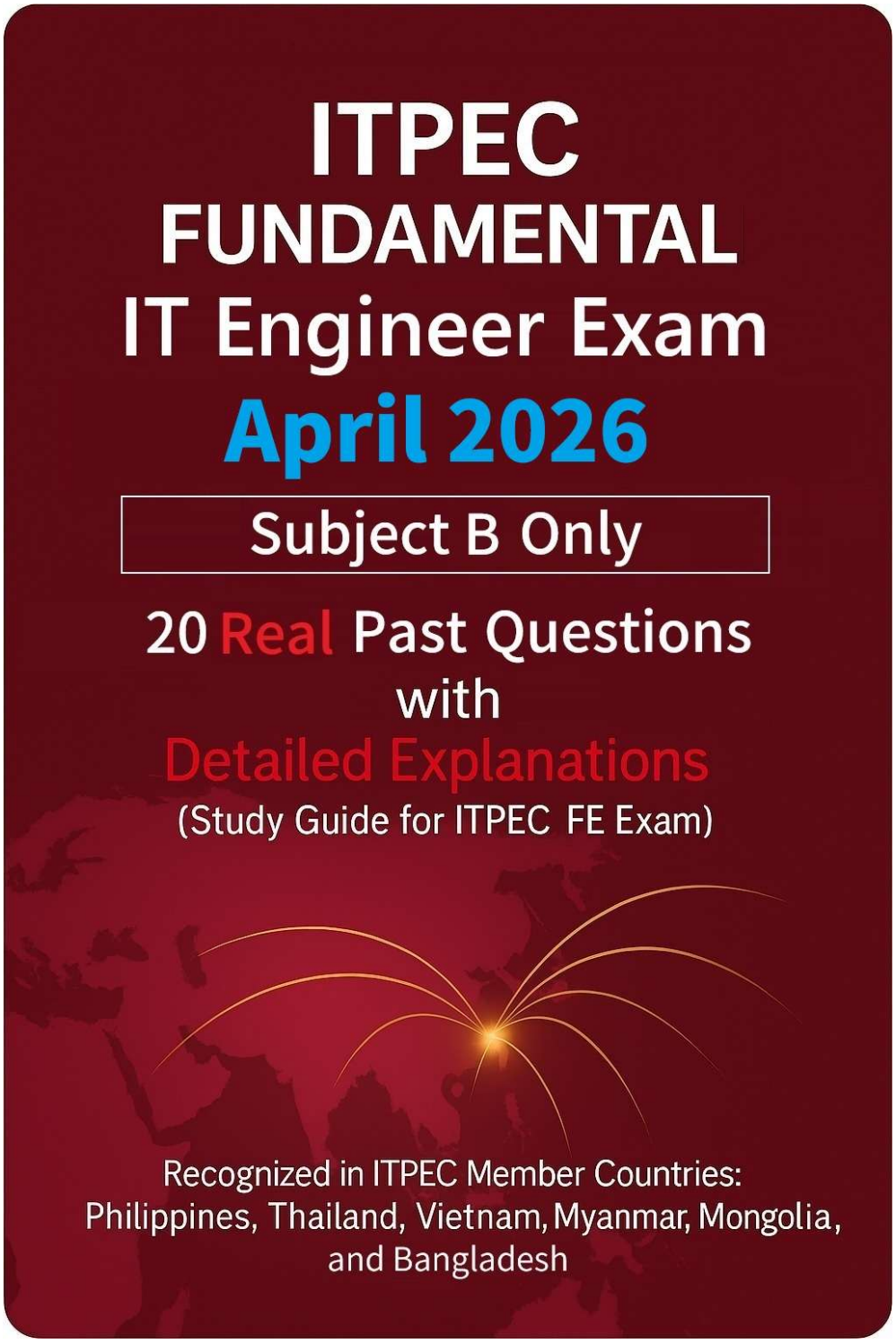
This Book is Sample Edition



ITPEC Fundamentals of Information Technology Engineer Examination (FE)

– English Edition April 2026 Subject B Version

Author: Takashi Narita

Based on past **FE Examination** questions published by ITPEC.

Disclaimer

This book contains past exam questions from the Fundamental Information Technology Engineer Examination (FE), administered by the Information Technology Professionals Examination Council (ITPEC), specifically from the April 2026 Examination. The questions are reproduced with permission for educational purposes, and the copyright of the original questions remains with ITPEC.

All explanations, translations, and analyses are original work by the author, Takashi Narita.

Preface

The **Fundamental Information Technology Engineer Examination (FE)** is a core examination that demonstrates your ability to apply fundamental knowledge of information technology, programming, algorithms, and system design. Originally developed in Japan, it is now recognized internationally through the ITPEC (Information Technology Professionals Examination Council) mutual recognition framework. Member countries include the Philippines, Thailand, Vietnam, Myanmar, Mongolia, and Bangladesh.

In many of these countries, the FE serves as a gateway for those aiming to work in Japan or for Japanese companies abroad. Where Japanese-owned IT firms operate, holding this certification can also provide a strong advantage in hiring and career advancement, as it reflects both technical competence and familiarity with Japanese corporate culture and IT standards.

This book is designed for learners who wish to prepare for the **FE Examination in English**. It offers clear explanations, answer analyses, and study tips to help you build genuine understanding rather than relying on rote memorization. Whether you are a student, a working professional, or someone seeking to advance your IT career, this book will support efficient, practical, and motivating study.

Important Notice

The ITPEC Fundamentals of Information Technology Engineer Examination (FE) consists of two parts: Subject A and Subject B.

Each subject has different characteristics and types of questions.

This book covers only Subject B.

➞ For Subject A, please purchase the separate volume.

➞ A combined edition (Subjects A & B) is also available.

About the Author

Takashi Narita is an IT instructor with more than 20 years of industry experience in system design, software development, programming, and database management. Now based in the Philippines, he teaches programming, system development, and system design remotely to students in Japan.

He began creating this English edition of **Fundamental Information Technology Engineer Examination (FE)** practice questions to support IT human resource development in the Philippines. Over time, he recognized that it could also serve as a valuable resource for learners across Asia seeking to succeed in Japanese-owned companies or in Japan itself.

Through this book, he aims to help readers build the skills and confidence needed to contribute meaningfully in Japanese and Japan-affiliated workplaces. He is passionate about making complex IT concepts accessible to learners of all backgrounds and believes that studying in English opens doors to international opportunities.

How to Use This Book

The fastest way to pass the **Fundamental Information Technology Engineer Examination (FE)** is to practice past questions repeatedly and understand the logic behind each answer.

This book provides past FE questions in English, along with clear explanations and answer analyses. Always review the explanations for any questions you miss, make sure you understand the reasoning, and then try again.

Recommended study flow:

1. Solve one set of questions under timed conditions, just like the actual exam.
2. Check which answers were incorrect and study the explanations carefully.
3. Reattempt the same set a few days later until you can achieve a high score with confidence.
4. Move on to the next set and repeat the process.

Study Tips

• **Do more than one set** – One set of past questions is not enough for thorough preparation. Aim for at least three sets to build both knowledge and confidence.

• **Consistency is key** – Even 10 minutes a day makes a difference. Use your phone, PC, or tablet to open this book and solve just one question. From my own experience earning multiple IT certifications, daily habit is the single most important factor in success. For me, solving one question before bed soon grew naturally to two or three without feeling forced. Short, consistent study sessions lead to long-term retention.

• **Set a study schedule** – Work backward from your exam date, create a plan, and track your progress.

• **Get used to English IT terms** — many are 3–4-letter abbreviations (e.g., CPU, LAN, DNS) and can feel intimidating at first, but with repeated exposure they'll become second nature.

Exam Overview

• **Format: Two parts** — **Subject A** (formerly “Morning”) and **Subject B** (formerly “Afternoon”).

• **Exam Areas (Subject A):** 60 multiple-choice questions in 90 minutes.

Subject A consists of **60 questions** drawn from the three domains below.

1) Technology Domain

IT fundamentals, hardware, software, databases, networks, security, etc.

2) Management Domain

project management, service management, system audit, etc.

3) Strategy Domain

management strategy, business law, system strategy, corporate activities, etc.

You should aim to spend about 1.5 minutes (90 seconds) per question.

• **Exam Areas (Subject B):** 20 multiple-choice questions in 100 minutes.

Algorithms & Programming (pseudo-code); Information Security.

(**20 questions** total: 16 Algorithms & Programming, 4 Information Security)

You should aim to spend about 5 minutes per question.

• **Passing Criteria (FE):**

You must pass both Subject A and Subject B. The passing mark for each subject is 60 out of 100.

Links

The test mode (CBT or paper), schedule, application method, and venue vary by country. Details are subject to change; always check your country's official website for the latest information.

Official Information Links:

- ITPEC Official Site: https://itpec.org/
- Philippines (DICT): https://dict.gov.ph/itpec/
- Thailand (NECTEC): https://www.nectec.or.th/itpec/
- Vietnam (VITC): https://www.vitc.vn/
- Myanmar (ITPEC Myanmar): https://www.myanmaritpec.org/
- Bangladesh (BCC): https://www.bcc.gov.bd/
- Mongolia (MITC): https://www.mitc.gov.mn/

Author's Note: A quick word on Subject B

Subject B tests **programming & algorithms**.

Unlike Subject A—where past-like items can be solved by "recognition"—you'll **rarely** see the exact same algorithm. What matters is **understanding and application**.

When it feels hard — do this

1. Peek if needed, fill the blanks, then **trace the finished code** yourself.
2. Use **tiny inputs** and a quick table (track i, j, sum, max, …).
3. Check **stopping conditions & boundaries** (loop end, indices, empty/one/many).
4. Learn the **shape** of formulas (e.g., relative error, Hamming distance)—deep math not required.

With these habits, fill-in questions get much easier. Unfamiliar names won't stop you if you can **trace correctly**.

Bottom line: Subject B is often the hurdle. **Master basics, repeat patterns, move one steady step at a time.**

Note: Styles and coverage can change—review the latest syllabus and past papers.

*ITPEC provides an official **Pseudo-code Specification** in the downloadable past exams. For authoritative syntax/semantics, refer to those documents.*

Exams are always a race against time. If you glance at a question and feel a strong sense of dread, move on to another one. The best strategy is to start with the questions that make you think, "I can do this!"

B-Q1

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

When the program is executed, the output is “ _____ ”.

```
[Program]
    integer: x ← 1
    integer: y ← 2
    integer: z ← 3
    integer []: ar ← {0, 0}
    y ← x
    ar[x] ← y
    z ← y
    x ← z
    ar[2] ← z + ar[1]
    output ar[1], ar[2]    // the values are separated by ", "
```

Answer group

a) 1, 0

b) 1, 1

c) 1, 2

d) 2, 1

e) 2, 2

f) 2, 3

(Source:2026S,FE,Subject-B,Q1)

Answer: c

Explanation

To solve this problem, trace the values of the variables step by step.

Statement	x	y	z	ar[1]	ar[2]
Initial values	1	2	3	0	0
y ← x	1	1	3	0	0
ar[x] ← y	1	1	3	1	0
z ← y	1	1	1	1	0
x ← z	1	1	1	1	0
ar[2] ← z + ar[1]	1	1	1	1	2

The final values are:

- **ar[1] = 1**
- **ar[2] = 2**

Therefore, the output is: **1, 2**

So, the correct answer is:

c) 1, 2

Author's Comment

The most important skill in Subject B is **tracing**.

Do not try to calculate everything in your head. Instead, write down the values of each variable after every assignment statement.

Remember that an assignment changes only the variable on the left side. The variables on the right side keep their current values.

Column

Why Is Tracing Important?

Tracing is one of the most important skills in programming. It means following a program step by step to see how the values of variables change during execution. Even experienced programmers use tracing when they debug programs.

Many beginners try to read an entire program at once and become confused. A better approach is to execute one statement at a time. After each assignment, update the values of the variables on paper or in a table. This makes it much easier to understand what the program is doing.

Tracing is especially useful when working with arrays, loops, and recursive functions. These topics appear frequently in the FE Subject B examination, so developing good tracing habits will help you solve more difficult problems later.

As programs become more complex, systematic tracing becomes more important than trying to remember every value. Careful tracing is often the fastest and most reliable way to find the correct answer.

Before You Start Python Practice

The FE pseudocode cannot be executed directly on a computer. So why not try running the same algorithm in Python?

Python is easy to install, and you should be able to run the following programs without much difficulty. We encourage you to install Python and try the examples yourself.

Running a program is one of the best ways to understand how an algorithm works. Seeing the results with your own eyes will help you deepen your understanding.

Python Practice

The following Python program implements the same algorithm as the pseudocode in this question. Some expressions have been adjusted to match Python syntax and programming conventions. However, the algorithm remains the same. Run the program and compare it with the pseudocode to deepen your understanding.

The array indexes have been adjusted where necessary to match Python's 0-based indexing. The equivalent Python program is shown below.

```
# Q1 Python Practice
x = 1
y = 2
z = 3
# Python arrays start at index 0.
ar = [0, 0]
y = x
ar[x - 1] = y     # ar[1] in the pseudocode
z = y
x = z
ar[1] = z + ar[0]  # ar[2] ← z + ar[1]
print(ar[0], ar[1])
```

Did you get the same output as the correct answer? If not, review the program step by step and compare it with the pseudocode.

B-Q2

From the answer group below, select the correct combination of answers to be inserted into ____A____ through ____C____ in the description.

Function **f** receives an integer value as argument `argval` and returns the grade character.
The return value of f(92), f(72), and f(-10) are "_A_","_B_" , and "_C_", respectively. Note that the function has a specification bug.

```
[Program]
    ○ character: f(integer: argval)
        character: retvar ← "w"
        if (argval > 90)
            retvar ← "a"
        endif
        if (argval > 70)
            retvar ← "b"
        elseif (argval > 60)
            retvar ← "c"
        elseif (argval ≤ 60)
            retvar ← "d"
        endif
        return retvar
```

Answer group

	A	B	C
a)	a	b	d
b)	a	c	d
c)	a	c	w
d)	b	b	d
e)	b	c	d
f)	b	c	w

(Source:2026S,FE,Subject-B,Q2)

Answer: d

Explanation

Trace the program for each function call.

f(92)

- `retval` is initialized to "w".
- Since **92 > 90**, `retval` becomes "a".
- The next if statement is independent.
- Since **92 > 70**, `retval` is updated to "b".

The function returns **"b"**.

---

f(72)

- `retval` starts as "w".
- 72 > 90 is false.
- 72 > 70 is true, so `retval` becomes "b".

The function returns **"b"**.

---

f(-10)

- `retval` starts as "w".
- -10 > 90 is false.
- -10 > 70 is false.
- -10 > 60 is false.
- -10 ≤ 60 is true, so `retval` becomes "d".

The function returns **"d"**.

Therefore,

- **A = b**
- **B = b**
- **C = d**

The correct answer is **d**.

Author's Comment

This problem tests your understanding of the program flow rather than your programming knowledge.
Be careful not to assume that all if statements are connected. In this program, the first if statement and the second if–elseif statement are independent. Therefore, both if statements may be executed.
When solving Subject B questions, read the program one statement at a time and trace the execution carefully. This approach helps you avoid mistakes caused by assumptions.

Column

Understanding if and elseif

Many programming languages use if and elseif statements to make decisions. Although they look similar, they work differently depending on how they are written.
An if–elseif–else structure is treated as a single decision. Once one condition is true, the remaining conditions are skipped. This is useful when only one choice should be selected.
However, two separate if statements are independent. Even if the first if statement is true, the second if statement is still evaluated. As a result, the value of a variable may be changed again later in the program.
This difference is a common source of programming errors. Beginners sometimes expect two separate if statements to behave like one if–elseif structure. Understanding this difference helps you read programs correctly and avoid logical mistakes.
When tracing a program, always pay attention to where each if statement begins and ends. Careful observation of the program structure is just as important as understanding the conditions themselves.

Python Practice

```
# Q2 Python Practice
def f(argval):
    retvar = "w"
    if argval > 90:
        retvar = "a"

    if argval > 70:
        retvar = "b"
    elif argval > 60:
        retvar = "c"
    elif argval <= 60:
        retvar = "d"
    return retvar

print(f(92))
print(f(72))
print(f(-10))
```

Did you get the same output as the correct answer?

If not, review the program step by step and compare it with the pseudocode.

Notes and Disclaimers

Source Note:

The questions in this book are based on the past questions from the April 2026 FE (Fundamentals of Engineering) Examination published on the official ITPEC website. The author translated them into English and added supplementary explanations.

Source: ITPEC – FE Examination Past Questions (https://itpec.org/)

Relationship with ITPEC:

This book is an independent publication. It is created in accordance with the usage policy for past exam questions published by ITPEC, but it is not officially affiliated with, authorized, or endorsed by ITPEC.

General Disclaimer:

The content of this book is based on information available at the time of writing. The examination format, scope, or content may change. Please refer to the official ITPEC website for the most up-to-date **information.**

License and Usage:

This book is intended for personal study purposes. Redistribution or commercial use of the explanations and translations by the author without permission is prohibited.

Feedback Welcome:

If you find any errors or have suggestions for improvement, please feel free to contact the author. Your input will help improve future editions of this book.

Copyright & Source Information