n" consecutive words are regarded as a bit string consisting of 16 × n bits. The subprogram `SHIFT` shifts this bit string right by m positions.

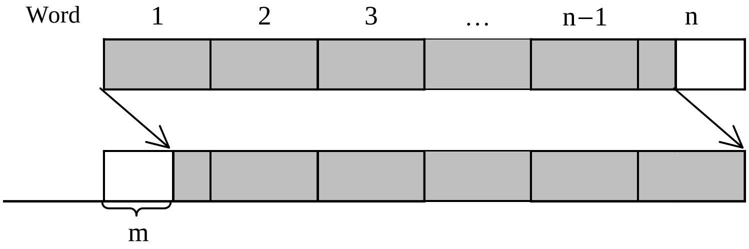


Figure 1: Processes by the Subprogram SHIFT

(1) The main program stores the following values in `GR1` through `GR3`, respectively, and then passes these values to `SHIFT`.

`GR1` : Address of the first word of the bit string

`GR2` : `m`

`GR3` : `n`

(2) Assume that $1 \leqq n \leqq 32767$ and that $0 \leqq m \leqq 16$.

(3) The shifted bit string is stored in the original area.

(4) "0" is stored in the empty bit positions before the left end of the string. The bits removed from the right side of the register are discarded.

(5) The functions of the `RPUSH` and `RPOP` macro instructions used in the program are as described below:

`RPUSH`: Saves the contents of `GR1` through `GR7` into the stack.

`RPOP`: Restores the contents of the stack into `GR1` through `GR7`.

[Program]

```
(Line No.)
   1  SHIFT START                         ;
   2        RPUSH                         ; Save registers
   3        [        a        ]           ;
   4        SUBA   GR4,GR2                ; (16·m)→GR4
   5        LAD    GR5,0                  ; ‘0’ bit string to be stored at the left end
   6  LOOP  SLL    GR5,0,GR4              ; Move to the left end the remaining bit string
                                            from the previous word
   7        LD     GR0,0,GR1              ; The bit string in the current word→GR0
   8        [        b        ]           ;
   9        OR     GR0,GR5                ; Catenate the left-over bit string
  10        LD     GR5,0,GR1              ; Save bit string in current word into GR5
  11        ST     GR0,0,GR1              ; Store catenated bit strings into the current
                                            word
  12        LAD    GR1,1,GR1              ; Obtain the address of the next word
  13        SUBA   GR3,=1                 ;
  14        [        c        ]           ;
  15        RPOP                          ; Restore registers
  16        RET                           ;
  17        END                           ;
```

Subquestion 1

From the answer groups below, select the correct answers to insert in the blanks [   ] in the program above.

Answer group for a:

a)	`AND`	`GR4,=16`	b)	`LAD`	`GR4,16`
c)	`LD`	`GR4,16`	d)	`OR`	`GR4,=16`

Answer group for b:

a)	`SLL`	`GR0,0,GR2`	b)	`SLL`	`GR0,0,GR4`
c)	`SRL`	`GR0,0,GR2`	d)	`SRL`	`GR0,0,GR4`

Answer group for c:

a)	`JMI`	`LOOP`	b)	`JOV`	`LOOP`
c)	`JPL`	`LOOP`	d)	`JZE`	`LOOP`

Subquestion 2

The LAD instruction in line 5 has been replaced with the following three instructions in order to store the bit substring removed from the right end in the open bit positions at the left end of the string, as shown in Figure 2 below. From the following answer group, select the correct answer to insert in the blank □ below.

```
LD      GR5,GR1
ADDL    GR5,GR3
□
```

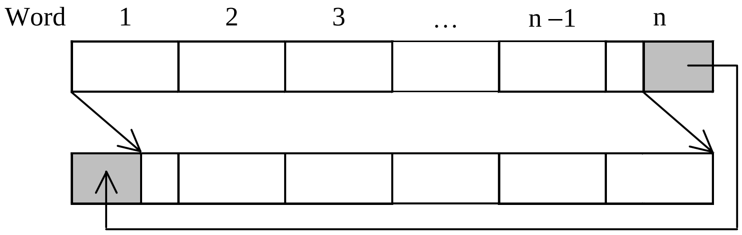


Figure 2: The removed bit substring is stored on the left side.

Answer group:

a) `LD GR5,–1,GR5` b) `LD GR5,0,GR5`

c) `LD GR5,1,GR5` d) `ST GR5,–1,GR1`

e) `ST GR5,0,GR1` f) `ST GR5,1,GR1`

Select one of the following four questions (Q10, Q11, Q12, or Q13). Be sure to mark the Ⓢ in the Selection Column on your answer sheet for the question that you answered.
If you select more than one question, only the first answer will be graded.

Q.10 Read the following description of a C program and the program itself, then answer Subquestions 1 and 2.

[Program Description]

A function, DrawLine ( ), was prepared, which draws a straight line between a coordinate (x1, y1) and another (x2, y2) on a graphic display screen. Another function, SetPixel( ), used to draw points at a coordinate (x, y), is already available.

[Program]

```
void  SetPixel( int x ,int y );

void  DrawLine( int x1 ,int y1 ,int x2 ,int y2  )
{
    int    Thresh = 0 ,Index ;
    int    Xunit = 1 ;          /* increment in X direction        */
    int    Yunit = 1 ;          /* increment in Y direction        */
    int    Xdiff = x2 - x1 ;    /* displacement in X direction     */
    int    Ydiff = y2 - y1 ;    /* displacement in Y direction     */

    if ( Xdiff < 0 ) {
        Xdiff = -Xdiff ; Xunit = -1 ;
    }
    if ( Ydiff < 0 ) {
        Ydiff = -Ydiff ; Yunit = -1 ;
    }

    if ( Xdiff > Ydiff ) {         /* determines the direction of loop     */
        for ( Index = 0 ; Index < Xdiff + 1 ; Index++ ) { /* loop in X direction */
            SetPixel( x1 ,y1 );
            x1 += Xunit;
            Thresh  += Ydiff;
            if ( Thresh >= Xdiff )    {
                Thresh -= Xdiff;
                y1 += Yunit;

            }
        }
    } else {
        for ( Index = 0 ; Index < Ydiff + 1 ; Index++ ) { /* loop in y direction */
            SetPixel( x1 ,y1 );
            y1 += Yunit;
            Thresh  += Xdiff;
            if ( Thresh >= Ydiff ) {
                Thresh -= Ydiff;
                x1 += Xunit;
            }
        }
    }
}
```

Subquestion 1

Select from the answer groups below the correct answers to insert in the blanks $\boxed{\quad\quad}$ in the following description explaining the performance of this program.

(1) The program calculates for the given two points the displacement in the X direction and that in the Y direction.

(2) The program determines $\boxed{\text{ a }}$ of the increment in the X direction and that in the Y direction from the result obtained in (1) above.

(3) The program compares $\boxed{\text{ b }}$ between the displacement in the X direction and that in the Y direction to determine the direction of the loop.

(4) Assuming the larger displacement as DH and the smaller displacement as DL based on the result of the comparison in (3) above, the number of points to draw is $\boxed{\text{ c }}$.

Answer group for a and b:

a) ratio
b) equation
c) tangent value
d) absolute value
e) cosine value
f) gradient
g) sign
h) difference
i) segment

Answer group for c:

a) | DH |
b) | DL |
c) | DH + 1 |
d) | DL + 1 |
e) | DH | + 1
f) | DL | + 1

Subquestion 2

If all the straight lines are drawn by loops in the X direction, what problem will occur when the displacement in the Y direction is large? Select the correct answer from the following answer group.

Answer group:

a) They can turn out to be straight lines whose gradient is different from what was intended.
b) There may be a risk of the denominator of the division being zero.
c) They may become vertical lines.
d) They may become horizontal lines.
e) They may become broken lines (that is, points being not adjacent).

Q.11 Read the following description of a COBOL program and the program itself, then answer the Subquestion.

[Program Description]

There is a customer file in which customers' names and addresses are registered. This program calculates the number of customers by area and prints those numbers out in descending order.

(1) The record format of the customer file is as follows:

Name	Address	Other information
X(20)	X(50)	X(30)

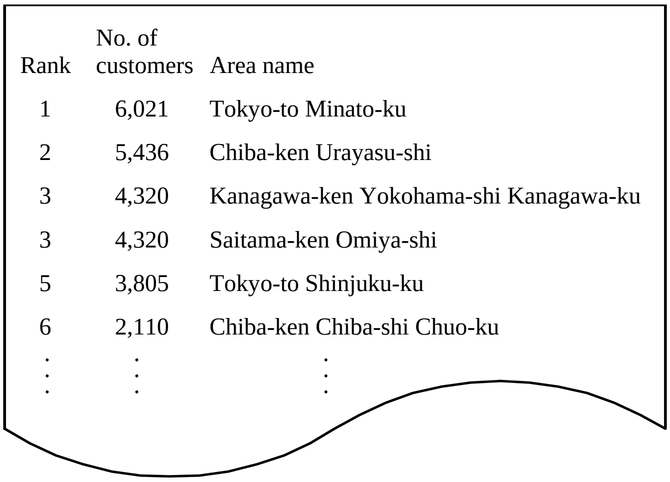
(2) Each address in the customer file has one "@" character to separate area names from addresses. An example is shown below:

Tokyo-to Minato-ku @ Toranomon 1-16-4 Urban Toranomon Bldg. 8F

(3) The customer file has one or more records.

(4) Rank numbers are assigned to areas, starting with 1 starting as the area with the largest number of customers. When two or more areas have the same number of customers, they will be assigned the same rank number. Areas sharing the same rank number are arranged by area name in alphabetical order.

(5) The print-out format is as follows:



```
       No. of
Rank   customers  Area name
  1      6,021    Tokyo-to Minato-ku
  2      5,436    Chiba-ken Urayasu-shi
  3      4,320    Kanagawa-ken Yokohama-shi Kanagawa-ku
  3      4,320    Saitama-ken Omiya-shi
  5      3,805    Tokyo-to Shinjuku-ku
  6      2,110    Chiba-ken Chiba-shi Chuo-ku
  :        :             :
```

(6) The headline should be printed only once.

(7) The number of areas and the number of customers per area should be within 5 digits.

[Program]

```
DATA DIVISION.
FILE SECTION.
FD   KOKYAKU-F.
01   KOKYAKU-R.
     02                        PIC   X(20).
     02   KOKYAKU-ADR        PIC   X(50).
     02                        PIC   X(30).
FD   CHIIKI-WORK-F.
01   CHIIKI-WORK-R          PIC   X(50).
FD   INJI-F.
01   INJI-R                  PIC   X(66).
SD   CHIIKI-F.
01   CHIIKI-R                PIC   X(50).
SD   KYAKUSU-F.
01   KYAKUSU-R.
     02   KYAKUSU-NUM        PIC   9(05).
     02   KYAKUSU-ADR        PIC   X(50).
WORKING-STORAGE SECTION.
01   FLG                      PIC   X(01) VALUE SPACE.
01   WORK-NUM                PIC   9(05).
01   JUNI                     PIC   9(05).
01   JUNI-S                  PIC   9(05) VALUE 1.
01   INJI1                    PIC   X(21) VALUE " Rank No. of customers Area name ".
01   INJI2.
     02   INJI2-NO           PIC   ZZ,ZZ9.
     02   INJI2-NUM          PIC   ZZZZ,ZZ9.
     02                        PIC   X(02) VALUE SPACE.
     02   INJI2-ADR          PIC   X(50).
PROCEDURE DIVISION.
START-RTN.
     SORT CHIIKI-F
         ON DESCENDING KEY CHIIKI-R
             INPUT  PROCEDURE IS HENKAN-RTN
             [                  a                  ].
     SORT KYAKUSU-F
         [                  b                  ]
             INPUT  PROCEDURE IS TASU-RTN
             OUTPUT PROCEDURE IS INJI-RTN.
     STOP RUN.
```

```
HENKAN-RTN.
    OPEN INPUT KOKYAKU-F.
    PERFORM UNTIL FLG = "E"
       READ KOKYAKU-F AT END
          MOVE "E" TO FLG
       NOT AT END
          UNSTRING KOKYAKU-ADR DELIMITED BY "@"
            INTO  CHIIKI-R
          END-UNSTRING
          RELEASE CHIIKI-R
       END-READ
    END-PERFORM.
    CLOSE KOKYAKU-F.
TASU-RTN.
    OPEN INPUT CHIIKI-WORK-F.
    INITIALIZE FLG KYAKUSU-NUM.
    READ CHIIKI-WORK-F.
    MOVE CHIIKI-WORK-R TO KYAKUSU-ADR.
    PERFORM UNTIL FLG = "E"
       COMPUTE KYAKUSU-NUM = KYAKUSU-NUM + 1
       READ CHIIKI-WORK-F AT END
          RELEASE KYAKUSU-R
          MOVE "E" TO FLG
       NOT AT END
          IF CHIIKI-WORK-R NOT = KYAKUSU-ADR THEN
             [              c              ]
             MOVE ZERO TO KYAKUSU-NUM
             MOVE CHIIKI-WORK-R TO KYAKUSU-ADR
          END-IF
       END-READ
    END-PERFORM.
    CLOSE CHIIKI-WORK-F.
INJI-RTN.
    OPEN OUTPUT INJI-F.
    WRITE INJI-R FROM INJI1 AFTER PAGE.
    INITIALIZE FLG.
    RETURN KYAKUSU-F.
    MOVE KYAKUSU-NUM TO WORK-NUM.
    [              d              ]
       MOVE KYAKUSU-NUM TO INJI2-NUM
       MOVE KYAKUSU-ADR TO INJI2-ADR
       [              e              ]
          MOVE JUNI-S TO INJI2-NO
       ELSE
          MOVE KYAKUSU-NUM TO WORK-NUM
          MOVE JUNI TO JUNI-S INJI2-NO
       END-IF
       WRITE INJI-R FROM INJI2 AFTER 2
       RETURN KYAKUSU-F AT END
          MOVE "E" TO FLG
       END-RETURN
    END-PERFORM.
    CLOSE INJI-F.
```

Subquestion

Select from the following answer groups the correct answers to insert in the blanks □ in the above program.

Answer group for a:

a) `GIVING CHIIKI-F`

b) `GIVING CHIIKI-WORK-F`

c) `OUTPUT PROCEDURE IS INJI-RTN`

d) `OUTPUT PROCEDURE IS TASU-RTN`

e) `USING CHIIKI-F`

f) `USING CHIIKI-WORK-F`

Answer group for b:

a) `ON ASCENDING KEY KYAKUSU-ADR`

b) `ON ASCENDING KEY KYAKUSU-NUM`

c) `ON ASCENDING KEY KYAKUSU-NUM DESCENDING KEY KYAKUSU-ADR`

d) `ON DESCENDING KEY KYAKUSU-ADR`

e) `ON DESCENDING KEY KYAKUSU-NUM`

f) `ON DESCENDING KEY KYAKUSU-NUM ASCENDING KEY KYAKUSU-ADR`

Answer group for c:

a) `COMPUTE KYAKUSU-NUM = KYAKUSU-NUM + 1`

b) `IF KYAKUSU-NUM > 1 THEN`

c) `MOVE "E" TO FLG`

d) `RELEASE KYAKUSU-R`

e) `RETURN KYAKUSU-F`

f) `WRITE KYAKUSU-R`

Answer group for d:

a) `PERFORM UNTIL FLG = "E"`

b) `PERFORM UNTIL JUNI = JUNI-S`

c) `PERFORM VARYING JUNI-S FROM 1 BY 1 UNTIL FLG = "E"`

d) `PERFORM VARYING JUNI-S FROM 1 BY 1 UNTIL JUNI-S > 1`

e) `PERFORM VARYING JUNI FROM 1 BY 1 UNTIL FLG = "E"`

f) `PERFORM VARYING JUNI FROM 1 BY 1 UNTIL JUNI > 1`

Answer group for e:

a) `IF KYAKUSU-NUM = WORK-NUM THEN`

b) `IF KYAKUSU-NUM > WORK-NUM THEN`

c) `IF KYAKUSU-NUM < WORK-NUM THEN`

d) `IF JUNI = JUNI-S THEN`

e) `IF JUNI > JUNI-S THEN`

f) `IF JUNI < JUNI-S THEN`

Q.12 Read the following description of a Java program and the program itself, then answer the Subquestion.

[Program description]

Program 1 is a general-purpose class `CSVParser`, which serves to analyze comma-separated (CSV format) data. Program 2 is a program that extends the class `CSVParser` and converts the input data file (which is in comma-separated format) to output data which has a tagged format.

The class `CSVParser` analyzes the comma-separated data file specified in the constructor using the method `parse`, and calls individual methods according to the states shown below:

State	Called method	Argument
File reading started	`startDocument`	None
New record reached	`startRecord`	`n:` Record number
One field in record read	`value`	`chars:` Field value `n:` Field number
Record end reached	`endRecord`	
File processing completed	`endDocument`	`None`

Figures 1 and 2 show the Program 2 input data file test.csv and output results, respectively.

```
Ichiro Tanaka,tanaka@sales.example.com,111-1111
Jiro Yamada,yamada@eng.example.com,222-2222
Saburo Suzuki,suzuki@mkt.example.com,333-3333
```

Figure 1: Input Data File Test.csv

```
<addressbook>
  <person id="1">
    <name>Ichiro Tanaka</name>
    <email>tanaka@sales.example.com</email>
    <phone>111-1111</phone>
  </person>
  <person id="2">
    <name>Jiro Yamada</name>
    <email>yamada@eng.example.com</email>
    <phone>222-2222</phone>
  </person>
  <person id="3">
    <name>Saburo Suzuki</name>
    <email>suzuki@mkt.example.com</email>
    <phone>333-3333</phone>
  </person>
</addressbook>
```

Figure 2: Output Result

The input data format is as follows:

(1) A single record consists of three fields, which are delimited by commas (“,”). There is a newline character at the end of a record.

(2) There are name, e-mail address, and telephone number fields. A single record contains one of each of these fields, arranged in the order listed here.

① Fields never contain commas.

② Fields cannot be omitted, and there are no fields consisting entirely of blank characters.

The output data format is as follows.

(1) A unit of data consisting of a value enclosed by a start tag and an end tag is called an element.

① The start tag is entered as **<tag name>,** and the end tag is entered as **</tag name>**.

② The start tag may contain an attribute, in the format **<tag name attribute name = “attribute value”>**.

③ In the explanation below, this element is referred to as “Element **tag name**”.

(2) Elements can be nested. More specifically, one element may be inserted into another.

(3) All start tags and end tags must correspond to each other. More specifically, an element starting with a given start tag must be closed by an end tag with the same tag name.

The procedure for converting the input data format to the output data format is as follows.

(1) Assemble the entire input data and create the element `addressbook`.

(2) The element `person` is created separately for each record in the input data. The recording reading sequence number is added as an attribute id value to the element `person`.

(3) The elements `name, email`, and `phone` are created from the respective fields in a single record, in the sequence in which the fields appear in the record.

The class `java.io.FileReader` is used to read the contents of a text file. The operations are as follows:

(1) When a character string representing a file name is given to the constructor, the input stream from that file is opened.

(2) The method `read` reads one character from the input stream and returns it as an int type.

(3) When the end of the input stream is reached, the method `read` is returned, “−1”.

[Program 1]

```
import java.io.FileReader;
import java.io.IOException;
import java.io.FileNotFoundException;

public class CSVParser {
   private FileReader reader;

   public CSVParser(String fileName) throws FileNotFoundException {
      // Generates a text input file reader.
      reader = new FileReader(fileName);
   }

   public void startDocument() {}
   public void startRecord(int n) {}
   [              a              ]
   public void endRecord(int n) {}
   public void endDocument() {}

   public void parse() throws IOException {
      int c;
      StringBuffer buf = new StringBuffer();
      boolean [              b              ];
      int fieldNumber = 0;
      int recordNumber = 0;

      startDocument();
      while ((c = reader.read()) != -1) { // Read a character from reader
// The method read returns "-1" if there are no usable characters in the
input stream.
         char ch = (char)c;
         switch (ch) {
           case ',':
              [              c              ];
              buf.delete(0, buf.length());
              break;
           case '\n':
              if (!endOfRecord) {
                 endOfRecord = true;
                 [              c              ];
                 buf.delete(0, buf.length());
                 fieldNumber = 0;
                 [              d              ];
              }
              break;
           default:
              if (endOfRecord) {
                 [              e              ];
              }
              endOfRecord = false;
              [              f              ];
         }
      }
      endDocument();
   }
}
```

[Program 2]

```
import java.io.FileNotFoundException;

public class TaggedDataGenerator extends CSVParser {
   public TaggedDataGenerator(String fileName)
      throws FileNotFoundException {
      super(fileName);
   }
   public void startDocument() {
      System.out.println("<addressbook>");
   }
   public void startRecord(int n) {
      System.out.println("\t<person id=\""+n+"\">");
   }
   public void value(String chars, int n) {
      String tag = (n == 1) ? "name" : (n == 2) ? "email" : "phone";
      System.out.println("\t\t<"+tag+">"+chars+"</"+tag+">");
   }
   public void endRecord(int n) {
      System.out.println("\t</person>");
   }
   public void endDocument() {
      System.out.println("</addressbook>");
   }
   public static void main(String [] args) {
      TaggedDataGenerator parser = null;
      try {
         parser = new TaggedDataGenerator("test.csv");
         parser.parse();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

Subquestion

From the answer groups below, select the correct answers to be inserted in the blanks □ in Program 1 above.

Answer group for a:

a) public void value(StringBuffer chars, int n) {}

b) public void value(String chars, int n) {}

c) abstract public void value(String chars, int n) {}

d) public void value(String chars, int n);

Answer group for b:

a) endOfRecord = null

b) endOfRecord = false

c) endOfRecord = true

d) endOfRecord

Answer group for c:

a) value(buf, fieldNumber)

b) value(buf.toString(), fieldNumber)

c) value(buf, ++fieldNumber)

d) (buf.toString(), ++fieldNumber)

Answer group for d and e:

a) startDocument()

b) startDocument(++recordNumber)

c) startRecord(recordNumber)

d) startRecord(++recordNumber)

e) endDocument()

f) endDocument(recordNumber)

g) endRecord()

h) endRecord(recordNumber)

Answer group for f:

a) buf += ch

b) buf[buf.length] = ch

c) buf.append(ch)

d) buf.append(ch.toString())

Q.13 Read the following description of an assembler program and the program itself, then answer the Subquestion.

[Program Description]

This is a subroutine to divide an input character string into words and store the head address and length of each word into the table.

(1) When called by the main program, subroutine WRDSCN inputs a character string of one record from the input device to the input area.

(2) It resolves the input character strings into words using blank characters as delimiters.

(3) The subroutine then stores the head address and the length of resolved words in an area TABLE in the subroutine as shown in the following figure:

TABLE + 0	Head address of the first word
+ 1	Length of the first word
+ 2	Head address of the second word
+ 3	Length of the second word
	⋮
+ n - 1	Head address of the last word
+ n	Length of the last word

Figure : Contents of TABLE

(4) After storing the data in TABLE, the subroutine stores the head address of TABLE in GR1 and returns control to the main program. When an empty record is entered or EOF is detected, the subroutine stores -1 in GR1 and returns control to the main program.

[Program]

```
WRDSCN   START                       ;
         IN      INBUF,INLNG         ;
         LD      GR3,INLNG           ;
         LEA     GR3,-1,GR3          ;
         JMI     ERROR               ;
         LEA     GR1,-1              ;
         LEA     GR2,0               ;
         LEA     GR3,0               ; Initialize flag to check if it is in the word or
                                       not
         ST      GR3,WRDLNG          ; Initialize length of word
LOOP     LEA     GR1,1,GR1           ;
         CPL     GR1,INLNG           ; Does the character string end?
         JZE     BREAK               ;
         LD      GR0,INBUF,GR1       ;
         CPL     GR0,SPACE           ; Blank character?
         JZE     NONWRD              ;
         LEA     GR3,0,GR3           ; In the word?
         JNZ     LNGUP               ;
         LEA     GR0,INBUF,GR1       ;
         ST      GR0,TABLE,GR2       ; Put head address of word into TABLE
         LEA     GR2,1,GR2           ;
         LEA     GR3,1               ; Put flag into the word
LNGUP    LEA     GR0,1               ; Add 1 to word length
         ADD     GR0,WRDLNG          ;
         ST      GR0,WRDLNG          ;
         JMP     LOOP                ;
NONWRD   LEA     GR3,0,GR3           ; In the word?
         JZE     LOOP                ;
         LD      GR0,WRDLNG          ;
         ST      GR0,TABLE,GR2       ; Put word length into TABLE
         LEA     GR2,1,GR2           ;
         LEA     GR3,0               ; Initialize flag
         ST      GR3,WRDLNG          ; Initialize word length
         JMP     LOOP                ;
BREAK    LEA     GR3,0,GR3           ; In the word?
         JZE     RETURN              ;
         LD      GR0,WRDLNG          ;
         ST      GR0,TABLE,GR2       ;
RETURN   LEA     GR1,TABLE           ;
         RET                         ;
ERROR    LEA     GR1,-1              ;
         RET                         ;
INBUF    DS      80                  ;
INLNG    DS      1                   ;
TABLE    DS      80                  ;
WRDLNG   DS      1                   ;
SPACE    DC      ' '                 ;
         END                         ;
```

Subquestion

Select from the answer group below the correct answers to insert in the blanks $\boxed{}$ in the following description about the program.

(1) When the following character string is inputted, the 22nd line is executed $\boxed{\text{a}}$ times. After execution, the content of GR2 at the exit to the main program is $\boxed{\text{b}}$ and that of GR3 is $\boxed{\text{c}}$.

Address no.

	100		102		104		106		108		110		112		114
INBUF	Δ	I	Δ	h	a	v	e	Δ	Δ	a	Δ	b	o	o	k

Note: Δ is a blank character.

INLNG	15

(2) To specify the end of the table, you want to modify the program in such a way that it stores -1 in the table entry immediately following the entry into which the length of the last word is stored. To do this, you just insert the two lines of ① below in front of the 39th line and the line of ② in front of the $\boxed{\text{d}}$ line.

①
```
LEA  GR0,-1
ST   GR0,TABLE,GR2
```

②
```
LEA  GR2,1,GR2
```

Answer group for a, b, and c:

a) 0　b) 1　c) 4　d) 6　e) 7

f) 8　g) 10　h) 11　i) 14　j) 15

Answer group for d:

a) 34　b) 35　c) 36　d) 37　e) 38