12-1 Algorithm

Point!

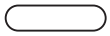
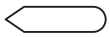
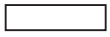
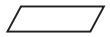
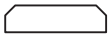
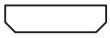
1 Algorithm

(1) ([1]Algorithm) : A method or procedure for solving a particular problem.

(2) Flowcharts and activity diagrams : Diagrams that represent algorithms in a visual and comprehensive way.

[1] ([2]Flowchart) : Method for illustrating the flow of a single process.

<List of Flowchart Symbols>

Symbol	Name	Meaning
	([3]Terminal)	Start/End
	([4]Display)	Display on a screen, etc.
	([5]Data)	Data input and output
	([6]Process)	Operations and other processes
	([7]Conditional branch)	Branching according to conditions
	([8]Repeat)	Start of a repetition
		End of a repetition
	([9]Line)	Flow of data and control

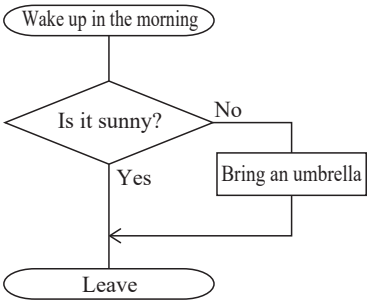
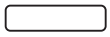
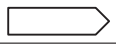


Figure 1: Process flow from waking up to leaving

[2] ([10]Activity diagram) : Suitable method for representing parallel process flows.

<List of Symbols in Activity Diagrams>

Symbol	Name
	([11]Start)
	([12]End)
	([13]Control)
	([14]Transition)
	([15]Conditional branch)
	([16]Parallel process)
	([17]Send)

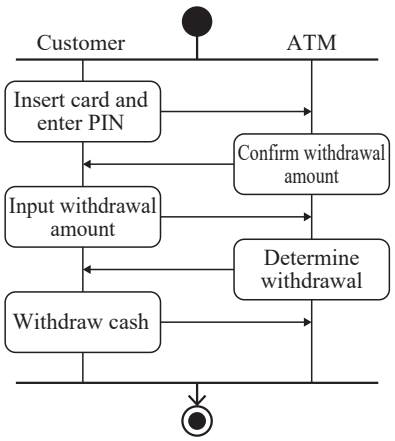
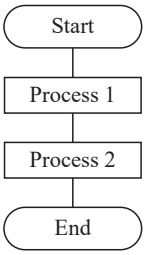
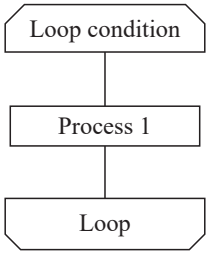
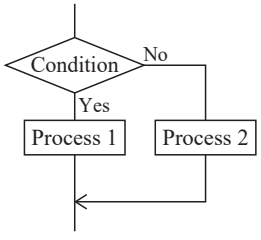


Figure 2: Cash withdrawal process at an ATM

2 Control Structure

Algorithms can essentially be represented using three control structures along with input/output.

([18]Sequential structure)	([19]Repeating structure)	([20]Branching structure)
Executes processes according to the order. 	Repeats the process while the conditions are met. 	Separates processes according to the conditions. 

3 Programming Language

(1) ([21]**Programming language**) : A language used for expressing algorithms in a way that can be understood by a computer.

- Creating a program (source code) using a programming language is referred to as ([22]**programming**).
- Programs are written using programming languages and then translated into a ([23]**machine language**) that can be understood by a computer.

(2) Examples of Programming Languages

[1] ([24]**Python**) : Language used in fields such as artificial intelligence (AI) and statistics, and can be executed with minimal coding.

[2] ([25]**JavaScript**) : Language that can be confirmed only within a web browser, making it optimal for web-related purposes.

[3] ([26]**Scratch**) : A visual programming language developed for educational purposes. Programming is possible using blocks, which is intuitive and easy to understand.

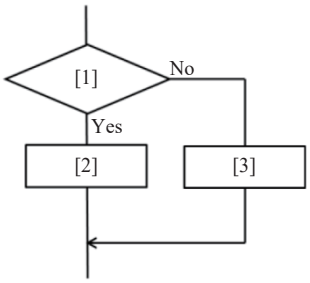
Warm Up

Answer the following questions.

(1) What is the term for describing the order of calculations or the sequence in which things are created?

(2) What is the term for a diagram that visually represents an algorithm and is ideal for showing the flow of parallel processes?

(3) The diagram on the right is a flowchart for pedestrians indicating "Proceed if the signal is green; otherwise, stop." Choose the appropriate option that best fits into the blanks [1] to [3] from the options **A** to **F** below, and answer using the letters.



A The signal is red **B** Signal is yellow
C The signal is not red **D** The signal is not green
E Possible to proceed **F** Stop

(4) What is the term for a control structure like (3)?

(5) What is the programming language that is used in fields like artificial intelligence and statistics, and can be executed with minimal description?

Explanation

(1) Algorithm
(2) Activity diagram
(3) [1] **A** [2] **F** [3] **E**
(4) Branching structure
(5) Python

Try

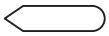
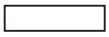
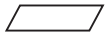
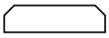
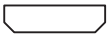
Answer the following questions.

(1) Choose the one that best fits into the blanks [1] to [4] in the following sentences from the options **A** to **F** below, and answer using the letters.

> A method or procedure for solving a particular problem is referred to as an ([1]). A language created to instruct a computer to execute an ([1]) is called ([2]), and creating a program using a ([2]) is called ([3]). In addition, a ([2]) is converted to a ([4]) that can be understood by a computer, and ultimately becomes a set of instructions composed of combinations of 0s and 1s.

A Program **B** Flowchart **C** Algorithm
D Programming **E** Programming language **F** Machine language

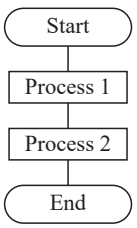
(2) The following table summarizes the symbols used in flowcharts. Choose the option that best fits into the blanks [1] to [8] from the options **A** to **H** below, and answer using the letters.

Symbol	Name	Meaning
	Terminal	([1])
	Display	([2])
	Data	([3])
	Process	([4])
	Conditional branch	([5])
	Repetition	([6])
		([7])
	Line	([8])

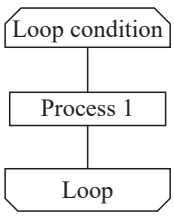
A Start of a repetition
B Start/End
C Flow of data and control
D Branching according to conditions
E End of a repetition
F Display on a screen, etc.
G Data input and output
H Operations and other processes

(3) Choose the one that represents [1] a branching structure and [2] a repeating structure from the following flowcharts **A** to **C**, and answer using the letters.

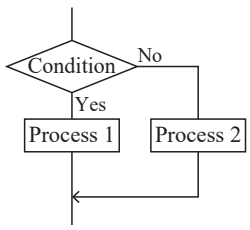
A



B

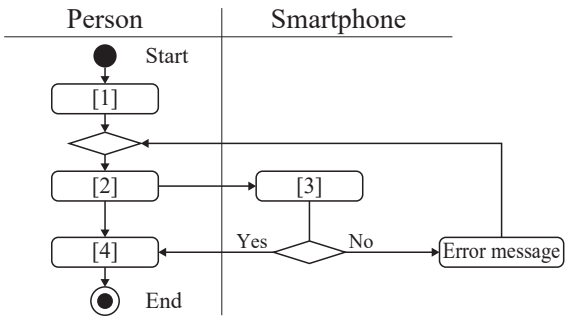


C



(4) The figure on the right is an activity diagram that illustrates the relationship between yourself and a smartphone when unlocking its screen. Choose the option that best fits into the blanks [1] to [4] from the options **A** to **D** below, and answer using the letters.

A Verify password
B Enter password
C Unlock screen
D Screen turns ON



(5) Name the programming language that can be confirmed only within a web browser, making it optimal for web-related purposes.

Exercise

1 Cover the **Point!** section on Pages 154 and 155 with a red sheet and test yourself by writing the items in order in your notebook.

2 Answer the following questions.

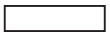
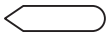
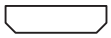
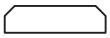
(1) Choose the term that best fits into the blanks [1] to [3] in the following sentences from the options **A** to **F** below, and answer using the letters.

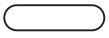
> The order of calculations or the sequence for creating something is referred to as an ([1]). Among the diagrams that visually represent this, the one that is suitable for representing a single process flow is referred to as a ([2]), while the one suitable for representing parallel process flows is referred to as an ([3]).

A Programming **B** Programming language **C** Activity diagram
D Source code **E** Algorithm **F** Flowchart

(2) Choose the symbols used in flowcharts that correspond to the meanings [1] to [4] from the options **A** to **F** below, and answer using the letters.

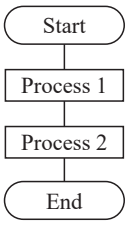
[1] Start/End [2] Branching according to conditions
[3] Start of a repetition [4] Data input and output

A  **B**  **C**  **D** 

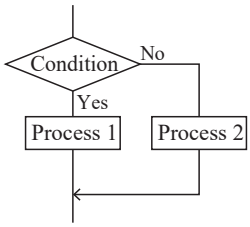
E  **F** 

(3) From the following flowcharts, choose the one from **A** to **C** that best represents a flowchart depicting the case where "If the product is in stock, purchase it; otherwise, stop shopping." Answer with the appropriate letter.

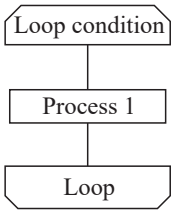
A



B

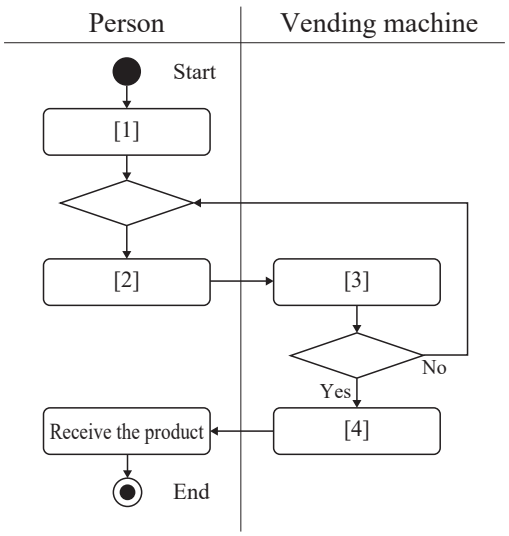


C



(4) The figure on the right is an activity diagram showing the relationship between yourself and a vending machine when purchasing a product. Choose the option that best fits into the blanks [1] to [4] from the options **A** to **D** below, and answer using the letters.

A Insert money
B Money is counted
C Press button to purchase the product
D Product is discharged



(5) Choose the visual programming language developed for beginners and children from the options **A** to **D**, and answer using the letter.

A Python **B** Scratch
C JavaScript **D** BASIC

12-2 Programming Basics [1] (Python)

Point!

Variables and Sequential Structure

(1) ([1]**print**) : Displays the character string or value inside the parentheses ( ).
The character string is enclosed with half-width single quotation marks (').
*Note that numerical values should not be enclosed using half-width quotation marks.

01	`print('Hanako Yamada')`	Displays "`Hanako Yamada`".
02	`print(2023)`	Displays "`2023`".

Execution result

```
Hanako Yamada
2023
```

(2) ([2]**Variable**) : Like a box to store data. Data can be stored in and retrieved from variables.
<Example> A program to display the character string "`Tokyo`"

01	`city = 'Tokyo'`	Assigns "`Tokyo`" to the variable "`city`".
02	`print(city)`	Displays the value for the variable "`city`".

Execution result

```
Tokyo
```

* In programming, the "=" symbol does not mean "equal." Rather, it means "([3]**assign the right side to the left side**)."

(3) ([4]**Assignment operator**) : Something such as "=" used to assign a value to a variable.

(4) ([5]**Arithmetic operator**) : A symbol such as "+" and "-" used in calculations.

Arithmetic operator	Meaning
`a + b`	([6]**Addition**)
`a - b`	([7]**Subtraction**)
`a * b`	([8]**Multiplication**)
`a / b`	([9]**Division**)
`a // b`	([10]**Quotient**)
`a % b`	([11]**Remainder when a is divided by b**)
`a ** b`	([12]**a to the power of b**)

<Example> A program that performs four basic arithmetic operations

01	`a = 5`	5 is assigned to variable a.
02	`b = 3`	3 is assigned to variable b.
03	`print(a + b)`	Displays a + b.
04	`print(a - b)`	Displays a - b.
05	`print(a * b)`	Displays a × b.
06	`print(a / b)`	Displays a ÷ b.
07	`print(a // b)`	Displays the quotient when a is divided by b.
08	`print(a % b)`	Displays the remainder when a is divided by b.
09	`print(a ** b)`	Displays a to the power of b.

Execution result

```
8
2
15
1.66666
1
2
125
```

Warm Up

Answer the following questions.

(1) Choose the program that displays "HelloWorld!" from the options **A** to **D** below, and answer using the letter.

A
```
HelloWorld!
```

B
```
'HelloWorld!'
```

C
```
print(HelloWorld!)
```

D
```
print('HelloWorld!')
```

(2) Choose the program that displays "Mr. Suzuki" from the options **A** to **D** below, and answer using the letter.

A
```
print(name)
name = 'Mr. Suzuki'
```

B
```
print(name)
name == 'Mr. Suzuki'
```

C
```
name = 'Mr. Suzuki'
print(name)
```

D
```
name == 'Mr. Suzuki'
print(name)
```

(3) When the following program is executed, give the results for items [1] to [5].

```
a = 6
print(a + 4)     .....[1]
print(a - 1)     .....[2]
print(a * 5)     .....[3]
print(a / 2)     .....[4]
print(a ** 3)    .....[5]
```

Explanation

(1) The `print` function is used to display character strings or values, and the character string "`HelloWorld`"! is enclosed with either ' or ". Therefore, **D**

(2) "`=`" is used to assign a value to a variable. In addition, a variable cannot be used before it is defined. Therefore, **C**

(3) Since the value 6 is assigned to the variable "`a`", [1] is 6 + 4, which equals 10; [2] is 6 - 1, which equals 5; [3] is 6 × 5, which equals 30; [4] is 6 ÷ 2, which equals 3; and [5] is 6 to the power of 3, which equals 216.
Therefore, the answers are [1] `10`, [2] `5`, [3] `30`, [4] `3`, [5] `216`

Try

Answer the following questions.

(1) Choose the program that displays "`Hello`" from the options **A** to **D** below, and answer using the letter.

A
```
Hello
```

B
```
'Hello'
```

C
```
print(Hello)
```

D
```
print('Hello')
```

(2) Choose the program that displays "`Correct`" from the options **A** to **D** below, and answer using the letter.

A
```
print(result)
result = 'Correct'
```

B
```
print(result)
result == 'Correct'
```

C
```
result = 'Correct'
print(result)
```

D
```
result == 'Correct'
print(result)
```

(3) Choose the program that displays "`2`" from the options **A** to **D** below, and answer using the letter.

A
```
a = 20
print(a % 3)
```

B
```
print(a % 3)
a = 20
```

C
```
a = 20
print(a * 3)
```

D
```
print(a * 3)
a = 20
```

(4) Choose the program that displays "`81`" from the options **A** to **D** below, and answer using the letter.

A
```
b == 9
print(b // 2)
```

B
```
b = 9
print(b // 2)
```

C
```
b == 3
print(b ** 4)
```

D
```
b = 3
print(b ** 4)
```

(5) When the following program is executed, give the results for items [1] to [5].

```
c = 5
print(c + 3)     .....[1]
print(c - 2)     .....[2]
print(c * 4)     .....[3]
print(c / 2)     .....[4]
print(c ** 2)    .....[5]
```

(6) What is the term for a symbol like "`=`" used to assign a value to a variable? Choose one from the options **A** to **D** below, and answer using the letter.

A Arithmetic operator
B Comparison operator
C Assignment operator
D Logical operator

Exercise

1 Cover the **Point!** section on page 158 with a red sheet and test yourself by writing the items in order in your notebook.

2 Answer the following questions.

(1) Choose the program that displays "`Nice to meet you`" from the options **A** to **D** below, and answer using the letter.

A
```
print('Nice to meet you')
```

B
```
print(Nice to meet you)
```

C
```
'Nice to meet you'
```

D
```
Nice to meet you
```

(2) Choose the program that displays "`17 years old`" from the options **A** to **D** below, and answer using the letter.

A
```
print(age)
age == '17 years old'
```

B
```
print(age)
age = '17 years old'
```

C
```
age == '17 years old'
print(age)
```

D
```
age = '17 years old'
print(age)
```

(3) Choose the program that displays "`15`" from the options **A** to **D** below, and answer using the letter.

A
```
a = 5
print(a % 3)
```

B
```
print(a % 3)
a = 5
```

C
```
a = 5
print(a * 3)
```

D
```
print(a * 3)
a = 5
```

(4) Choose the program that displays "`2`" from the options **A** to **D** below, and answer using the letter.

A
```
b == 8
print(b // 3)
```

B
```
b = 8
print(b // 3)
```

C
```
b == 8
print(b ** 3)
```

D
```
b = 8
print(b ** 3)
```

(5) When the following program is executed, give the results for items [1] to [5].

```
c = 9
print(c + 2)     ……[1]
print(c - 4)     ……[2]
print(c * 2)     ……[3]
print(c // 2)    ……[4]
print(c % 2)     ……[5]
```

(6) What is the term for symbols such as "`+`" and "`-`" that are used in calculations? Choose one from the options **A** to **D** below, and answer using the letter.

A Comparison operator
B Logical operator
C Assignment operator
D Arithmetic operator

12-3 Programming Basics [2] (Python)

Point!

1 Loop Structure

(1) Using a ([1]for statement) makes it possible to execute a process repeatedly.
A "for" statement is written as shown below.

```
for␣variable␣in␣range([Variable range]):
␣␣␣␣[Process to repeat]
```

In a "for" statement, the indented block of code is executed repeatedly.

(2) In a "for" statement, the range() instruction can be used to specify a range for a variable.

Writing style	Meaning
range(end value)	The value of the variable increases by 1, starting from ([2]0) up to the ([3]end value - 1).
range(start value, end value)	The value of the variable increases by 1, starting from the ([4]start value) up to the ([5]end value - 1).
range(start value, end value, increment)	The value of the variable increases by the specified ([6]increment), starting from the ([7]start value) up to the ([8]end value - increment).

<Example> A program that displays integers from 0 to 3

```
01 for i in range(0, 4):      Repeat variable "i" increasing by 1 from 0 to 3.
02     print(i)               Displays the value of variable "i".
```

Execution result

```
0
1
2
3
```

<Example> A program that displays odd numbers from 1 to 5

```
01 for i in range(1, 7, 2):   Repeat variable "i" increasing by 2 from 1 to 6.
02     print(i)               Displays the value of variable "i".
```

Execution result

```
1
3
5
```

2 Branching Structure

(1) ([9]Comparison operator) : Operator used to compare expressions or values.

Comparison operator	Meaning	Example	Meaning of the example
==	([10]Equal)	x == 70	x is equal to 70
! =	([11]Not equal)	x ! = 70	x is not equal to 70
<	([12]Less than)	x < 70	x is less than 70
>	([13]Greater than)	x > 70	x is greater than 70
<=	([14]Less than or equal to)	x <= 70	x is less than or equal to 70
>=	([15]Greater than or equal to)	x >= 70	x is greater than or equal to 70

(2) Using ([16]**if statements**) allows conditions to be branched and processed as conditional expressions.

- ([17]**Conditional expression**) : Expression that determines whether conditions are met and returns "true" if met and "false" if not.

[1] Using ([18]**if ~ else**) makes it possible to describe the process that occurs when a condition is not met.

```
if␣[Conditional expression]:
    [Process when a condition is true]
else:
    [Process when a condition is false]
```

<Example> Program for determining pass or fail

01	`x = 70`	Assign the value `70` to variable `x`.
02	`if␣x >= 60:`	If `x` is `60` or greater,
03	`␣␣␣␣result = 'Pass'`	If true, assign "`Pass`" to the variable `result`.
04	`else:`	If not,
05	`␣␣␣␣result = 'Fail'`	Assign "`Fail`" to the variable `result`.
06	`print(result)`	Displays the value of the variable `result`.

Execution result

```
Pass
```

[2] Using ([19]**if ~ elif ~ else**) makes it possible to determine multiple conditions sequentially.

```
if [Conditional expression 1]:
␣␣␣␣[Process when conditional expression 1 is true]
elif [Conditional expression 2]:
␣␣␣␣[Process when conditional expression 1 is false and conditional expression 2 is true]
else:
␣␣␣␣[Process when both conditional expression 1 and conditional expression 2 are false]
```

<Example> Program for evaluating based on test scores

01	`x = 70`	Assign the value `70` to variable `x`.
02	`if␣x >= 90:`	If `x` is `90` or greater,
03	`␣␣␣␣result = 'Grade is A'`	If true, assign "`Grade is A`" to the variable `result`.
04	`elif␣x >= 50:`	Otherwise, if `x` is `50` or greater,
05	`␣␣␣␣result = 'Grade is B'`	Assign "`Grade is B`" to the variable `result`.
06	`else:`	If none of the conditions are met,
07	`␣␣␣␣result = 'Grade is C'`	Assign "`Grade is C`" to the variable `result`.
08	`print(result)`	Displays the value of the variable `result`.

Execution result

```
Grade is B
```

Warm Up

Answer the following questions.

(1) Choose the program that displays "`Grade is A`" from the options **A** to **D** below, and answer using the letter.

A
```
point = 90
if point >= 85:
    result = 'Grade is A'
else:
    result = 'Grade is B'
```

B
```
point = 90
if point >= 85:
    result = 'Grade is A'
else:
    result = 'Grade is B'
print(result)
```

C
```
point 90
if point >= 85:
    result = 'Grade is A'
else:
    result = 'Grade is B'
print(result)
```

D
```
point 90
if point >= 85:
    result = 'Grade is A'
else:
    result = 'Grade is B'
```

(2) The following program displays the total and the average of all integers from `1` to `10`. Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
total = 0
for i in range(1, [ A ]):
    total = [ B ]
average = [ C ] /10
print(total)
print(average)
```

Explanation

(1) "`=`" is needed to assign a value to the variable "`point`", and the print function is needed to display the value of the variable "`result`". Therefore, **B**

(2) Set the variable "`total`" to an initial value of `0` and then calculate the total by continuously adding integers `1` to `10` as variable "`i`" to the variable "`total`". When counting variables from `1` to `10`, be sure to specify up to the next number, `11`. Therefore, **A**: `11`, **B**: `total+i`, **C**: `total`

Try

Answer the following questions.

(1) Choose the appropriate execution result when the following program is executed from the options **A** to **D** below, and answer using the letter.

```
for i in range(0, 5, 1):
    print(i)
```

A
```
0
1
2
3
4
```

B
```
0
1
2
3
4
5
```

C
```
1
2
3
4
```

D
```
1
2
3
4
5
```

(2) Choose the program that displays "`Pass`" from the options **A** to **D** below, and answer using the letter.

A
```
score 95
if score < 90:
    result = 'Fail'
else:
    result = 'Pass'
print(result)
```

B
```
score 95
if score < 90:
    result = 'Fail'
else:
    result = 'Pass'
```

C
```
score = 95
if score < 90:
    result = 'Fail'
else:
    result = 'Pass'
```

D
```
score = 95
if score < 90:
    result = 'Fail'
else:
    result = 'Pass'
print(result)
```

(3) The following program counts down from 5 to 0, and when it reaches 0, it displays "`Start!`" Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
for i in [ A ]:
    count = [ B ]
    if [ C ]:
        print('Start!')
    else:
        print(count)
```

(4) The following program displays "`It is an even number`" for even numbers from 1 to 100. Fill in the blanks **A** and **B** with the appropriate characters or numbers to complete the program.

```
for i in range(1, [ A ]):
    if [ B ]:
        print('It is an even number')
    else:
        print(i)
```

Exercise

1 Cover the **Point!** section on pages 162 and 163 with a red sheet and test yourself by writing the items in order in your notebook.

2 Answer the following questions.

(1) Choose the program that would produce the same execution result as the following program from the options **A** to **D** below, and answer using the letter.

```
for i in range(3):
    print(i)
```

A
```
for i in range(0, 3):
    print(3)
```

B
```
for i in range(0, 3, 1):
    print(i)
```

C
```
for i in range(0, 3, 1):
    print(3)
```

D
```
for i in range(0, 3, 2):
    print(i)
```

(2) Choose the appropriate execution result when the following program is executed from the options **A** to **C** below, and answer using the letter.

```
x = 7
if x < 3:
    print('We will seat you at the counter')
elif x <= 10:
    print('We will seat you at a table')
else:
    print('No seats are available')
```

A `We will seat you at the counter`
B `We will seat you at a table`
C `No seats are available`

(3) The following program displays the total for all numbers from 1 to 100. Fill in the blanks **A** and **B** with the appropriate characters or numbers to complete the program.

```
total = 0
for i in range(1, [ A ]):
    total = [ B ]
print(total)
```

(4) The following program displays the number of even numbers between 1 and 10. Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
for i in [ A ]:
    if i % 2 == 0:
        count = count + [ B ]
print( [ C ] )
```

12-4 Application of Programming [1] (Python)

Point!

Array (List)

(1) ([1]**Array (list)**) : A collection of data items arranged in sequence. Data can be managed collectively.

Array declaration	Array name = [data0, data1, data2, …]
Adding elements to the end of an array	Array name.append(data)

<Example> Character string array `week = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']`
Array of numbers `number = [7, 22, 11, 34, 17]`

(2) ([2]**Element**) : Each value included in an array. Displays the array name and its place number
Using ([3]**subscripts**)(element numbers) makes it possible to retrieve elements from an array.
Note that subscripts do not start from 1, but from ([4]0).

a	7	22	11	34	17
	a[0]	a[1]	a[2]	a[3]	a[4]

(3) Array operation : Allows array element to be managed using a single subscript.
<Example> Array declaration and method for accessing elements
Using a variable as a subscript such as in `a[i]` makes it possible to specify elements with the variable.
This is convenient when processing elements one at a time using a "for" statement.

01	`a = [7, 22, 11, 34, 17]`	Declares array "a" and assigns a numerical value to it.
02	`for␣i␣in␣range(0,5,1):`	Repeats while increasing variable `i` from `0` to ([5]4) in increments of `1`.
03	`␣␣␣␣print(a[i])`	Displays elements from array "a" that satisfy the condition ([6]`a[i]`).

Execution result
```
7
22
11
34
17
```

<Example> Adding an element to the end of a one-dimensional array

01	`a = [ ]`	Declares the array "a".
02	`a.append(1)`	Adds ([7]1) to the end of array "a".
03	`a.append(4)`	Adds ([8]4) to the end of array "a".
04	`a.append(9)`	Adds ([9]9) to the end of the array "a".
05	`print(a)`	Displays the values for array "a".

Execution result
```
[1, 4, 9]
```

(4) ([10]**Two-dimensional array**) : An array that manages data using subscripts in both the row and column directions.
An element at row "i" and column "j" would be specified using the two subscripts as `a[i][j]`.

<Example> Definition of a two-dimensional array and the method for accessing elements

01	`a=[['A', 'B', 'C'], ['D', 'E', 'F'],]`	Declares array "a" and assigns a character string to it.
02	`print(a[0][0])`	Displays the value for ([11]`a[0][0]`).
03	`print(a[1][2])`	Displays the value for ([12]`a[1][2]`).

Execution result
```
A
F
```

Warm Up

Answer the following questions.

(1) For programs **A** and **B** below, give the values displayed when each is executed.

A
```
a = [57, 16, 29, 44]
print(a[2])
```

B
```
a = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
print(a[2][1])
```

(2) The following program is designed to find the minimum value among the elements in array "a". Fill in the blanks **A** and **B** with the appropriate characters or numbers to complete the program.

```
a = [34, 52, 11, 40, 17]
min = a[0]
for i in range(1, [ A ], 1):
    if a[i] < min:
        min = [ B ]
print(min)
```

(3) Fill in the blanks **A** to **E** in the following program with the appropriate characters or numbers to complete the program so that it displays the "Execution result" as shown.

```
a = [['A', 'B', 'C', 'D'],
     ['E', 'F', 'G', 'H']]
for i in range([ A ], [ B ], 1):
    for j in range([ C ], [ D ], 1):
        print([ E ])
```

Execution result

```
A
B
C
(Omitted)
H
```

Explanation

(1) Note that array subscripts start at `0`, not `1`. In addition, an element in a two-dimensional array can be represented as `a[i][j]`, where "i" indicates the row and "j" indicates the column. Therefore, **A**: `29`, **B**: `8`

(2) Set "`min`" as the variable that stores the minimum value of the array "`a`" so it is `a[0]`. Next, examine each element of array "`a`" sequentially, and if `a[i]` is smaller than `min`, update `min` to that value.
Therefore, **A**: `5`, **B**: `a[i]`

(3) **A**: `0`, **B**: `2`, **C**: `0`, **D**: `4`, **E**: `a[i][j]`

Try

Answer the following questions.

(1) For programs **A** to **C** below, give the values displayed when each is executed.

A
```
a = [1, 4, 9, 16, 25]
print(a[3])
```

B
```
a = [ ]
a.append(8)
a.append(28)
print(a[1])
```

C
```
a = [['A', 'B', 'C', 'D', 'E'],
     ['F', 'G', 'H', 'I', 'J'],
     ['K', 'L', 'M', 'N', 'O']]
print(a[2][1])
```

(2) The following program finds the total of the elements in array "a". Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
sum = 0
for i in range( A , B , 1):
    sum = sum + C
print(sum)
```

(3) The following program finds the maximum value among the elements in array "a". Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
a = [7, 22, 11, 34, 17]
max = 0
for i in range(0, A , 1):
    if a[i] B max:
        max = C
print(max)
```

(4) Fill in the blanks **A** to **E** in the following program with the appropriate characters or numbers to complete the program so that it displays the "Execution result" as shown.

```
a = [['A', 'B', 'C', 'D', 'E'],
     ['F', 'G', 'H', 'I', 'J']]
for i in range( A , B , 1):
    for j in range( C , D , 1):
        print( E )
```

Execution result

```
A
B
C
(Omitted)
J
```

Exercise

1 Cover the **Point!** section on page 167 with a red sheet and test yourself by writing the items in order in your notebook.

2 Answer the following questions.

(1) For programs **A** to **C** below, give the values displayed when each is executed.

A
```
a = [0, 1, 2, 3]
print(a[1])
```

B
```
a = ['A']
a.append('B')
a.append('C')
print(a[1])
```

C
```
a = [['Japanese', 'Mathematics', 'English'],
     ['Social studies', 'Science', 'Information studies']]
print(a[1][2])
```

(2) The following program is designed to find the minimum value among the elements in array "a". Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
a = [24, 43, 9, 25, 17]
min = [ A ]
for i in range(0, [ B ], 1):
    if a[i] < min:
        min = [ C ]
print(min)
```

(3) The following program counts the number of elements in array "a" that are greater than 10. Fill in the blanks **A** to **C** with the appropriate characters or numbers to complete the program.

```
a = [12, 8, 9, 13, 11]
count = 0
for i in range([ A ], [ B ], 1):
    if [ C ] > 10:
        count = count + 1
print(count)
```

(4) Fill in the blanks **A** to **E** in the following program with the appropriate characters or numbers to complete the program so that it displays the "Execution result" as shown.

```
a = [['Japanese', 'Mathematics', 'English'],
     ['Social studies', 'Science', 'Information studies']]
for i in range([ A ], [ B ], 1):
    for j in range([ C ], [ D ], 1):
        print([ E ])
```

Execution result

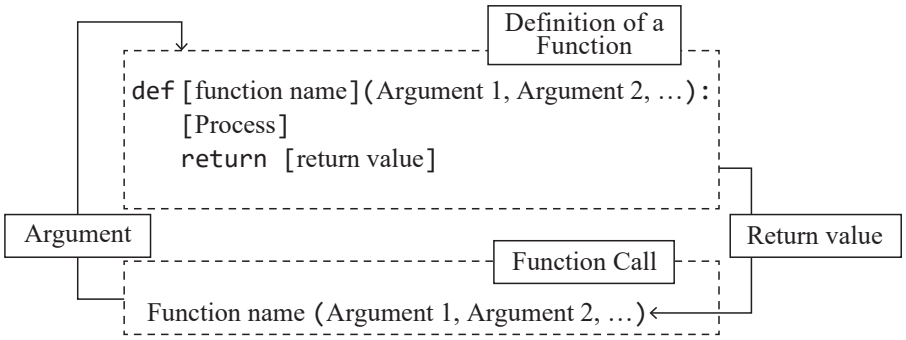
```
Japanese
Mathematics
English
Social studies
Science
Information studies
```

12-5 Application of Programming [2] (Python)

Point!

Functions

(1) ([1]**Function**) : A set of operations performed by a computer described as a cohesive unit. A variable specified to receive a value within a function is referred to as an ([2]**argument**). Using ([3]**return**) makes it possible to pass the value resulting from the process within the function to the caller as a ([4]**return value**). In addition, when a function is used, it is necessary to perform the process of ([5]**calling**) the function.



<Example> Example of a function with one argument: A program to find the function y = 2x

01	`def function(x):`	Defines a `function` called ([6]**function**).
02	`    y = 2 * x`	Assigns the value from `2*x` to variable "`y`".
03	`    return y`	Returns "`y`" as ([7]**return value**).
04	`print(function(5))`	Passes "`x`" as 5 to the `function`.

Execution result

```
10
```

<Example> Example of a function with two arguments: A program to find the area of a triangle

01	`def area(base, height):`	Defines a function called "`area`".
02	`    S= base*height/2`	Assigns the value from "`base*height/2`" to variable "`S`".
03	`    return S`	Returns "`S`" as the return value.
04	`print(area(10, 5))`	Passes "`base`" as ([8]`10`) and "`height`" as ([9]`5`) to the function "`area`".

Execution result

```
25
```

(2) Functions include ([10]**built-in functions**) such as "`print()`" that can be used without definition, and ([11]**user-defined functions**) that you can define.

Warm Up

Answer the following questions.

(1) In the following program, the function "`circle`" is used to calculate the area of a circle. Give the program that fits in blanks **A** and **B**. Also, give the value displayed when row [1] is executed.

```
 A  circle(r):
    S = r * r * 3.14
    B  S
a = circle(5)
print(a)      # [1]
```

(2) In the following program, the function "`judge`" takes the test score as an argument and displays "`Pass`" if the score is 80 or above, and "`Fail`" if it is below 80. Give the program that fits in blanks **A** and **B**.

```
def judge(score):
    if score  A  80:
        print('Pass')
    B :
        print('Fail')
judge(40)
```

Explanation

(1) In programming, the function is defined as shown below.

```
def function name(argument):
    [Process]
    return [return value]
```

Therefore, the programs that fit in blanks **A** and **B** are as follows.

A: `def`, **B**: `return`

In addition, when the function "`circle`" is called in [1], 5 is passed as argument "`r`" in the function "`circle`".

Therefore, when row [1] is executed, it is `5 × 5 × 3.14 = 78.5`, so `78.5` is displayed.

(2) In the programming, a conditional branch was defined as shown below.

```
if conditional expression:
    [Process when a condition is true]
else:
    [Process when a condition is false]
```

Therefore, the programs that fit in blanks **A** and **B** are as follows.

A: `>=`, **B**: `else`

Try

Answer the following questions.

(1) Choose the term that best fits into the blanks [1] and [2] from the options **A** to **D** below, and answer using the letters.

```
def function name ( [1] ):
    [Process]
    return [2]
```

A Argument **B** Return value **C** Complement **D** Random number

(2) In the following program, the function "`area`" is used to calculate the area of a triangle. Give the program that fits in blanks **A** and **B**. Also, give the values displayed when rows [1] and [2] are executed.

```
def area(base, height):
    S = A
    B S
a = area(10, 5)
b = area(6, 7)
print(a)      # [1]
print(b)      # [2]
```

(3) In the following program, the function "`celsius_to_fahrenheit`" converts temperatures from `Celsius` to `Fahrenheit`. Give the program that fits in blanks **A** and **B**.

```
def celsius_to_fahrenheit(celsius):
    fahrenheit = (celsius * 9/5) + 32
    return A
temp_celsius = 25
result = celsius_to_fahrenheit( B )
print(result)
```

(4) In the following program, test scores are given as arguments to the function "`evaluate`". In addition, the relationship between scores and grades is according to the table on the right. Give the program that fits in the blanks **A** to **C**.

Score	Grade
80 or higher	A
50 or higher but less than 80	B
Less than 50	C

```
def evaluate(score):
    if score A 80:
        print('A')
    B score C 50:
        print('B')
    else:
        print('C')
evaluate(40)
```

Exercise

1 Cover the **Point!** section on page 171 with a red sheet and test yourself by writing the items in order in your notebook.

2 Answer the following questions.

(1) In the following program, the function "`add_number`" adds two numbers together. Give the program that fits in blanks **A** and **B**.

```
 A  add_numbers(a, b):
    return  B
result = add_numbers(3, 5)
print(result)
```

(2) In the following program, the function "`area`" is used to calculate the area of a quadrilateral. Give the program that fits in blanks **A** and **B**. Also, give the values displayed when rows [1] and [2] are executed.

```
def area(width, height):
    S =  A
     B  S
a = area(10, 6)
b = area(7, 5)
print(a)      # [1]
print(b)      # [2]
```

(3) In the following program, egg weight is given as an argument to the function "`check_size`". In addition, the relationship between egg weight and size is according to the table on the right. Give the program that fits in blanks **A** and **B**.

Egg weight	Size
Less than 45 g	S
45 g or higher but less than 55 g	M
55 g or higher	L

```
def check_size(weight):
    if weight < 45:
       print('S')
    elif weight  A  55:
       print('M')
     B :
       print('L')
check_size(40)
```

(4) What is the term for a function provided as a standard feature of a programming language such as "`print()`"?