



<< Sample Questions >>

Fundamental IT Engineer Examination (Subject B)

Question Nos.	Q1 – Q20
Question Selection	Compulsory
Examination Time	100 minutes

Pseudo programming language notations

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

[Pseudo programming language notations]

Notation	Description
\circ <i>procedure</i> (<i>type</i> : <i>arg1</i> , ...)	Declares a <i>procedure</i> and its argument(s) <i>arg1</i> ,
\circ <i>ret-type</i> : <i>function</i> (<i>type</i> : <i>arg1</i> , ...)	Declares a <i>function</i> , its argument(s) <i>arg1</i> , ... , and type of return value <i>ret-type</i> .
<i>type</i> : <i>var1</i> , ... <i>type</i> []: <i>array1</i> , ...	Declares variables <i>var1</i> , ... and arrays <i>array1</i> , ... by data <i>type</i> such as integer, real, and string.
<i>/* comment */</i>	Describes a comment between <i>/*</i> and <i>*/</i> .
<i>// comment</i>	Describes a comment after <i>//</i> till end of line.
<i>variable</i> ← <i>expression</i>	Assigns the value of the <i>expression</i> to the <i>variable</i> .
<i>procedure</i> (<i>arg1</i> , ...)	Calls a <i>procedure</i> by passing arguments <i>arg1</i> ,
<i>function</i> (<i>arg1</i> , ...)	Calls a <i>function</i> by passing arguments <i>arg1</i> , ... , and receiving the return value.
output <i>arg1</i> , ...	Outputs values of <i>arg1</i> , ... to a printable device.
return <i>ret-val</i>	Finishes a function by passing return value <i>ret-val</i> .
<pre> if (<i>condition-i</i>) } *1 <i>process-i</i> elseif (<i>condition-ei</i>) } *2 <i>process-ei</i> else } *3 <i>process-e</i> endif </pre>	<p>Indicates the selection process.</p> <p>*1 If <i>condition-i</i> is true, then execute <i>process-i</i>. Otherwise, proceed to the next elseif or else.</p> <p>*2 If <i>condition-ei</i> is true, then execute <i>process-ei</i>. Otherwise, proceed to the next elseif or else.</p> <p>*3 If all conditions are false, execute <i>process-e</i>.</p> <p>Note: *2 and *3 can be omitted. *2 may exist twice or more.</p>
<pre> for (<i>sequence</i>) <i>process</i> endfor </pre>	<p>Indicates the “for” iteration process.</p> <p>In the order specified in the <i>sequence</i>, execute the <i>process</i> repeatedly.</p>
<pre> while (<i>condition</i>) <i>process</i> endwhile </pre>	<p>Indicates the “while” iteration process.</p> <p>While the <i>condition</i> is true, execute the <i>process</i> repeatedly.</p>
<pre> do <i>process</i> while (<i>condition</i>) </pre>	<p>Indicates the “do-while” iteration process.</p> <p>Execute the <i>process</i> once, and then while the <i>condition</i> is true, execute the <i>process</i> repeatedly.</p>

Pseudo programming language notations
(continued)

[Operators and their precedence]

Type of operator	Operators	Precedence	Note
Expression	(), . ⁽¹⁾	High ↑ ↓ Low	(1) accessing member or method
Unary operator	+, -, not ⁽²⁾		(2) logical negation
Binary operator	x, ÷, mod ⁽³⁾		(3) remainder
	+, -		
	>, <, ≥, ≤, =, ≠		
	and ⁽⁴⁾		(4) logical product
	or ⁽⁵⁾	(5) logical sum	

[Boolean-type constants]

true, false

[Array reference]

	1-dimensional array	2-dimensional array	Array of arrays																													
Array declaration	<i>type</i> []: <i>name</i> ...	<i>type</i> [,]: <i>name</i> ...	<i>type</i> [] []: <i>name</i> ...																													
Example	integer []: a1 1 2 3 4 5 <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>1</td><td>3</td><td>5</td><td>7</td><td>9</td></tr> </table>	1	3	5	7	9	integer [,]: a2 1 2 3 <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>1</td><td>11</td><td>12</td><td>13</td></tr> <tr><td>2</td><td>14</td><td>15</td><td>16</td></tr> <tr><td>3</td><td>17</td><td>18</td><td>19</td></tr> </table>	1	11	12	13	2	14	15	16	3	17	18	19	integer [] []: aa 1 2 3 <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>1</td><td>21</td><td>22</td><td></td></tr> <tr><td>2</td><td>23</td><td>24</td><td>25</td></tr> <tr><td>3</td><td>26</td><td></td><td></td></tr> </table>	1	21	22		2	23	24	25	3	26		
1	3	5	7	9																												
1	11	12	13																													
2	14	15	16																													
3	17	18	19																													
1	21	22																														
2	23	24	25																													
3	26																															
Data reference	Data 7 is referred to by a1[4]	Data 16 is referred to by a2[2,3]	Data 25 is referred to by aa[2][3]																													
Notation of array contents	{1, 3, 5, 7, 9}	{{11, 12, 13}, {14, 15, 16}, {17, 18, 19}}	{{21, 22}, {23, 24, 25}, {26}}																													

Note: The indexes of example arrays start at 1.

[undefined state]

undefined is a state in which no value is set to a variable (or an element of an array).
By setting undefined to a variable, the variable is transformed into undefined state.

Q1. From the answer group below, select the correct answer to be inserted in in the description.

When the program is executed, the output is "".

[Program]

```
integer: x ← 1
```

```
integer: y ← 2
```

```
integer: z ← 3
```

```
x ← y
```

```
y ← z
```

```
z ← x
```

```
output y, z // the values are separated by ", "
```

Answer group

a) 1, 2

b) 1, 3

c) 2, 1

d) 2, 3

e) 3, 1

f) 3, 2

Q2. From the answer group below, select the correct combination of the answers to be inserted in through in the program.

The function `fizzBuzz` receives a value that is given as an argument, and returns “Divisible by 3” if the value is divisible by 3 but is not divisible by 5, “Divisible by 5” if the value is divisible by 5 but is not divisible by 3, and “Divisible by 3 and 5” if the value is divisible by 3 and 5. Otherwise, it returns “Not divisible by 3 or 5”.

[Program]

```
O string: fizzBuzz(integer: num)
  string: result
  if (num is divisible by )
    result ← "Divisible by  "
  elseif (num is divisible by )
    result ← "Divisible by  "
  elseif (num is divisible by )
    result ← "Divisible by  "
  else
    result ← "Not divisible by 3 or 5"
  endif
  return result
```

Answer group

	A	B	C
a)	3	3 and 5	5
b)	3	5	3 and 5
c)	3 and 5	3	5
d)	5	3	3 and 5
e)	5	3 and 5	3

Q3. From the answer group below, select the correct answer to be inserted in in the description. Here, the array indexes start at 1.

The function `makeNewArray` receives an integer array with at least two elements as an argument, and returns an integer array. When the function `makeNewArray` is called as `makeNewArray({3, 2, 1, 6, 5, 4})`, the value at element number 5 of the array that is returned as the return value is .

[Program]

```
○ integer []: makeNewArray(integer []: in)
  integer []: out ← {} // An array with 0 elements
  integer: i, tail
  add the value of in[1] to the end of out
  for (increase i from 2 to the number of elements in in by 1)
    tail ← out[the number of elements in out]
    add the value of (tail + in[i]) to the end of out
  endfor
  return out
```

Answer group:

- | | | | |
|-------|-------|-------|-------|
| a) 5 | b) 6 | c) 9 | d) 11 |
| e) 12 | f) 17 | g) 21 | |

Q4. From the answer group below, select the correct combination of the answers to be inserted in through in the program.

The function gcd uses the properties (1) through (3) below to calculate the greatest common divisor for the two positive integers num1 and num2 that are given as arguments.

- (1) When num1 and num2 are equal, the greatest common divisor for num1 and num2 is num1.
- (2) When num1 is greater than num2, the greatest common divisor for num1 and num2 is equal to the greatest common divisor for (num1 - num2) and num2.
- (3) When num2 is greater than num1, the greatest common divisor for num1 and num2 is equal to the greatest common divisor for (num2 - num1) and num1.

[Program]

```

integer: gcd(integer: num1, integer: num2)
integer: x ← num1
integer: y ← num2

  if ()
    x ← x - y
  else
    y ← y - x
  endif

return x

```

Answer group

	A	B	C
a)	if (x ≠ y)	x < y	endif
b)	if (x ≠ y)	x > y	endif
c)	while (x ≠ y)	x < y	endwhile
d)	while (x ≠ y)	x > y	endwhile

Q5. From the answer group below, select the correct answer to be inserted in in the program.

The function `calc` receives the positive real numbers `x` and `y`, and returns the result of the calculation of $\sqrt{x^2 + y^2}$. The function `pow` that the function `calc` uses receives the positive real number `a` as its first argument and the real number `b` as its second argument, and returns the real type value of a^b (`a` to the power of `b`).

[Program]

```
○ real: calc(real: x, real: y)
  return 
```

Answer group

- a) $(\text{pow}(x, 2) + \text{pow}(y, 2)) \div \text{pow}(2, 0.5)$
- b) $(\text{pow}(x, 2) + \text{pow}(y, 2)) \div \text{pow}(x, y)$
- c) $\text{pow}(2, \text{pow}(x, 0.5)) + \text{pow}(2, \text{pow}(y, 0.5))$
- d) $\text{pow}(\text{pow}(\text{pow}(2, x), y), 0.5)$
- e) $\text{pow}(\text{pow}(x, 2) + \text{pow}(y, 2), 0.5)$
- f) $\text{pow}(x, 2) \times \text{pow}(y, 2) \div \text{pow}(x, y)$
- g) $\text{pow}(x, y) \div \text{pow}(2, 0.5)$

Q6. From the answer group below, select the correct answer to be inserted in in the program.

The function `rev` receives the 8-bit type argument `byte`, and returns these bits in reverse order. For example, when the function `rev` is called as `rev(01001011)`, the return value is `11010010`.

Here, the operator `&` represents a bitwise logical product, the operator `|` represents a bitwise logical sum, the operator `>>` represents a logical shift to the right, and the operator `<<` represents a logical shift to the left. For example, `v >> n` performs a logical shift of the value of `v` by `n` bits to the right, and `v << n` performs a logical shift of the value of `v` by `n` bits to the left.

[Program]

```
○ 8-bit: rev(8-bit: byte)
  8-bit: rbyte ← byte
  8-bit: r ← 00000000
  integer: i
  for (increase i from 1 to 8 by 1)
    
  endfor
  return r
```

Answer group

- a) `r ← (r << 1) | (rbyte & 00000001)`
`rbyte ← rbyte >> 1`
- b) `r ← (r << 7) | (rbyte & 00000001)`
`rbyte ← rbyte >> 7`
- c) `r ← (rbyte << 1) | (rbyte >> 7)`
`rbyte ← r`
- d) `r ← (rbyte >> 1) | (rbyte << 7)`
`rbyte ← r`

Q7. From the answer group below, select the correct answer to be inserted in in the program.

The function `factorial` receives the non-negative integer `n` as an argument, and returns its factorial. The factorial of the non-negative integer `n` is 1 when `n` is 0, and in all other cases, it is the number obtained by multiplying all integers from 1 to `n`.

[Program]

```
○ integer: factorial(integer: n)
  if (n = 0)
    return 1
  endif
  return 
```

Answer group

- | | |
|---|---------------------------------------|
| a) $(n - 1) \times \text{factorial}(n)$ | b) $\text{factorial}(n - 1)$ |
| c) n | d) $n \times (n - 1)$ |
| e) $n \times \text{factorial}(1)$ | f) $n \times \text{factorial}(n - 1)$ |

Q8. From the answer group below, select the correct answer to be inserted in in the description.

The program performs operations on a priority queue.

A priority queue is a queue where the handled elements have a priority assigned to them, and when elements are extracted, they are extracted in the order of highest priority first.

The class PrioQueue represents a priority queue. The Figure shows an explanation of the class PrioQueue. Here, the priority is the integer value 1, 2, or 3, and the smaller the value the higher the priority.

When the procedure prioSched is called, the order of the output is .

Constructor	Description
PrioQueue()	Creates an empty priority queue.

Method	Type of return value	Description
enqueue(string: s, integer: prio)	None	Adds the string s as an element to a priority queue with the priority prio.
dequeue()	string	Extracts the element with the highest priority in a priority queue and returns it. If there are multiple elements with the highest priority, it extracts the element that was added first and returns it.
size()	integer	Returns the number of elements that are stored in a priority queue.

Figure Explanation of the class PrioQueue

[Program]

```
○ prioSched()  
  PrioQueue: prioQueue ← PrioQueue()  
  prioQueue.enqueue("A", 1)  
  prioQueue.enqueue("B", 2)  
  prioQueue.enqueue("C", 2)  
  prioQueue.enqueue("D", 3)  
  prioQueue.dequeue() /* Does not use a return value */  
  prioQueue.dequeue() /* Does not use a return value */  
  prioQueue.enqueue("D", 3)  
  prioQueue.enqueue("B", 2)  
  prioQueue.dequeue() /* Does not use a return value */  
  prioQueue.dequeue() /* Does not use a return value */  
  prioQueue.enqueue("C", 2)  
  prioQueue.enqueue("A", 1)  
  while (prioQueue.size() is not equal to 0)  
    output the return value of prioQueue.dequeue()  
  endwhile
```

Answer group

- a) "A", "B", "C", "D"
- b) "A", "B", "D", "D"
- c) "A", "C", "C", "D"
- d) "A", "C", "D", "D"

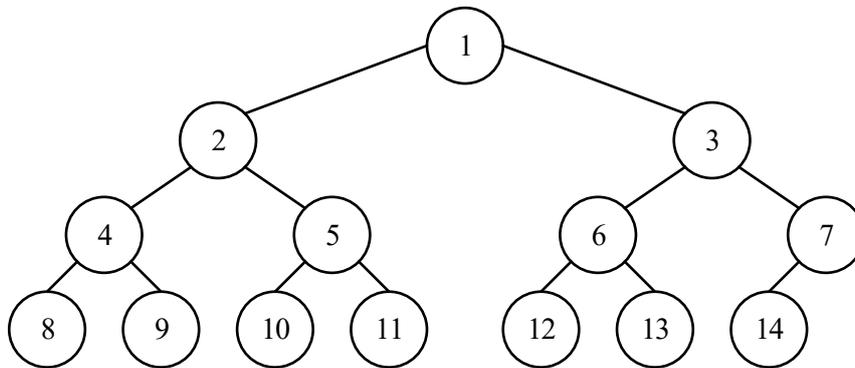
Q9. From the answer group below, select the correct answer to be inserted in in the description. Here, the array index starts at 1.

The procedure order traces through a subtree of the binary tree in the Figure, and outputs all node numbers in the subtree. The node number of the root node of the subtree is specified with the argument n.

The global array tree represents the binary tree in the Figure. Each element of the array tree is an array that stores the node numbers of the corresponding child nodes in the order of left child then right child.

For example, the element at element number 1 in the array tree is an array that contains the node numbers of the children of node number 1, and it stores node number 2 of the left child and node number 3 of the right child as the array {2, 3}.

When the procedure order is called as order(1), the output is in the order .



Note 1: A number in a ○ is a node number.

Note 2: If there is only one child node, it is handled as a left child node.

Figure Binary tree that is handled by the program

[Program]

```
global: integer [][]: tree ← {{2, 3}, {4, 5}, {6, 7}, {8, 9},  
                             {10, 11}, {12, 13}, {14}, {}, {}, {},  
                             {}, {}, {}, {}}  
// {} is an array with 0 elements
```

```
○ order(integer: n)
```

```
  if (the number of elements in tree[n] is equal to 2)
```

```
    order(tree[n][1])
```

```
    output n
```

```
    order(tree[n][2])
```

```
  elseif (the number of elements in tree[n] is equal to 1)
```

```
    order(tree[n][1])
```

```
    output n
```

```
  else
```

```
    output n
```

```
  endif
```

Answer group

- a) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
- b) 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14
- c) 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 3, 14, 7
- d) 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 7, 3, 1

Q10. From the answer group below, select the correct answer to be inserted in in the program.

The procedure `delNode` deletes an element from a singly-linked list at the position specified with the argument `pos`. The argument `pos` is a positive integer that is equal to or less than the number of elements in the list. The position at the top of the list is 1.

The class `ListElement` represents an element in a singly-linked list. The table provides an explanation of the member variables of the class `ListElement`. `ListElement`-type variables store references to instances of the class `ListElement`. A reference to the first element in the list is pre-stored in the global variable `listHead`.

Table Explanation of the member variables of the class `ListElement`

Member variable	Type	Description
<code>val</code>	character	The value of an element.
<code>next</code>	<code>ListElement</code>	A reference for the next element. If there is no next element, the status is undefined.

[Program]

```

global: ListElement: listHead // stores the first element in the list

O delNode(integer: pos)      /* pos is a positive integer that is equal to or
                               less than the number of elements in the list */

  ListElement: prev
  integer: i
  if (pos is equal to 1)
    listHead ← listHead.next
  else
    prev ← listHead
    /* if pos is equal to 2, the following iteration process is not executed */
    for (increase i from 2 to pos - 1 by 1)
      prev ← prev.next
    endfor
    prev.next ← 
  endif

```

Answer group:

- a) `listHead` b) `listHead.next` c) `listHead.next.next`
- d) `prev` e) `prev.next` f) `prev.next.next`

Q11. From the answer group below, select the correct answer to be inserted in in the description. Here, the array indexes starts at 1.

When the function `binSort` is called as `binSort()`, the values are sorted in ascending order with no undefined elements included in the array of the return value.

[Program]

```
integer []: binSort(integer []: data)
  integer: n ← the number of elements in data
  integer []: bins ← {n undefined values}
  integer: i

  for (increase i from 1 to n by 1)
    bins[data[i]] ← data[i]
  endfor

  return bins
```

Answer group

- | | |
|-----------------------|-----------------------|
| a) {2, 6, 3, 1, 4, 5} | b) {3, 1, 4, 4, 5, 2} |
| c) {4, 2, 1, 5, 6, 2} | d) {5, 3, 4, 3, 2, 6} |

Q12. From the answer group below, select the correct answer to be inserted in in the program. Here, the array indexes start at 1.

The function `simRatio` compares the two character arrays `s1` and `s2` that are given as arguments. `s1` and `s2` have one or more elements.

If the number of elements in each array is equal, it returns the following real value as an index of how similar the arrangement of the content in the arrays is:

$$\frac{\text{(number of pairs of elements where the characters at the same element number match)}}{\text{the number of elements in s1}}$$

For example, the return value is 1 when all of the elements in the arrays match, and the return value is 0 when there are no matching elements.

If the number of elements in each array is not equal, the function returns -1.

The table shows examples of `s1` and `s2` given to the function `simRatio` and the return values. In the program, areas outside of the arrays must not be referenced.

Table Examples of `s1` and `s2` given to the function `simRatio` and the return values

s1	s2	Return value
{"a", "p", "p", "l", "e"}	{"a", "p", "p", "l", "e"}	1
{"a", "p", "p", "l", "e"}	{"a", "p", "r", "i", "l"}	0.4
{"a", "p", "p", "l", "e"}	{"m", "e", "l", "o", "n"}	0
{"a", "p", "p", "l", "e"}	{"p", "e", "n"}	-1

[Program]

```

O real: simRatio(character []: s1, character []: s2)
  integer: i, cnt ← 0
  if (the number of elements in s1 ≠ the number of elements in s2)
    return -1
  endif
  for (increase i from 1 to the number of elements in s1 by 1)
    if ()
      cnt ← cnt + 1
    endif
  endfor
  return cnt ÷ the number of elements in s1 /* calculate as real numbers */

```

Answer group

- | | |
|---------------------------------|-------------------------------|
| a) <code>s1[i] ≠ s2[cnt]</code> | b) <code>s1[i] ≠ s2[i]</code> |
| c) <code>s1[i] = s2[cnt]</code> | d) <code>s1[i] = s2[i]</code> |

Q13. From the answer group below, select the correct answer to be inserted in in the description. Here, the array index starts at 1.

The function search receives the array specified with the argument data, and if it contains the value specified with the argument target, returns the relevant element number. If the array does not contain the value specified with target, it returns -1. data is sorted in ascending order, and there are no duplicate values.

The function search has a defect. For example, if , the function enters an endless loop.

[Program]

```
integer: search(integer []: data, integer: target)
integer: low, high, middle

low ← 1
high ← the number of elements in data

while (low ≤ high)
  middle ← integer part of ((low + high) ÷ 2)
  if (data[middle] < target)
    low ← middle
  elseif (data[middle] > target)
    high ← middle
  else
    return middle
  endif
endwhile

return -1
```

Answer group

- a) one or more elements in data contain the value -1
- b) the number of elements in data is 1 and target is equal to the value of the element
- c) the number of elements in data is 2 and target is equal to the value of the first element of data
- d) the number of elements in data is 2 and target is equal to the value of the last element of data

Q14. From the answer group below, select the correct answer to be inserted in in the description. Here, the array indexes start at 1.

The function `summarize` receives the array `sortedData` with at least one element and that is sorted in ascending order, and returns five values that characterize the array.

When the function `summarize` is called as `summarize({0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1})`, the return value is .

[Program]

```
○ real: findRank(real []: sortedData, real: p)
  integer: i
  i ← round up (p × (the number of elements in sortedData - 1))
  // round up the value to the nearest whole number, e.g. round up (3.75) is 4.
  return sortedData[i + 1]

○ real []: summarize(real []: sortedData)
  real []: rankData ← {} /* array with 0 elements */
  real []: p ← {0, 0.25, 0.5, 0.75, 1}
  integer: i
  for (increase i from 1 to the number of elements in p by 1)
    add the return value of findRank(sortedData, p[i]) to the end of rankData
  endfor
  return rankData
```

Answer group

- a) {0.1, 0.3, 0.5, 0.7, 1}
- b) {0.1, 0.3, 0.5, 0.8, 1}
- c) {0.1, 0.3, 0.6, 0.7, 1}
- d) {0.1, 0.3, 0.6, 0.8, 1}
- e) {0.1, 0.4, 0.5, 0.7, 1}
- f) {0.1, 0.4, 0.5, 0.8, 1}
- g) {0.1, 0.4, 0.6, 0.7, 1}
- h) {0.1, 0.4, 0.6, 0.8, 1}

Q15. From the answer group below, select the correct combination of the answers to be inserted in and in the description.

In tic-tac-toe, the moves that give a player the highest chance of winning are to be determined. In the following procedure, the state transition of a game is expressed with a tree structure, and the evaluated value of each node other than the root is calculated. As a result, the move with the highest evaluated value among the child nodes of the root is deemed to be the move with the highest chance of winning. The moves that the player chooses are represented with \circ , and the moves that the opponent chooses are represented with \times .

[Procedure]

- (1) The current state of the board is taken as the root, and all conceivable moves up to a win, loss, or draw are expressed by using a tree structure.
- (2) The state of the leaf (bottom node) is evaluated as below.
 - (a) If the player wins: 10
 - (b) If the player loses: -10
 - (c) If the game is a draw: 0
- (3) The evaluated value of nodes other than leaves is determined based on the evaluated values of all child nodes of the relevant node.
 - (a) For a node on the player's turn, the evaluated value of the node is the maximum evaluated value among the child nodes.
 - (b) For a node on the opponent's turn, the evaluated value of the node is the minimum evaluated value among the child nodes.

When the game is in a state with the root as shown at the top of the Figure, there are three moves that the player can choose from. Among these, the evaluated value of the child indicated by A is , and the evaluated value of the child indicated by B is .

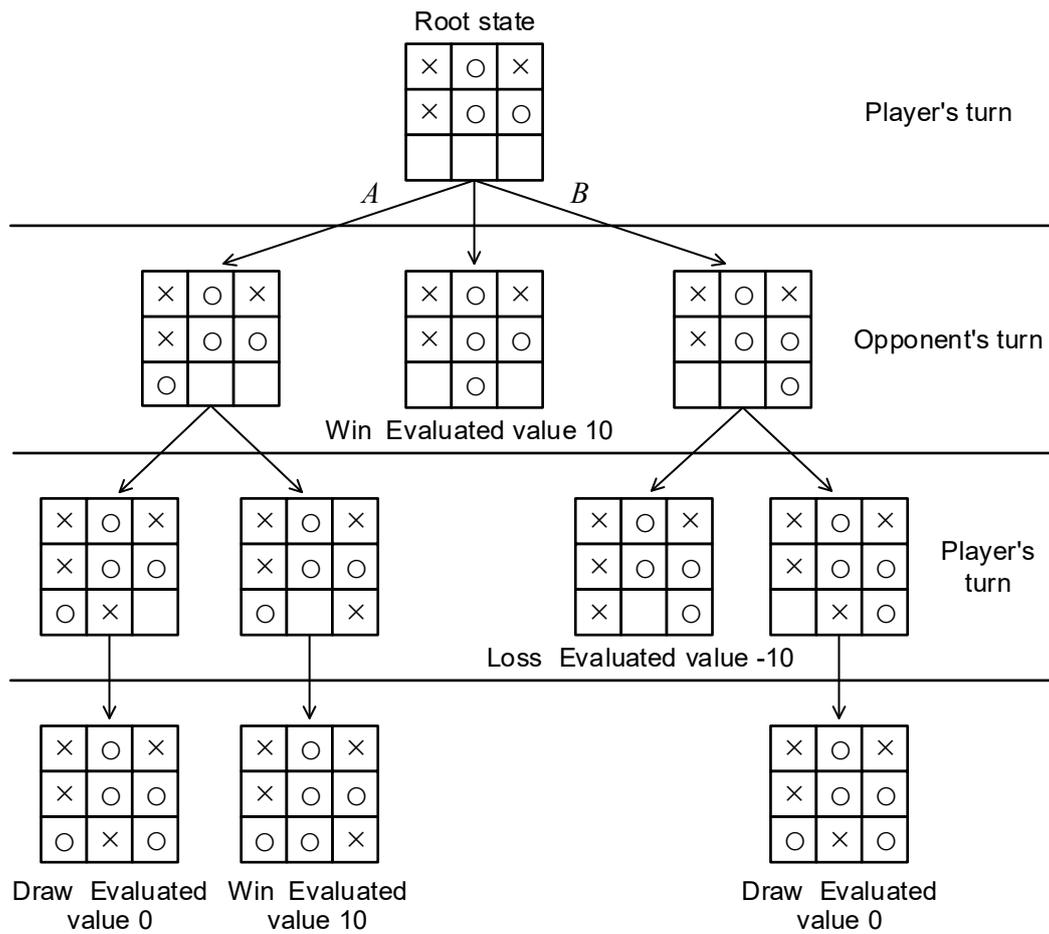


Figure State transition for tic-tac-toe

Answer group

	A	B
a)	0	-10
b)	0	0
c)	10	-10
d)	10	0

Q16. From the answer group below, select the correct answer to be inserted in in the program. The same answer goes into both blanks . Here, the array index starts at 1.

The program converts the code point of Unicode to UTF-8 encoding.

In this question, “(16)” after a numerical value indicates that it is a hexadecimal value.

Each Unicode character is given an integer value that is called a code point. UTF-8 is one of the encoding methods for Unicode characters, and a character with a code point that ranges between 800(16) and FFFF(16) is encoded to a 3-byte value as below.

Let the bit pattern with a 3-byte length be $1110\underline{\text{x}}\text{x}\text{x}\text{x}$ $10\underline{\text{x}}\text{x}\text{x}\text{x}\text{x}\text{x}$ $10\underline{\text{x}}\text{x}\text{x}\text{x}\text{x}\text{x}$. The underlined 16 “x” positions in the bit pattern store the 16-bit code point. The code point is justified to the right, and 0 is stored in any leftover “x” positions. This 3-byte value is UTF-8 encoded. For example, when the code point 266B(16) for Beamed Eighth Notes character “” is represented in binary, it is 10011001101011. When this is stored right-justified in the “x” positions in the bit pattern above, it is 1110xx10 10011001 10101011. When 0 is stored in the two leftover “x” positions, the UTF-8 encoding for Beamed Eighth Notes character “” 11100010 10011001 10101011 is obtained.

The function encode converts a Unicode code point that is passed as an argument to UTF-8 encoding, and returns an integer array that stores it one byte per element from the start of the array. It is assumed that only an integer value that ranges between 800(16) and FFFF(16) is passed to encode as an argument.

[Program]

```

○ integer []: encode(integer: codePoint)
  /* the initial value of utf8Bytes is the value when “x”s in the bit pattern are re-
     placed with 0, divided into three 8-bit blocks, and each deemed to be binary */
  integer []: utf8Bytes ← {224, 128, 128}
  integer: cp ← codePoint
  integer: i
  for (decrease i from the number of elements in utf8Bytes to 1 by 1)
    utf8Bytes[i] ← utf8Bytes[i] + (cp ÷ remainder of )
    cp ← cp ÷ quotient of 
  endfor
  return utf8Bytes

```

Answer group

a) $((4 - i) \times 2)$

c) (2 to the power of i)

e) 2

g) 16

i) 256

b) (2 to the power of $(4 - i)$)

d) $(i \times 2)$

f) 6

h) 64

Q17. From the answer group below, select the appropriate combination of the answer to be inserted in , , and in Figure 1.

Company X, a manufacturer, sells its products to consumers on Company X’s EC site (hereinafter, the site). On the site, the consumers can search Company X’s products and services. The site provides log-in function, and manages member information of registered consumers such as names and mail addresses.

The site operates on Company Y’s PaaS. It uses DBMS and application servers on the PaaS. Company X outsources the development and operation of the site to Company Z. The outsourcing contract between Company X and Company Z states that Company Z is responsible to control and fix vulnerabilities in Web application programs.

Recently, a security incident occurred on a Web site operated by another company in the same profession. A Web vulnerability was exploited and that resulted in leakage of personal information. Accordingly, Company X conducted vulnerability assessment of the site and three findings that need security actions were pointed out. Figure 1 shows the findings and responsible companies that should take corrective actions.

No	Finding	Action by
1	Attacks that exploit a vulnerability may occur due to a known vulnerability in the OS of the application server used in the site.	Company <input type="text" value="A"/>
2	Member information may be leaked due to a vulnerability of cross-site scripting in the site.	Company <input type="text" value="B"/>
3	Attacks that exploit a vulnerability may occur due to a known vulnerability in the DBMS used in the site.	Company <input type="text" value="C"/>

Figure 1 Findings and responsible companies that should take corrective actions

Answer group

	A	B	C
a)	X	X	X
b)	X	X	Z
c)	X	Y	Y
d)	Y	Y	Y
e)	Y	Y	Z
f)	Y	Z	Y
g)	Y	Z	Z
h)	Z	Y	Y
i)	Z	Y	Z
j)	Z	Z	Y

Q18. From the answer group below, select the most appropriate answer to be inserted in in the description.

Company A is a company that undertakes IT development and has 1,000 employees. The General Affairs Department has 50 employees, the Sales Department has 50 employees, and all other employees are in the Development Department. 90% of Development Department employees are stationed at customers' offices. The current status of the use of PCs at Company A is as shown in Figure 1.

1. Company A's PCs
 - o Company A prepares and lends one PC (hereinafter, a Company PC) to each employee in the General Affairs Department, each employee in the Sales Department, and each employee in the Development Department who works at Company A's office.
 - o Company A does not lend Company PCs to Development Department employees who are stationed at customers' offices, but instead, it prepares shared PCs for them to use when they come to work at Company A's office.
2. Use of the business system by Development Department employees who are stationed at customers' offices
 - o Development Department employees who are stationed at customers' offices need to use a shared PC in order to use the business system for labor claiming and expense claiming.
3. Use of Company A's VPN
 - o A VPN server is installed at Company A, and Sales Department employees can use the business system when they are on business trips by establishing a VPN connection to the internal network from a Company PC via the Internet. The rules state that a Company PC is to be used for VPN connections.

Figure 1 Current status of the use of PCs at Company A

Company A considers that it is inefficient for Development Department employees who are stationed at customers' offices to come to Company A's office only for the purpose of using the business system, and so decided to consider permitting them to use their private PC for business use (BYOD) and for connecting to the VPN.

Assuming that Development Department employees who are stationed at customers' offices are permitted to connect to the VPN from their private PC.

Among the risk descriptions (1) to (5) below, the two risks that will increase or appear for the first time are .

- (1) The risk of lowering the availability because of an increase in VPN connections.
- (2) The risk that a Development Department employee stationed at a customer's office loses a Company PC.
- (3) The risk that a Development Department employee stationed at a customer's office clicks on a URL in a phishing e-mail and infects his/her private PC with malware.
- (4) The risk that a General Affairs Department employee establishes a VPN connection from his/her private PC.
- (5) The risk that a private PC with a malware infection establishes a VPN connection to the internal network, and the malware will spread on the internal network.

Answer group

- | | | | |
|----------------|----------------|----------------|----------------|
| a) (1) and (2) | b) (1) and (3) | c) (1) and (4) | d) (1) and (5) |
| e) (2) and (3) | f) (2) and (4) | g) (2) and (5) | h) (3) and (4) |
| i) (3) and (5) | j) (4) and (5) | | |

Q19. From the answer group below, select the appropriate combination of the terms or phrases to be inserted in

A

 and

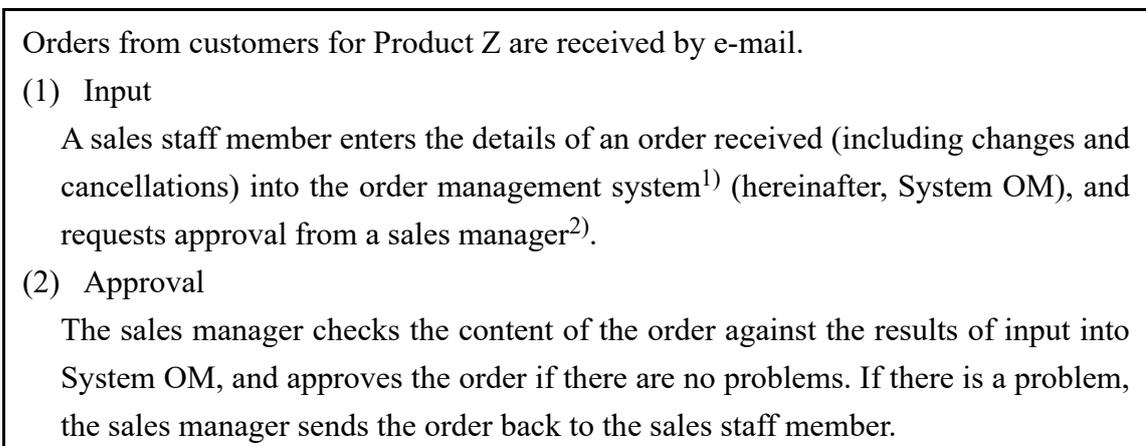
B

 in Table 1.

Company X is a mail-order sales company with 200 employees. It sells general lifestyle goods, gift products, and other such products to general consumers. One of the products that it offers is Product Z, and Sales Section Z is responsible for selling this product.

[The work of Sales Section Z]

Currently, the staff members of Sales Section Z perform order management work and handle inquiries concerning Product Z. Figure 1 shows the procedure for order management work.



Note ¹⁾ The system is operated by the Information Systems Department in Company X. The users include sales managers and sales staff members.

Note ²⁾ Approval is required only from the manager of Sales Section Z.

Figure 1 Procedure for order management work

[System OM operation permissions]

Regarding System OM, Sales Section Z has defined the usage policies below.

[Policy 1] Information that is entered by a certain user is approved by another user.

[Policy 2] Sales managers can view information for all work in Sales Section Z.

In System OM, a function to grant the operation permissions based on business needs to users is implemented.

Because of an increase in orders, order management work for Product Z is increasing. Therefore, Company X decides to outsource a part of the work for Product Z to Company Y. Specifically, sales staff members in Company Y perform input operations for order management work. A sales manager in Company Y views the input results of the sales staff members in Company Y and verbally requests Company X to send them back if they are incomplete.

For this outsourcing, Sales Section Z conveys the requirements below concerning System OM to the Information Systems Department.

[Requirement 1] If input is performed by Company Y, it is to be approved by Company X.

[Requirement 2] If input is performed by sales staff members in Company X, it is to be approved by Company X's sales manager as usual.

On the basis of the above, the Information Systems Department compiles the operation permissions to be granted to each user in Table 1.

Table 1 Operation permissions to be granted to each user

User	Operation permissions to be granted	System OM		
		View	Input	Approve
	(Omitted)	○		○
	Sales staff members in Sales Section Z	(Omitted)	(Omitted)	(Omitted)
	A	○		
	B	○	○	

Note: ○ indicates that an operation permission is granted.

Answer group

	A	B
a)	Sales manager in Company Y	Sales manager in Sales Section Z
b)	Sales manager in Company Y	Sales staff members in Company Y
c)	Sales manager in Sales Section Z	Sales manager in Company Y
d)	Sales manager in Sales Section Z	Sales staff members in Company Y
e)	Sales staff members in Company Y	Sales manager in Company Y

Q20. From the answer group below, select the most appropriate answer to be inserted in in the description.

Company A offers nutritional supplements and has 500 employees. Company A's information systems, including its servers and firewall (hereinafter, FW), are run by the Information Systems Department.

One day, there was an audit from the Internal Audit Department, and when the manager of the Information Systems Department provided the explanation in Figure 1, he received the findings in Table 1.

- The information systems, including the FW, are run by six operation staff members who belong to the Operations Team in the Information Systems Department.
- There are three aspects to FW operation: editing FW rules, checking operation logs, and checking FW rules after editing and approving operations (hereinafter, checking FW rules after editing and approving operations is referred to as operation approval).
- Editing of FW rules is performed in line with operation instructions that are created in advance.
- There are three FW functions: FW rule editing, operation log checking, and operation approval.
- In order to change FW rules, both FW rule editing and operation approval are required. Prior to operation approval, operation log checking is performed.
- User IDs for the FW are issued to individual operation staff members, and user IDs are not shared.
- For the FW, permission to use a function can be granted for each operation staff member's user ID.
- Currently, each of the six operation staff members is granted all of the permissions, and the same operation staff member that edited FW rules then checks that there are no errors in the operation, and approves the operation.
- A password is used for logging in to the FW. Passwords are composed of eight alphabetical and numeric letters.
- In FW operation, whether login can be performed from the console without going via the network, and whether login can be performed remotely via the network, are set for each operation staff member's user ID.
- The same FW functions can be used regardless of whether login is performed from the console without going via the network or whether login is performed remotely via the network.
- The FW is installed in the server room. In the server room, there are also several other types of servers installed.
- Only operation staff members are permitted to enter the server room.

Figure 1 FW operation status

Table 1 Findings from the Internal Audit Department

Finding	Finding Details
Finding 1	In FW operation work, duties are not properly separated.
Finding 2	(Omitted)
Finding 3	(Omitted)
Finding 4	(Omitted)

The manager of the Information Systems Department investigates action plans for the findings in Table 1.

The following is the action plan that prevents erroneous changes to FW rules concerning Finding 1 in Table 1.

Action plan concerning Finding 1:

Answer group

- a) Assign the operation staff members who will perform the work for each day of the week.
- b) Divide the operation staff members into those who can log in from the console and those who can log in remotely using the FW's access control function.
- c) Divide the operation staff members into those who edit FW rules and those who check operation logs and approve operations, and only grant them the minimum permissions necessary.
- d) Install Endpoint Detection and Response (EDR) in the console in order to enhance monitoring.
- e) Limit the number of operation staff members for the FW to one.
- f) Make some of the operation staff members' role only to check operation logs, and only grant these members the permission to check operation logs.
- g) Use multi-factor authentication instead of password authentication for login to the FW by operation staff members.

Correct answers

Q1 f)

Q2 c)

Q3 f)

Q4 d)

Q5 e)

Q6 a)

Q7 f)

Q8 d)

Q9 c)

Q10 f)

Q11 a)

Q12 d)

Q13 d)

Q14 h)

Q15 a)

Q16 h)

Q17 f)

Q18 d)

Q19 b)

Q20 c)

First Edition: 2023-05-15

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 the text.