April 2014

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

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

Instructions:

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

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

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

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

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

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

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

   Answer group
   a) March b) April c) May d) June

   Since the correct answer is "b) April", 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.

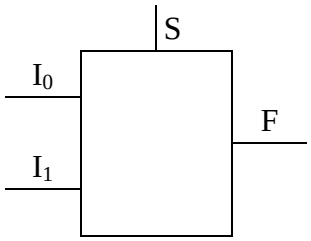
Q1. Read the following description concerning electronic circuits, and then answer Subquestion.

The multiplexer circuits and demultiplexer circuits are widely used in various applications in digital systems such as data selection, data routing, operation sequencing, and logic-function generation.

[Multiplexer]

A multiplexer circuit is a circuit which selects one input line among multiple input lines and outputs the binary value of the selected input line to an output line.

Figure 1 shows a schema of 2-to-1 multiplexer. "2-to-1" means 2 inputs and 1 output.



I_0, I_1: Input lines
S: Selection control line
F: Output line

Figure 1 Schema of 2-to-1 multiplexer

A binary value, 0 or 1, is associated with each of I_0, I_1, S and F.

The 2-to-1 multiplexer shown in Figure 1 works as follows:

- If S is 0, then the binary value of I_0 is output to F.
- If S is 1, then the binary value of I_1 is output to F.

This specification can be expressed by the truth table shown in Table 1.

Table 1 Truth table of 2-to-1 multiplexer

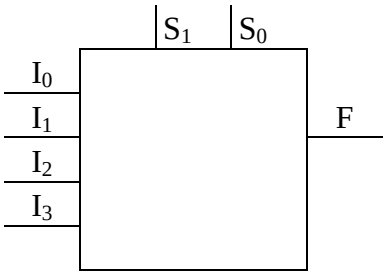
Given binary values			Output
S	I_0	I_1	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	
1	0	0	
1	0	1	A
1	1	0	
1	1	1	1

This specification can also be expressed by the Boolean equation as follows:

$$F = (!S \,\&\, I_0) \mid (S \,\&\, I_1)$$

Here, the operators "&", "|" and "!" denote Boolean operations AND, OR and NOT respectively.

Figure 2 shows a schema of 4-to-1 multiplexer.



I_0, I_1, I_2, I_3: Input lines
S_0, S_1: Selection control lines
F: Output line

Figure 2 Schema of 4-to-1 multiplexer

A binary value, 0 or 1, is associated with each of I_0, I_1, I_2, I_3, S_0, S_1 and F.
The 4-to-1 multiplexer shown in Figure 2 works as follows:

- If S_1 is 0 and S_0 is 0, then the binary value of I_0 is output to F.
- If S_1 is 0 and S_0 is 1, then the binary value of I_1 is output to F.
- If S_1 is 1 and S_0 is 0, then the binary value of I_2 is output to F.
- If S_1 is 1 and S_0 is 1, then the binary value of I_3 is output to F.

Namely, the binary value of I_n is output to F, where *n* is [B].
This specification can be expressed by the truth table shown in Table 2.

Table 2 Truth table of 4-to-1 multiplexer

Given binary values						Output
S_1	S_0	I_0	I_1	I_2	I_3	F
0	0	0	any	any	any	0
0	0	1	any	any	any	1
0	1	any	0	any	any	0
0	1	any	1	any	any	1
1	0	any	any	0	any	0
1	0	any	any	1	any	1
1	1	any	any	any	0	0
1	1	any	any	any	1	1

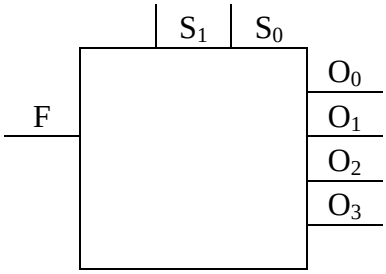
This specification can also be expressed by the Boolean equation as follows:

F = [C]

[Demultiplexer]

A demultiplexer circuit is a circuit which selects one output line among multiple output lines and outputs the binary value of an input line to the selected output line.

Figure 3 shows a schema of 1-to-4 demultiplexer.



F: Input line
S_0, S_1: Selection control lines
O_0, O_1, O_2, O_3: Output lines

Figure 3 Schema of 1-to-4 demultiplexer

A binary value, 0 or 1, is associated with each of F, S_0, S_1, O_0, O_1, O_2 and O_3.
The 1-to-4 demultiplexer shown in Figure 3 works as follows:

- If S_1 is 0 and S_0 is 0, then the binary value of F is output to O_0.
- If S_1 is 0 and S_0 is 1, then the binary value of F is output to O_1.
- If S_1 is 1 and S_0 is 0, then the binary value of F is output to O_2.
- If S_1 is 1 and S_0 is 1, then the binary value of F is output to O_3.
- In either case, the binary value 0 is output to other 3 output lines.

This specification can be expressed by the truth table shown in Table 3.

Table 3 Truth table of 1-to-4 demultiplexer

Given logic values			Outputs			
S_1	S_0	F	O_0	O_1	O_2	O_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

This specification can also be expressed by the Boolean equation. For example, the binary value on the output line O_2 is expressed as follows:

O_2 = D

Subquestion

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

Answer group for A

a)

0
0
1
1

b)

0
1
0
1

c)

0
1
1
0

d)

1
0
0
1

e)

1
0
1
0

f)

1
1
0
0

Answer group for B

a) $2 \times S_0 + S_1$ b) $2 \times S_1 + S_0$ c) $3 - (2 \times S_0 + S_1)$ d) $3 - (2 \times S_1 + S_0)$

Answer group for C

a) $(!S_1 \,\&\, !S_0 \,\&\, I_0) \mid (!S_1 \,\&\, S_0 \,\&\, I_1) \mid (S_1 \,\&\, !S_0 \,\&\, I_2) \mid (S_1 \,\&\, S_0 \,\&\, I_3)$

b) $(!S_1 \,\&\, !S_0 \,\&\, I_0) \mid (S_1 \,\&\, !S_0 \,\&\, I_1) \mid (!S_1 \,\&\, S_0 \,\&\, I_2) \mid (S_1 \,\&\, S_0 \,\&\, I_3)$

c) $(S_1 \,\&\, S_0 \,\&\, I_0) \mid (!S_1 \,\&\, S_0 \,\&\, I_1) \mid (S_1 \,\&\, !S_0 \,\&\, I_2) \mid (!S_1 \,\&\, !S_0 \,\&\, I_3)$

d) $(S_1 \,\&\, S_0 \,\&\, I_0) \mid (S_1 \,\&\, !S_0 \,\&\, I_1) \mid (!S_1 \,\&\, S_0 \,\&\, I_2) \mid (!S_1 \,\&\, !S_0 \,\&\, I_3)$

Answer group for D

a) $!((!S_1 \,\&\, !S_0 \,\&\, F) \mid (!S_1 \,\&\, S_0 \,\&\, F) \mid (S_1 \,\&\, S_0 \,\&\, F))$

b) $!((!S_1 \,\&\, !S_0 \,\&\, F) \mid (S_1 \,\&\, !S_0 \,\&\, F) \mid (S_1 \,\&\, S_0 \,\&\, F))$

c) $!S_1 \,\&\, S_0 \,\&\, F$

d) $S_1 \,\&\, !S_0 \,\&\, F$

Q2. Read the following description concerning the software requirements of a video rental system, and then answer Subquestions 1 and 2.

Company Z is a company that imports videos from many countries. Company Z is now planning to expand its business by opening video rental chain stores. Mr. G is an IT manager of the company, and is appointed to create a project for a video rental system development. He explained an overview of software requirements and business conditions of this system to his software development team, and asked the team to design and develop the system. The following description shows the information provided by Mr. G concerning the video rental services.

- The media that stores deal in are DVDs, BDs (Blu-ray Discs), and CDs.
- The rental charge is determined by the medium type. For example, BD rentals are more expensive than DVD rentals.
- The stores do video rental business only. They do not sell any products, magazines, foods or drinks.
- Each customer has to subscribe for a membership before he/she can rent any items in the store.
- A customer can log in to the application for searching the videos and other information. After finishing the search, he/she logs out the application.
- A customer can rent videos at the cashier counter, where the operation is performed by the clerk or the customer can do the self-rent using the self-rent system at the cashier counter.
- A clerk also can log in to the application for searching the videos and other information. After finishing the search, he/she logs out the application.
- The rental period is 7 days.
- A customer must return the rented videos at the cashier counter.
- On completion of a rental, a customer will receive a transaction receipt that contains the details of the rented videos including the date of return and the rental charge.
- In case of late return, the customer has to pay the fine for each item. The fine is determined by the rental charge and the number of days that exceeds the due date.
- All transactions of the rental charges and fines can be paid by cash or credit card.
- A clerk will manage the information on videos in the system.
- An administrator will maintain the system. Also, he/she will manage all user accounts and customer memberships.

Figure 1 shows the use case diagram of the video rental system.

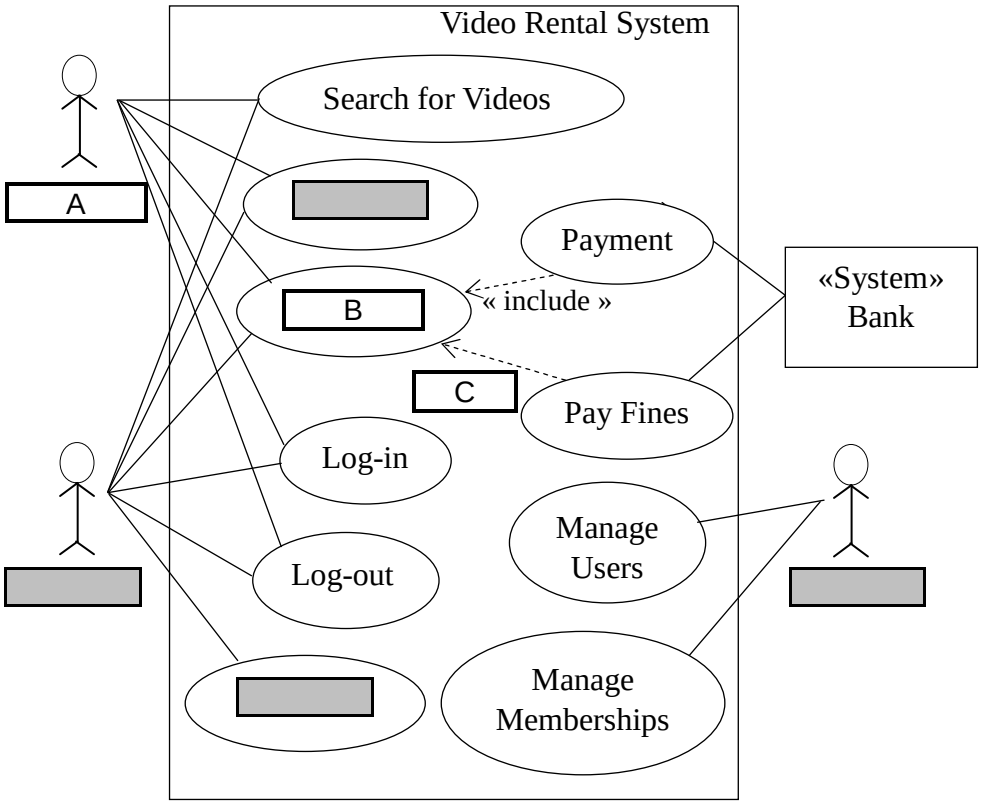


Figure 1 Use case diagram of the video rental system

Subquestion 1

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

Answer group

a) « extend »	b) « include »	c) Administrator
d) Clerk	e) Customer	f) Manage Videos
g) Rent Videos	h) Return Videos	

Subquestion 2

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

Figure 2 shows a part of the class diagram of the video rental system.

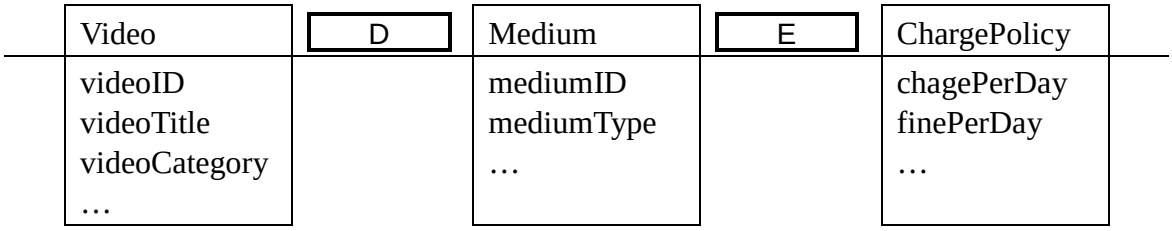


Figure 2 Class diagram of the video rental system (partial)

Answer group

a) 0..* 1 b) 0..1 1 c) 1 0..*

d) 1 0..1 e) 1 1

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

Company JKL is a recruitment agency for sailors. JKL's customers are shipping companies that want to hire sailors. Each shipping company has many vessels of various types. JKL recruits sailors of various ranks, starting from crew, second officer, first officer, engineer up to captain, and sends them to vessels of shipping companies.

JKL has a database system for sailor management for their working processes. The database system has the following tables.

Table: Sailor

Sailor_ID, Sailor_name, Date_of_birth, Marital_status, Address, Height, Weight, Blood_type, Shoe_size, Job_status (1: new comer, 2: signed on, 3: vacation, 4: ready, 5: inactive)

Table: Rank

Rank_ID, Rank_name, Rank_type (1: officer, 2: non-officer)

Table: Company

Company_ID, Company_name, Contact_person, Email, Phone, Address

Table: Vessel

Vessel_ID, Vessel_name, Company_ID, Vessel_type (tanker, cargo, …), Flag (Panama, Singapore, Japan, …)

Table: Service_history

Sequence_number, Sailor_ID, Rank_ID, Vessel_ID, Sign_on_date, Sign_on_port, Sign_off_date, Sign_off_port, Contract_period (number of months), Contract_end_date

New sailors come and apply for job at JKL. After qualification test and interview, sailors are registered into Sailor table as Job_status = 1 (new comer).
Many sailors registered at JKL are currently working on various vessels of shipping companies. They are recorded as Job_status = 2 (signed on). A record is added to Service_history table for every sea service. Service_history table contains the information such as "Sailor John is working on vessel MV Oasis with the rank of second officer signed on dated 2013-09-01 at Singapore with the contract period of 12 months and contract end date (target date for going home) is 2014-08-31". While they are working on board, Sign_off_date is set to NULL.

Sailors who have accomplished the contract period are signed off from the vessels and go back to their home countries for vacation. Sign_off_date is set in the corresponding record in Service_history table. Also, Job_status in Sailor table is changed to 3 (vacation).
After vacation, the sailors come and report at JKL that they are ready for next assignment. Then, Job_status in Sailor table is changed to 4 (ready).

When there is a vacancy for a sailor of a specific rank and specific vessel type (e.g. second officer for tanker vessel MV Royal Jesper of shipping company ABC), a JKL staff makes an inquiry to select a suitable sailor on the condition that Job_status is 1 (new comer) or 4 (ready), by searching Sailor table. When a sailor is selected, a job contract is signed between JKL and that sailor, and his Job_status is changed to 2 (signed on). Then, the sailor goes to the country where the vessel is currently anchored, and signs on to the vessel. He will work on the vessel until the end of the contract period.
After accomplishing the job contract, some sailors retire due to old age and some do not come back to JKL. For those sailors, Job_status is changed to 5 (Inactive).

Subquestion 1

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

There is a vacancy for first officer for tanker. The following SQL statement outputs a list of ready sailors who have experience of working on tanker with any officer post.

```
SELECT  Sailor_name, Vessel_name, Rank_name,
        Sign_on_date, Sign_off_date
FROM  Service_history h
JOIN  Sailor s  ON  h.Sailor_ID = s.Sailor_ID
JOIN  Rank   r  ON  h.Rank_ID   = r.Rank_ID
JOIN  Vessel v  ON  h.Vessel_ID = v.Vessel_ID
WHERE  [      A      ]  AND  [      B      ]
```

Output example:

Sailor name	Vessel name	Rank name	Sign on date	Sign off date
San Minn	MV Cruiser	2nd Officer	2013-04-01	2014-03-31
Aung Myo	MV Buddy	1st Officer	2013-01-01	2013-12-31

Answer group

a) `Job_status = 2`
b) `Job_status = 4`
c) `Rank_type = 1`
d) `Sign_off_date IS NOT NULL`
e) `Sign_off_date IS NULL`

Subquestion 2

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

Management wants a report on the head count of currently working sailors. The following SQL statement outputs a list of vessels and the number of signed on sailors on board.

```
SELECT  Company_name, Vessel_name, Flag,
        SUM(SO) Officers, SUM(SN) Non_officers
FROM ( SELECT  Company_name, Vessel_name, Flag, [ C ]
         FROM  Service_history h
         JOIN  Rank    r  ON  h.Rank_ID    = r.Rank_ID
         JOIN  Vessel  v  ON  h.Vessel_ID  = v.Vessel_ID
       JOIN  Company c  ON  v.Company_ID = c.Company_ID
         WHERE  Rank_type = 1 AND Sign_off_date [ D ]
         GROUP BY  Company_name, Vessel_name, Flag
         [ E ]
         SELECT  Company_name, Vessel_name, Flag, [ F ]
         FROM  Service_history h
         JOIN  Rank    r  ON  h.Rank_ID    = r.Rank_ID
         JOIN  Vessel  v  ON  h.Vessel_ID  = v.Vessel_ID
         JOIN  Company c  ON  v.Company_ID = c.Company_ID
         WHERE  Rank_type = 2 AND Sign_off_date [ D ]
         GROUP BY  Company_name, Vessel_name, Flag
     ) t
GROUP BY  Company_name, Vessel_name, Flag
HAVING  SUM(SO) > 0 OR SUM(SN) > 0
```

Output example:

Company name	Vessel name	Flag	Officers	Non officers
SkyLine	MV Oasis	Panama	3	15
TLT	MV Mandalay	Singapore	2	8
Venus Shipping	MV Athena	Panama	5	0

Answer group for C and F

a) 0 SO, COUNT(*) SN
b) COUNT(*) SO, 0 SN
c) COUNT(*) SO, COUNT(*) SN
d) SUM(SO), SUM(SN)

Answer group for D and E

a) IS NOT NULL
b) IS NULL
c) JOIN
d) NOT IN
e) OR
f) UNION ALL

Q4. Read the following description concerning the process of sending e-mail, and then answer Subquestions 1 through 3.

The most widely used protocol for transmission of e-mail is the Simple Mail Transfer Protocol (SMTP). SMTP assumes that the content of the message is text only. Another electronic mail standard, known as the Multipurpose Internet Mail Extensions (MIME), is further introduced to support transmission of multimedia contents.

Once an e-mail client presses the "send" button, the e-mail is submitted to the mail server for relaying through SMTP sender. SMTP sender then communicates with SMTP receiver at the recipients' site to transfer the e-mail and preserve them in user mailboxes. The receiving client then uses either the Post Office Protocol (POP), or Internet Message Access Protocol (IMAP), or a proprietary system to access their mailbox accounts on the mail server.

Figure 1 shows the outline of e-mail transfer.

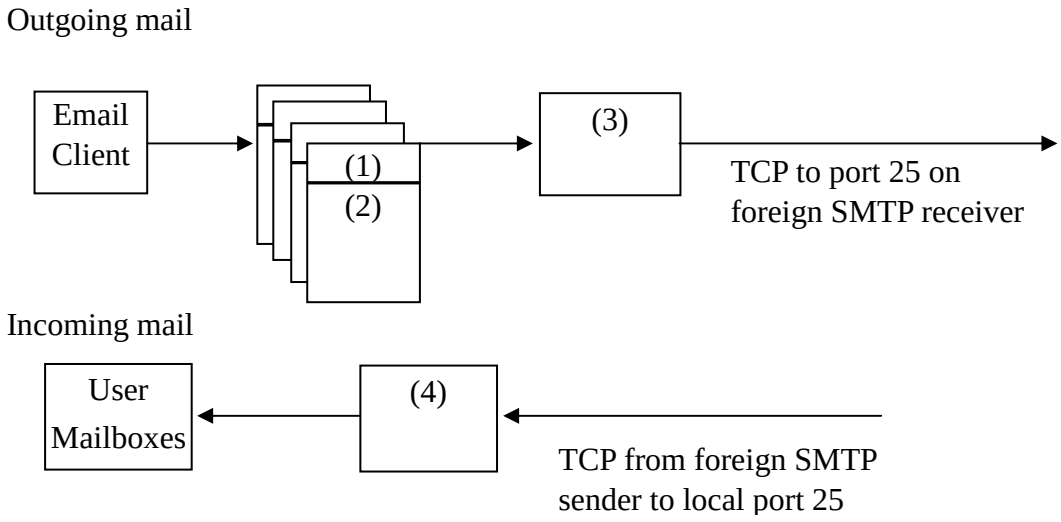


Figure 1 Outline of e-mail transfer

SMTP is operated based on series of commands and responses between the SMTP sender and SMTP receiver. SMTP sender at first establishes a TCP connection and then sends commands to the SMTP receiver over the communication channels. Once a TCP connection is established, the SMTP sender may send one or more e-mail to the SMTP receiver. The header or message envelope of each e-mail includes necessary information required for recipient(s) of the e-mail body. SMTP transfers messages in efficient way in the sense that it does not send the same message to the same destination twice.

Subquestion 1

In Figure 1, (1) and (2) represent two parts of an e-mail whereas (3) and (4) represent two key agents responsible for e-mail communications. From the answer group below, select the correct combination of the terms correspond to (1) through (4) in Figure 1.

Answer group

	(1)	(2)	(3)	(4)
a)	Body	Header	Client Sender	Inbox
b)	Body	Header	Outbox	Inbox
c)	E-mail	Header	Client Sender	Mail Server
d)	Header	Body	Mail server	Client Receiver
e)	Header	Body	SMTP Sender	SMTP Receiver
f)	Header	E-mail	Outbox	SMTP Receiver

Subquestion 2

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

Figure 2 shows an example of SMTP session where an SMTP sender has successfully sent an e-mail to the designated SMTP receiver.

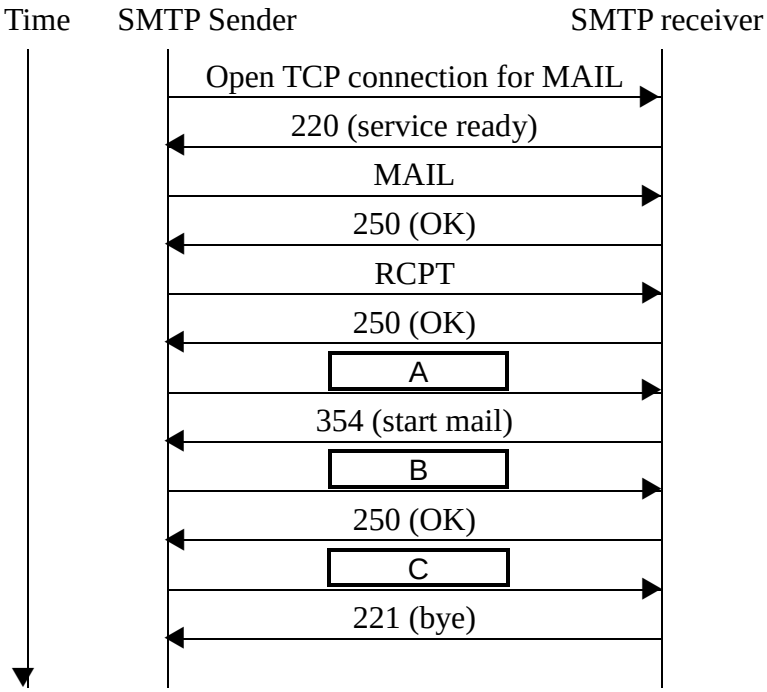


Figure 2 Example of SMTP session

In SMTP session in Figure 2, the phases of transferring a message are as follows:

(1) MAIL command identifies the originator of the message. If the command was executed successfully, reply code 250 (OK) is returned. This command gives the reverse path, which can be used to report errors.

(2) RCPT command identifies an individual recipient of the mail data. One or more RCPT commands can be issued. For each RCPT command, reply code 250 (OK) is returned if the command was executed successfully.

(3) DATA command transfers the message text. The SMTP sender uses DATA command to initiate the transfer of the message. If the reply code 354 is returned, the sender proceeds to send the message over the TCP connection. A line containing only a period indicates the end of the message.

(4) QUIT command closes the TCP connection.

Answer group

a) 250 (OK) b) DATA c) MAIL
d) QUIT e) RCPT f) Send message

Subquestion 3

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

An e-mail header consists of date and time, sender's e-mail address, receivers' e-mail addresses, carbon copy (CC), blind carbon copy (BCC), and subject of the message.

Date: Tuesday, April 1, 2014 Time: 10:37:17 GMT
From: "Suman Biswas" <suman@alpha.com>
To: "John Smith" <smith@beta.com>, helal@beta.com, "Jean Lane" <jln@delta.com>
Cc: hannan@alpha.com
Bcc: lim@gamma.org, leo@alpha.org
Subject: The Syntax in RFC 822

Consider the e-mail header shown above. The e-mail has [D] destination servers and therefore [D] copies of e-mail are to be produced. When John Smith receives this e-mail, he can find maximum [E] e-mail addresses from this header excluding his own e-mail address.

Answer group

a) 3 b) 4 c) 5 d) 6 e) 7 f) 8

Q5. Read the following description concerning a program development, and then answer Subquestions 1 and 2.

[Program Description]

This is an income tax calculation program for a certain country. When a user starts the program, a screen titled "Income Tax Calculator" is displayed. Figure 1 shows an example of income tax calculator screen. With this screen, when the user enters input data in the fields (a) through (c), the program calculates the amount of income tax, and displays the results in the fields (d) through (j).

The process for calculating the income tax is as follows:

(1) Receives data entered by the user.
  (a) Monthly income: His/her average monthly income in local currency.
  (b) Have spouse: 'Y' if he/she has a spouse who is not working, otherwise 'N'.
  (c) Number of children: The number of his/her children.

(2) Calculates the annual income, 3 types of exemptions, and income to assess tax.
  (d) Annual income: 12 × (a).
  (e) Basic exemption: 20% of (d).
  (f) Exemption for spouse: 300 if (b) is 'Y', otherwise 0.
  (g) Exemption for children: 200 × (c).
  (h) Income to assess tax: (d) – (e) – (f) – (g).

(3) Calculates the income tax. Here, if (h) ≤ 500, the income tax is wholly exempted.
  (i) Income tax per year: If (h) > 500, calculate the tax by using Table 1, otherwise 0.
  (j) Income tax per month: (i) ÷ 12

	Income Tax Calculator		Unit: Dollars
	Input data		
(a)	Monthly income	500	
(b)	Have spouse	Y	"Y" if yes
(c)	Number of children	2	
	Calculated results		
(d)	Annual income	6,000	12 × (a)
(e)	Basic exemption	1,200	
(f)	Exemption for spouse	300	
(g)	Exemption for children	400	
(h)	Income to assess tax	4,100	(d) – (e) – (f) – (g)
(i)	Income tax per year	167	
(j)	Income tax per month	13	(i) ÷ 12

Figure 1 Example of income tax calculator screen

In (3) above, Income tax per year (i) is calculated by using Table 1. The last two columns (y) and (z) show the sample data when the amount of Income to assess tax (h) is 4,100.
The program divides and stores the amount of Income to assess tax (h) into the cells in Allocated income (y) column from top to bottom, so that each cell should have the value to the maximum of Amount (w) on the same row.
The program then calculates Tax to pay (z) for each row, with (z) = (y) × (x) ÷ 100. Finally, Income tax per year (i) is obtained by totaling the values in Allocated income (y) column.

Table 1 Income tax calculation table

From	To	Amount (w)	Tax rate % (x)	Allocated income (y)	Tax to pay (z)
1	500	500	1	500	5
501	1,000	500	2	500	10
1,001	1,500	500	3	500	15
1,501	2,000	500	4	500	20
2,001	3,000	1,000	5	1,000	50
3,001	4,000	1,000	6	1,000	60
4,001	6,000	2,000	7	100	7
6,001	8,000	2,000	8		
8,001	10,000	2,000	10		
10,001	15,000	5,000	12		
15,001	20,000	5,000	15		
20,001	unlimited	the rest	20		
			Total	4,100	167

The program handles all numeric values as integers. With division of integers, integer quotient is returned as a result, that is, the figures below the decimal point are omitted.
Assuming that the arrays `From`, `To`, `Amount` and `TaxRate` have already been initialized with the values shown in Table 1, and the arrays `AllocatedIncome` and `TaxToPay` have already been initialized with 0s. Indexes of the arrays start at 0.
The program uses the following two functions:

`GetData()`: Get the data from the input data area on the screen.

`PutData()`: Put the data to the calculated results area on the screen.

[Program]

```
○Integer: MonthlyIncome, NumberOfChildren, AnnualIncome,
          BasicExemption, ExemptionForSpouse, ExemptionForChildren,
          IncomeToAssessTax, IncomeTaxPerYear, IncomeTaxPerMonth
○Character: HaveSpouse
○Integer: From[12], To[11], Amount[11], TaxRate[12],
          AllocatedIncome[12], TaxToPay[12]
○Integer: i

• GetData (MonthlyIncome, HaveSpouse, NumberOfChildren)
• AnnualIncome ← 12 × MonthlyIncome
• BasicExemption ← AnnualIncome × 20 ÷ 100
▲ HaveSpouse = 'Y'
  • ExemptionForSpouse ← 300
──
  • ExemptionForSpouse ← 0
▼
• ExemptionForChildren ← 200 × NumberOfChildren
• IncomeToAssessTax ← AnnualIncome - BasicExemption -
                      ExemptionForSpouse - ExemptionForChildren
• IncomeTaxPerYear ← 0
▲ [      A      ]
  • AllocatedIncome[0] ← IncomeToAssessTax
  ■ i: 0, (i < 11) and (AllocatedIncome[i] > 0), 1
    ▲ AllocatedIncome[i] > Amount[i]
      • [      B      ]
      • [      C      ]
    ▼
  ■
  ■ i: 0, ( [      D      ] ) and (AllocatedIncome[i] > 0), 1
    • TaxToPay[i] ← AllocatedIncome[i] × TaxRate[i] ÷ 100
    • IncomeTaxPerYear ← IncomeTaxPerYear + TaxToPay[i]
  ■
▼
• IncomeTaxPerMonth ← IncomeTaxPerYear ÷ 12
• PutData (AnnualIncome, BasicExemption, ExemptionForSpouse,
           ExemptionForChildren, IncomeToAssessTax,
           IncomeTaxPerYear, IncomeTaxPerMonth)
```

Subquestion 1

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

Answer group for A

a) `AnnualIncome > 0`
b) `AnnualIncome > 500`
c) `IncomeToAssessTax > 0`
d) `IncomeToAssessTax > 500`

Answer group for B and C

a) `AllocatedIncome[i] ← AllocatedIncome[i] - Amount[i]`
b) `AllocatedIncome[i] ← Amount[i]`
c) `AllocatedIncome[i] ← Amount[i] - AllocatedIncome[i]`
d) `AllocatedIncome[i+1] ← AllocatedIncome[i]`
e) `AllocatedIncome[i+1] ← AllocatedIncome[i] - Amount[i]`
f) `AllocatedIncome[i+1] ← Amount[i] - AllocatedIncome[i]`

Answer group for D

a) `i < 10`
b) `i < 11`
c) `i < 12`
d) `TaxToPay[i] = 0`

Subquestion 2

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

The program calculates Tax to pay by the following statement:

- `TaxToPay[i]` ← `AllocatedIncome[i]` × `TaxRate[i]` ÷ `100` … (*)

Assuming that, when an expression has two or more arithmetic operators with the same priority, they are executed from left to right.

Here are three statements that are mathematically equivalent to the statement (*):

- `TaxToPay[i]` ← `AllocatedIncome[i]` ÷ `100` × `TaxRate[i]` … (1)
- `TaxToPay[i]` ← `TaxRate[i]` × `AllocatedIncome[i]` ÷ `100` … (2)
- `TaxToPay[i]` ← `TaxRate[i]` ÷ `100` × `AllocatedIncome[i]` … (3)

Among the statements (1) through (3), the statement(s) that is(are) computationally equivalent to the statement (*) is(are) [E].

The program calculates Income tax per month by the following statement:

- `IncomeTaxPerMonth` ← `IncomeTaxPerYear` ÷ `12`

When the value of `IncomeTaxPerYear` is 167, the real value of Income tax per month will be 167 ÷ 12 = 13.91⋯, however, the value obtained in `IncomeTaxPerMonth` is 13, because the fractions are omitted according to the division specification.

In such a case, it is advisable that the result should be rounded off to the nearest integer, instead of omitting the fractions. For that, the above statement should be changed to:

- `IncomeTaxPerMonth` ← [F]

Answer group for E

a) (1) b) (1) and (2) c) (1) and (3)
d) (2) e) (2) and (3) f) (3)

Answer group for F

a) `(IncomeTaxPerYear + 5) ÷ 12` b) `(IncomeTaxPerYear + 6) ÷ 12`
c) `(IncomeTaxPerYear + 9) ÷ 12` d) `(IncomeTaxPerYear - 5) ÷ 12 + 1`

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

The geometric sweeping method is for finding information about coverings of figures. An example of the problem about coverings of figures is shown in Figure 1.

In Figure 1, there are 4 numbered segments marked S1, S2, S3, S4, and a background segment. S4 is the closest one to the observer. Segments are opaque, so some of them can be partly or completely behind another one. For each given segment, the problem finds the total length of visible parts of the segment. In this case, the length of visible part of S4 is 5 (from 0 to 5), S3 is 4 (from -4 to 0), S2 is 3 (from 5 to 8), and S1 is 0 (not visible).

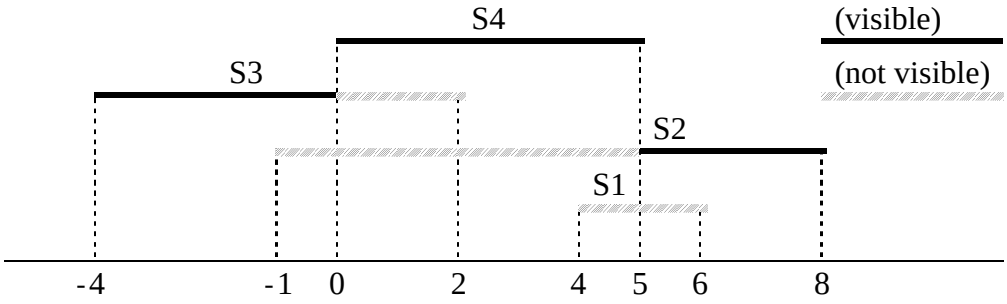


Figure 1 Example of the problem about coverings of figures

[Program Description]

Subprogram `SegmentCovering()` is a one dimensional implementation of the geometric sweeping method.

(1) Structure `EndPoint` is defined to store information about segments. Structure `EndPoint` has 3 members `c`, `s` and `e`. A set of `c`, `s` and `e` identifies one end-point of one segment. Member `c` holds the coordinate value of the end-point, member `s` holds the segment number to which the end-point belongs, and member `e` holds the constant `LEFT` or `RIGHT` which signifies left-end of the segment or right-end of the segment. For example, when `c`, `s` and `e` hold the values 8, 2 and `RIGHT` respectively, it indicates that the coordinate value of the right-end point of segment number 2 is 8.

(2) Table 1 shows the global variables used in the program.

Table 1 Global variables used in the program

Variable	Description
`n`	Number of the given segments
`Points[2×n]`	Information about end-points of each segment.
`Active[n]`	The status (1: active, 0: inactive) of each segment.
`Length[n]`	Total length of visible parts of each segment.

(3) `Points[2×n]` is `EndPoint` type structured array. Since each segment has 2 end-points, the array has 2 × `n` elements.

(4) When a member is referred to, for example, when member `c` of `Points[1]` is referred to, it is described as `Points[1].c`.
(5) A segment with larger segment number is closer to the observer.
(6) In this program, the geometric sweeping method proceeds as per the following steps:
   1) Clear `Active[]` and `Length[]` with 0s.
   2) Sort `Points[]` in ascending order of the coordinate value.
   3) Set `i` to 1.
   4) If the `i`-th end-point is the left-end point of segment number *s*, set `Active[`*s*`]` to 1. If it is the right-end point of segment number *s*, set `Active[`*s*`]` to 0.
   5) Find the segment number `max` which is the largest index of `Active[]` elements that have value 1.
   6) Add the value `Points[i+1].c - Points[i].c` to `Length[max]`.
   7) Increase `i` by 1.
   8) Repeat 4) through 8) until `i` is greater than 2 × `n` − 1.
(7) Indexes of the arrays start at 1.

Subprogram `SegmentCovering()` is executed for the sample problem given in Figure 1. Table 2 shows the contents of `Points[]` after executing step 2) of (6), and Table 3 shows the contents of `Active[]` after executing step 4) of (6) in each iteration.

Table 2 Contents of `Points[]`

k	`Points[k].c`	`Points[k].s`	`Points[k].e`
1	-4	3	LEFT
2	-1	2	LEFT
3	0	4	LEFT
4	2	3	RIGHT
5	4	1	LEFT
6	5	4	RIGHT
7	6	1	RIGHT
8	8	2	RIGHT

Table 3 Contents of `Active[]` in each iteration

i	`Active[1]`	`Active[2]`	`Active[3]`	`Active[4]`
1	0	0	1	0
2	0	1	1	0
3	A			
4	0	1	0	1
5	1	1	0	1
6	1	1	0	0
7	0	1	0	0

[Program]

```
OGlobal: Integer:  n, Active[n], Length[n]
OGlobal: Structure EndPoint:  Points[2×n]

OSubprogram:  SegmentCovering()
OInteger:  i, j, max
OStructure EndPoint:  t

■ i: 1, i ≤ n, 1
  • Active[i] ← 0
  • Length[i] ← 0
■

■ i: 1, i ≤ 2×n-1, 1
  ■ j: i+1, j ≤ 2×n, 1
      B
    • t ← Points[i]
    • Points[i] ← Points[j]
    • Points[j] ← t
  ■
■

■ i: 1, i ≤ 2×n-1, 1
    Points[i].e = LEFT
    • Active[     C     ] ← 1
    ──────────────────────
    • Active[     C     ] ← 0
  • max ← FindMax()
    max > 0
    • Length[max] ← Length[max] + (Points[i+1].c - Points[i].c)
■

OSubprogram:  FindMax()
OInteger:  i, m

• m ← -1
■ i: 1, i ≤     D     , 1
    E
  • m ← i
■
• return m
```

Subquestion 1

From the answer group below, select the correct answer to be inserted into the blank [ ] in Table 3 and in the above program.

Answer group for A

a) | 0 | 0 | 0 | 0 |

b) | 0 | 0 | 1 | 0 |

c) | 0 | 1 | 0 | 0 |

d) | 0 | 1 | 1 | 1 |

e) | 1 | 1 | 1 | 0 |

Answer group for B

a) `Points[i].c < Points[j].c`
b) `Points[i].c > Points[j].c`
c) `Points[i].e = Points[j].e`
d) `Points[i].e ≠ Points[j].e`
e) `Points[i].s < Points[j].s`
f) `Points[i].s > Points[j].s`

Answer group for C

a) `i`
b) `n`
c) `Points[i].c`
d) `Points[i].s`
e) `Points[i+1].c`
f) `Points[i+1].s`

Answer group for D

a) `2 × n`
b) `i`
c) `n`
d) `Points[i].s`

Answer group for E

a) `Active[i] = 0`
b) `Active[i] = 1`
c) `Points[i].e = LEFT`
d) `Points[i].e = RIGHT`

Subquestion 2

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

In the above implementation, most of the segments are activated one time and inactivated one time. Only the segment [F] is left in active state.

Answer group

a) number 1
b) number n
c) with the maximum left-end coordinate
d) with the maximum right-end coordinate
e) with the minimum left-end coordinate
f) with the minimum right-end coordinate

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

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

The Barycentric technique is one method that can determine if a point is within a triangle.

[Program Description]
The program holds the *x*-*y* coordinate values of 5 triangles (numbered 0 to 4) and 5 points (numbered 0 to 4), and determines if each point is within how many number of triangles. Figure 1 shows an example of 5 triangles and 5 points.

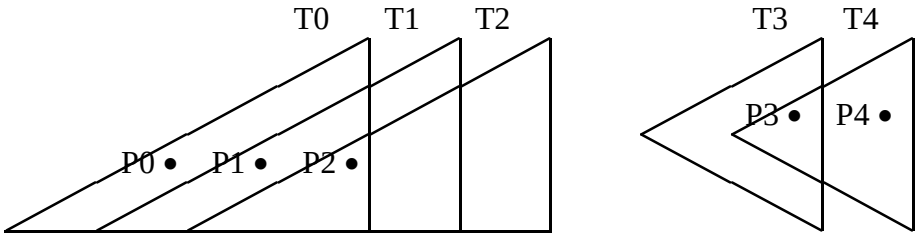


Figure 1 Example of 5 triangles (T0 to T4) and 5 points (P0 to P4)

In this case, the program determines that P0 is within T0, P1 is within T0 and T1, and so on, and outputs the result as follows:

```
Point 0 is in 1 triangle(s)
Point 1 is in 2 triangle(s)
Point 2 is in 3 triangle(s)
Point 3 is in 2 triangle(s)
Point 4 is in 1 triangle(s)
```

(1) A point is defined by the structure `point`, and a triangle is defined by the structure `triangle`, as follows:

```
struct point { int x, y; };
struct triangle { struct point A, B, C; };
```

(2) The array `struct point list_P[5]` holds the coordinate values of 5 given points, and the array `struct triangle list_T[5]` holds the coordinate values of 5 given triangles. The coordinate values are loaded into `list_P[]` and `list_T[]` by the functions `initialize_list_P()` and `initialize_list_T()` respectively.

(3) The function `in_triangle(struct point P, struct triangle T)` determines whether point `P` is within triangle `T` or not by the following steps. Here, descriptions of the algorithm and mathematical backgrounds are omitted.

(i) Calculates the difference of the coordinate values between one vertex of `T` and other 2 vertices of `T` and point `P`. Results are stored into `struct vector v0`, `v1` and `v2`.

(ii) Calculates the vector values by calling the function `dot()`, by using the values obtained in (i).

(iii) Calculates the values of `u` and `v`, by using the values obtained in (ii).

(iv) Determines if point `P` is within triangle `T` or not, based on the following criteria.

If `u` < 0 or `v` < 0, then point `P` is outside triangle `T`.

If `u` > 1 or `v` > 1, then point `P` is also outside triangle `T`.

If `u` + `v` > 1, then point `P` is also outside triangle `T`.

Otherwise, point `P` is within triangle `T`.

(v) Returns 1 if point `P` is within triangle `T`, otherwise returns 0.

[Program]

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

struct point { int x, y; };
struct triangle { struct point A, B, C; };
struct vector { int x, y; };

struct point    list_P[5];              // Array for 5 points
struct triangle list_T[5];              // Array for 5 triangles

float dot(struct vector va, struct vector vb) {
   return va.x * vb.x + va.y * vb.y;
}

int in_triangle(struct point P, struct triangle T) {
   float  dot00, dot01, dot02, dot11, dot12;
   float  dv, u, v;
   struct vector v0, v1, v2;

   v0.x = T.C.x - T.A.x;
   v0.y = T.C.y - T.A.y;
   v1.x = T.B.x - T.A.x;
   v1.y = T.B.y - T.A.y;
   v2.x = P.x - T.A.x;
   v2.y = P.y - T.A.y;
```

```
   dot00 = dot(v0, v0);
   dot01 = dot(v0, v1);
   dot02 = dot(v0, v2);
   dot11 = dot(v1, v1);
   dot12 = dot(v1, v2);
   dv = dot00 * dot11 - dot01 * dot01;
   u = (dot11 * dot02 - dot01 * dot12) / dv;
   v = (dot00 * dot12 - dot01 * dot02) / dv;

   if (      A      )
      return 1;
   else
      return 0;
}

void main() {
   int  i, j, triangle_cnt;

   initialize_list_P();
   initialize_list_T();

   for (i = 0; i < 5; i++) {            // index for points
      triangle_cnt = 0;
      for (j = 0; j < 5; j++) {         // index for triangles
         if (      B      ) == 1)
            triangle_cnt += 1;
      }
      printf("Point %d is in %d triangle(s)\n", i, triangle_cnt);
   }
}
```

Subquestion 1

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

Answer group for A

a) `(u + v > 1) && (v > 1) && (u > 1) && (u < 0) && (v < 0)`
b) `(u < 0) && (v < 0) && (u + v > 1)`
c) `(u < 0) || (v < 0) || (u > 1) || (v > 1)`
d) `(u > 0) && (v > 0) && (u + v < 1)`
e) `(u >= 0) && (v >= 0) && (u + v <= 1)`
f) `(u >= 0) || (v >= 0) || (u + v <= 1)`

Answer group for B

a) `in_triangle(list_P[i], list_T[j])`
b) `in_triangle(list_P[j], list_T[i])`
c) `in_triangle(list_T[i], list_P[j])`
d) `in_triangle(list_T[j], list_P[i])`

Subquestion 2

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

On this program, when invalid coordinate values are given, it may be that "zero divide exception" will occur. Specifically, this exception will occur in function `in_triangle()` when the value of [C] is equal to the value of [D].

Answer group

a) 0
b) 1
c) `dot00 * dot11`
d) `dot01 * dot01`
e) `dot01 * dot12`
f) `dot11 * dot02`

Subquestion 3

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

Assuming that [B] contains the correct answer in Subquestion 1.

The program `main()` is revised to output "Point *p* is in the largest number of triangles" and "Point *p* is in Triangle *a*, Triangle *b*, … " additionally. For example, when the revised program is executed for the data shown in Figure 1, the added last 2 lines are as follows:

```
Point 2 is in the largest number of triangles
Point 2 is in Triangle 0, Triangle 1, Triangle 2
```

[Revised Program]

```
void main() {
   int  i, j, k, triangle_cnt;
   int  m = 0, mp = 0;
   int  triangle_list[5], m_list[5];

   initialize_list_P();
   initialize_list_T();

   for (i = 0; i < 5; i++) {            // index for points
      triangle_cnt = 0;
      for (j = 0; j < 5; j++) {         // index for triangles
         if ( [ B ] == 1) {
            [ E ];
            triangle_cnt += 1;
         }
      }
      printf("Point %d is in %d triangle(s)\n", i, triangle_cnt);
      if (triangle_cnt > m) {
         mp = i;
         m = triangle_cnt;
         for (k = 0; k < m; k++)
            m_list[k] = triangle_list[k];    <-------- α
      }
   }
   if (m > 0) {
      printf("Point %d is in the largest no of triangles\n", mp);
      printf("Point %d is in ", mp);
      for (k = 0; k < m-1; k++)
         printf("Triangle %d, ", m_list[k]);
      printf("Triangle %d\n", m_list[m-1]);
   }
}
```

The revised program is executed for the data shown in Figure 2.

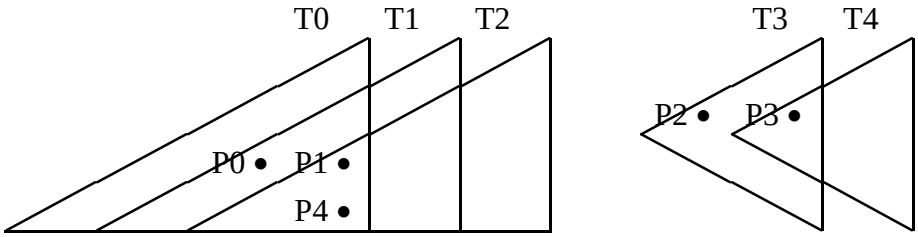


Figure 2 Another example of 5 triangles and 5 points

In this case, the statement marked by α will be executed [F] times. And finally, the following output will be obtained.

```
Point 0 is in 2 triangle(s)
Point 1 is in 3 triangle(s)
Point 2 is in 1 triangle(s)
Point 3 is in 2 triangle(s)
Point 4 is in 3 triangle(s)
Point [ G ] is in the largest number of triangles
Point [ G ] is in Triangle 0, Triangle 1, Triangle 2
```

Answer group for E

a) `m_list[i] = j`
b) `m_list[j] = i`
c) `triangle_list[i] = j`
d) `triangle_list[triangle_cnt] = i`
e) `triangle_list[triangle_cnt] = j`
f) `triangle_list[j] = i`

Answer group for F and G

a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

Q8. Read the following description of a Java program and the program itself, and then answer Subquestions 1 and 2.

This program assists a user in playing a game named Tic Tac Toe. Tic Tac Toe is a game in which 2 players put the marks X and O alternately on a board with 3 rows and 3 columns. Initially, all 9 cells are vacant. On a turn, each player puts his/her mark in any vacant cell. The game ends when one player wins by getting 3 his/her marks in a row, horizontally, vertically or diagonally. The game ends in a tie when all 9 cells are filled without win.
Figure 1 shows some examples of wins and a tie.

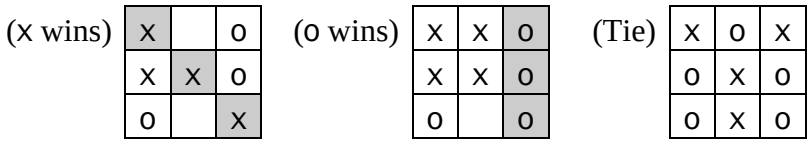


Figure 1 Examples of wins and a tie

[Program Description]

(1) The program plays against a user. The program plays as X, and the user plays as O. The player X moves first.

(2) The array `squares[][]` holds the status of each cell on the board. There are 3 status codes: `'X'`(occupied by X), `'O'`(occupied by O), and `'.'`(the cell is empty).

(3) Major methods provided in class `TicTacToe` are as follows:

- Method `isGameOver` determines whether the game is over or not.
- Method `calcScore` checks each of the 8 possible victory lines (3 vertical, 3 horizontal, and 2 diagonal lines), and returns the score value. There are 3 score values: 1 (X won), -1 (O won), and 0 (otherwise).
- Method `playBestMove` determines the best cell to put X. It tests each of the empty cells one by one to find the score value of the cell by calling method `minimaxForO`. After all empty cells are tested, it puts X into the cell that has the best (maximum) score value.
- Methods `minimaxForO` and `minimaxForX` call each other to try all possible moves out to the end of the game to find the score value. Method `minimaxForO` is trying to find the minimum score value returned from method `minimaxForX`, because the score value -1 means O won. On the other hand, method `minimaxForX` is trying to find the maximum score value returned from method `minimaxForO`, because the score value 1 means X won.

(4) Methods `minimaxForO` and `minimaxForX` are designed based on the minimax algorithm to determine how the opponent might reply.
Figure 2 shows how method `playBestMove` tests all possible moves and finds the best

cell to put X with the minimax algorithm. Assuming that the current status of the board is as shown in 1). Method `playBestMove` temporarily puts X in each of the 3 empty cells as 2), 3) and 4), and calls method `minimaxForO` for each case. In case 2), called method `minimaxForO` temporarily puts O in each of the 2 empty cells as 21) and 22), and finds that the score value of 21) is -1 and that of 22) is 0. So it returns the best (minimum) score value -1. In this way, the score values of 2), 3) and 4) are determined as -1, 0 and -1 respectively. Method `playBestMove` finds that the best (maximum) score value among them is 0, so it puts X as in 3).

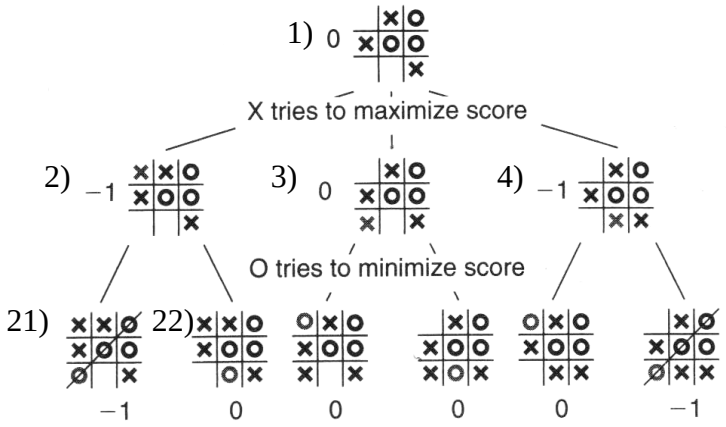


Figure 2 The minimax algorithm at work

(5) When the user inputs his/her choice, it is assumed that he/she correctly enters the row and column numbers of the selected cell and that the selected cell is empty.

(6) The results of executing the method `main` are as follows:

```
Welcome to Tic Tac Toe.

X..
...
...

Enter row: 2
Enter column: 2
X.X
...
..O

Enter row: 1
Enter column: 1
XXX
.O.
..O

Game over.
```

[Program]

```
import java.util.Scanner;

public class TicTacToe {

   /* For reading from the console */
   public static final Scanner INPUT = new Scanner(System.in);

   /* Squares on the board, each of which is '.', 'X', or 'O' */
   private char[][] squares;
   char player;

   public TicTacToe() {
      squares = new char[][] {{'.', '.', '.'},
                              {'.', '.', '.'},
                              {'.', '.', '.'}};
   }

   public boolean isGameOver() {
      if (calcScore() != 0)  return true;
      for (int row = 0; row < 3; row++) {
         for (int column = 0; column < 3; column++) {
            if (squares[row][column] == '.')  return false;
         }
      }
      return true;
   }

   /* Return the value of the current position if it is O's turn */
   protected int minimaxForO() {
      int score = calcScore();
      if (isGameOver())  return score;
      int bestScore = 2;
      for (int row = 0; row < 3; row++) {
         for (int column = 0; column < 3; column++) {
            if (squares[row][column] == '.') {
               squares[row][column] = 'O';
               score = minimaxForX();
               if (score < bestScore)  bestScore = score;
               squares[row][column] = '.';
            }
         }
      }
      return bestScore;
   }

   /* Return the value of the current position if it is X's turn */
   protected int minimaxForX() {
      int score = calcScore();
      if (isGameOver())  return score;
      int bestScore = -2;
```

```
   for (int row = 0; row < 3; row++) {
      for (int column = 0; column < 3; column++) {
         if (squares[row][column] == '.') {
            squares[row][column] = 'X';
            score = minimaxForO();
            if (score > bestScore)  bestScore = score;
            squares[row][column] = '.';
         }
      }
   }
   return bestScore;
}

/* Play one game */
public void play() {
   player = 'X';
   for (int move = 0; move < 9; move++) {
      if (isGameOver())  return;
      if (player == 'X') {
         [      A      ];
         player = 'O';
      } else {
         System.out.println(this);
         System.out.print("Enter row: ");
         int row = INPUT.nextInt();
         System.out.print("Enter column: ");
         int column = INPUT.nextInt();
         [      B      ];
         player = 'X';
      }
   }
}

/* Find the best move for X and play it on the board */
protected void playBestMove() {
   int score;
   int bestScore = -2;
   int bestRow = -1;
   int bestColumn = -1;
   for (int row = 0; row < 3; row++) {
      for (int column = 0; column < 3; column++) {
         if (squares[row][column] == '.') {
            squares[row][column] = 'X';
            score = [      C      ];
            if (score > bestScore) {
               bestScore = score;
               bestRow = row;
               bestColumn = column;
            }
            [      D      ] = '.';
         }
```

```
            }
         }
         E          = 'X';
      }

      /* Return 1 if X has won, -1 if O has won, and 0 otherwise */
      public int calcScore() {
         int lineScore;
         for (int i = 0; i < 3; i++) {
            lineScore = scoreLine(squares[i][0],
                                  squares[i][1],
                                  squares[i][2]);
            if (lineScore != 0)  return lineScore;
            lineScore = scoreLine(squares[0][i],
                                  squares[1][i],
                                  squares[2][i]);
            if (lineScore != 0)  return lineScore;
         }
         lineScore = scoreLine(squares[0][0],
                               squares[1][1],
                               squares[2][2]);
         if (lineScore != 0)  return lineScore;
         return scoreLine(squares[0][2], squares[1][1], squares[2][0]);
      }

      /* Return 1 if 3 chars are all 'X', -1 if all 'O', 0 otherwise */
      protected int scoreLine(char a, char b, char c) {
         if ((a == 'X') && (b == 'X') && (c == 'X'))  return  1;
         if ((a == 'O') && (b == 'O') && (c == 'O'))  return -1;
         return 0;
      }

      public String toString() {
         String result = "";
         for (int row = 0; row < 3; row++) {
            for (int column = 0; column < 3; column++) {
               result += squares[row][column];
            }
            result += "\n";
         }
         return result;
      }

      public static void main(String[] args) {
         TicTacToe game = new TicTacToe();
         System.out.println("Welcome to Tic Tac Toe.\n");
         game.play();
         System.out.println(game);
         System.out.println("Game over.");
      }

   }
```

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) `calcScore()`
b) `minimaxForO()`
c) `minimaxForX()`
d) `play()`
e) `playBestMove()`

Answer group for B

a) `minimaxForO()`
b) `minimaxForX()`
c) `play()`
d) `squares[row][column] = 'O'`
e) `squares[row][column] = 'X'`

Answer group for D and E

a) `player`
b) `squares[bestRow][bestColumn]`
c) `squares[bestrow][column]`
d) `squares[row][bestcolumn]`
e) `squares[row][column]`

Subquestion 2

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

Assuming that method `playBestMove` is called when the current status of the board is as shown in (3a) in Figure 3. After method `playBestMove` tests all 7 empty cells, the score values shown in (3b) in Figure 3 are obtained. Based on this result, method `playBestMove` puts X into the cell at [F], which holds the best score value.

X		
		O

(3a) current status

	column 0	1	2
row 0		-1	1
1	-1	0	0
2	1	0	

(3b) score values

Figure 3 Current status (3a) and score values (3b)

Answer group

a) row 0, column 1
b) row 0, column 2
c) row 1, column 0
d) row 2, column 0