

Sample Edition

**ITPEC
FUNDAMENTAL
IT Engineer Exam
October 2025**

Subject A Only

**60 Real Past Questions
with
Detailed Explanations**
(Study Guide for ITPEC FE Exam)

Recognized in ITPEC Member Countries:
Philippines, Thailand, Vietnam, Myanmar, Mongolia,
and Bangladesh

ITPEC Fundamentals of Information Technology Engineer Examination (FE)

– English Edition October 2025 Subject A Version

Author: Takashi Narita

© 2026 Takashi Narita. All rights reserved.

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 October 2025 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 A.

- For Subject B, 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 A

Subject A assesses **fundamental IT knowledge**. The most reliable way to reach the passing mark is to **practice many questions**. In recent years, Subject A has often included **reused or closely related items** from past exams, so broad exposure to previous questions is especially effective.

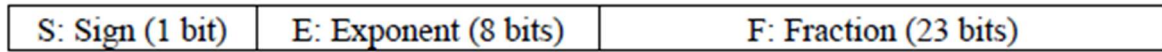
If you are new to the FE, consider taking the **IT Passport** exam first. Building that foundation makes Subject A **feel much easier** for many learners, allowing you to focus your energy on **Subject B** afterward.

*ITPEC provides official Subject A materials in the downloadable past exams. For authoritative conventions—especially **logic-circuit notation and rules**—please 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!”

A-Q1

In the single precision format of the “IEEE Standard for Binary Floating-Point Arithmetic” (IEEE 754), a 32-bit floating point number is represented as below:



If $0 < E < 255$, then the value of the number is $(-1)^S \times 2^{(E-127)} \times (1+F)$

where **F** represents the fractional part of the number, stored as a 23-bit binary sequence. This **F** is the sum of its weighted binary components:

$F = b_1 \times 2^{-1} + b_2 \times 2^{-2} + \dots + b_{23} \times 2^{-23}$, where **b₁, b₂, ..., b₂₃** are the binary digits of **F**.

According to the IEEE-754 standard, which of the following is the decimal equivalent of the 32-bit floating point number given below?

01000001011000000000000000000000

- a) 0.1875
- b) 0.4375
- c) 6.0
- d) 14.0

(Source:2025A,FE,Subject-A,Q1)

Answer: d

Explanation

To convert the 32-bit binary sequence into a decimal number, we must parse it according to the IEEE 754 single-precision format: 0 1000010 1100000000000000000000

1. **Sign (S):** The first bit is 0.

- Since $S = 0$, the number is positive ($(-1)^0 = 1$).

2. **Exponent (E):** The next 8 bits are 1000010.

- Converting this to decimal:

$$1 \times 2^7 + 0 \times 2^6 + \cdot \cdot \cdot + 1 \times 2^1 + 0 \times 2^0 = 128 + 2 = 130.$$

- Applying the bias (E - 127) : $130 - 127 = 3$. Thus, the multiplier is 2^3 .

3. **Fraction (F):** The remaining 23 bits start with 11 followed by zeros.

- $F = 1 \times 2^{-1} + 1 \times 2^{-2} = 0.5 + 0.25 = 0.75$.

4. **Final Calculation:**

- Formula: $(1 + F) \times 2^{E-127}$

- Result: $(1 + 0.75) \times 2^3 = 1.75 \times 8 = 14.0$

Author's Comment

When solving floating-point problems, the most common mistake is forgetting the "implicit 1" in the fraction part (1 + F). In IEEE 754, the leading 1 is not stored to save space, but it must be added during calculation.

Also, remember that the bias for single precision is always 127.

Column: Why do we use "Bias"?

Why is the exponent calculated as E - 127 instead of using a standard signed integer? This is called "Biased Representation."

If we used a standard sign bit for the exponent, comparing two numbers would be computationally expensive because the computer would have to handle sign logic. By adding a bias (127), we ensure the stored exponent (E) is always a positive integer (from 1 to 254 for normalized numbers). This allows the hardware to compare the magnitudes of two floating-point numbers quickly, just like comparing regular integers.

A-Q2

Let n be a binary integer represented in two's complement. Which of the following is the operation that results in the value $9 \times n$ using only bit shifting and an addition or subtraction?

- a) Shift n 2 bits to the left, then add n to the result.
- b) Shift n 2 bits to the left, then subtract n from the result.
- c) Shift n 3 bits to the left, then add n to the result.
- d) Shift n 3 bits to the left, then subtract n from the result.

(Source:2025A,FE,Subject-A,Q2)

Answer: c

Explanation

In binary arithmetic, shifting a number to the left is equivalent to multiplying it by powers of 2.

1. Bit Shifting Rule:

- Shifting 1 bit to the left = $n \times 2^1 = 2n$
- Shifting 2 bits to the left = $n \times 2^2 = 4n$
- Shifting 3 bits to the left = $n \times 2^3 = 8n$

2. Target Calculation (9n):

- We want to achieve $9n$.
- Using the shift operation, we can easily get $8n$ by shifting n three bits to the left.
- To reach $9n$, we simply add the original value (n) to the result: $8n + n = 9n$.

Therefore, "Shift n 3 bits to the left ($8n$) and add n " is the correct operation.

Author's Comment

Computers are much faster at bit shifting and addition than they are at general-purpose multiplication. This is why compilers often optimize a simple multiplication like $x * 9$ into a shift and an addition behind the scenes. When you see a "multiplication" problem in the FE-Exam, always think: "How can I represent this multiplier as a sum of powers of 2?"

Column: Two's Complement and Shifting

Does this shifting rule work for negative numbers represented in two's complement? Yes, it does!

In **arithmetic left shifts**, as long as the sign bit doesn't change (which would indicate an overflow), shifting left effectively multiplies the value by 2, regardless of whether the number is positive or negative. For example, if $n = -2$, shifting it left by 3 bits would result in -16. Adding n (-2) to that gives -18, which is exactly $9 \times (-2)$. This efficiency is one of the reasons why two's complement is the standard for signed integer representation in modern computing.

A-Q3

Three friends and 4 other people are randomly seated in 7 seats arranged in a row. What is the probability that the 3 friends sit next to one another?

- a) $\frac{1}{168}$ b) $\frac{1}{42}$ c) $\frac{1}{35}$ d) $\frac{1}{7}$

(Source:2025A,FE,Subject-A,Q3)

Answer: d

Explanation

To find the probability, we first calculate the "total number of arrangements" and then the "number of arrangements where the 3 friends sit together."

1. **Total number of arrangements:**

- There are 7 people in total. The number of ways to seat them in a row is given by 7 factorial (7!).
- $7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5,040$.

2. **Number of arrangements where the 3 friends sit together:**

- **Step 1: Treat the 3 friends as a single block.** Now we have 5 units to arrange: this "1 block" plus the "4 other people." The number of ways to arrange these 5 units is 5!.
- **Step 2: Arrange the 3 friends within their block.** Even though they stay together, their relative order matters. The number of ways they can sit within the block is 3!.
- **Total specific cases:** $5! \times 3! = 120 \times 6 = 720$.

3. **Calculate the Probability:**

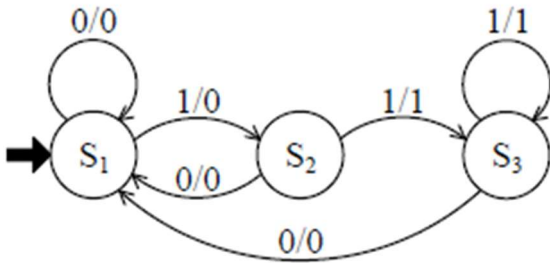
- Probability = $\frac{\text{Specific cases}}{\text{Total cases}} = \frac{5!3!}{7!}$
- Simplifying the expression: $\frac{5! \times (3 \times 2 \times 1)}{7 \times 6 \times 5} = \frac{6}{7 \times 6} = \frac{1}{7}$

Author's Comment

When calculating probabilities involving factorials, avoid multiplying large numbers immediately. As shown in Step 3, writing the formula as fractions and canceling out terms (like 5!) is much faster and reduces the risk of calculation errors. In the FE-Exam, time management is key!

A-Q4

For an automaton with input and output symbols $\{0, 1\}$ and a state transition diagram shown below, which of the following is the output string for the input string 0011001110? Here, S_1 represents the initial state, and the label x/y on each arc indicates that if x is an input, then y is the corresponding output at the state transition.



- a) 0001000110
- b) 0001001110
- c) 0010001000
- d) 0011111110

(Source:2025A,FE,Subject-A,Q4)

Answer: a

Explanation

To find the output, we must trace the state transitions starting from S_1 for each bit of the input string: 0011001110. The label x/y means: "Input is x / Output is y ".

Step	Input	Current State	Next State	Output
1	0	S1	S1	0
2	0	S1	S1	0
3	1	S1	S2	0
4	1	S2	S3	1
5	0	S3	S1	0
6	0	S1	S1	0
7	1	S1	S2	0
8	1	S2	S3	1
9	1	S3	S3	1
10	0	S3	S1	0

Connecting the outputs from each step: **0001000110**.

Author's Comment

This type of question tests your "**tracing ability**"— the skill to carefully follow a process one step at a time. It may seem simple, but this is a fundamental skill. When tracing, I recommend using your finger or a pen to point at the current state on the diagram as you go. Notice the pattern: this specific automaton only outputs 1 when it receives 1 while in state S_2 or S_3 . In S_1 , all inputs result in an output of 0.

Column: What is a Mealy Machine?

The automaton in this question is known as a **Mealy Machine**. In computer science, finite state machines (FSM) are categorized into two main types:

1. **Mealy Machine:** The output depends on both the **current state** and the **current input**. (In our diagram, the output is on the transition arc x/y).
2. **Moore Machine:** The output depends solely on the **current state**. (The output is usually written inside the state circle).

A-Q5

The table below shows the data in a list structure that has bidirectional pointers. In this table, a new Employee *G* is to be inserted between Employee *A* and Employee *K*. Which of the following contains only the pointers (among *a* through *f*) whose values change after the insertion?

Table

Address	EmployeeName	NextPointer	PreviousPointer
100	Employee <i>A</i>	300	0
200	Employee <i>T</i>	0	300
300	Employee <i>K</i>	200	100

Table after the addition

Address	EmployeeName	NextPointer	PreviousPointer
100	Employee <i>A</i>	<i>a</i>	<i>b</i>
200	Employee <i>T</i>	<i>c</i>	<i>d</i>
300	Employee <i>K</i>	<i>e</i>	<i>f</i>
400	Employee <i>G</i>	<i>x</i>	<i>y</i>

- a) *a, b, e, f*
- b) *a, e, f*
- c) *a, f*
- d) *b, e*

(Source:2025A,FE,Subject-A,Q5)

Answer: c

Explanation

A doubly linked list uses two pointers for each node: a **NextPointer** to the subsequent node and a **PreviousPointer** to the preceding node. Let's analyze the state before and after inserting Employee G between Employee A and Employee K.

1. Original State:

- Employee A (Addr: 100) points Forward to Employee K (Addr: 300).
- Employee K (Addr: 300) points Backward to Employee A (Addr: 100).
- Relationship: $A \Leftrightarrow K \Leftrightarrow T$

2. Insertion Requirements (G between A and K):

- **New sequence:** $A \Leftrightarrow G \Leftrightarrow K \Leftrightarrow T$
- Employee A must now point Forward to **G** (Addr: 400). → Value '**a**' changes from 300 to 400.
- Employee K must now point Backward to **G** (Addr: 400). → Value '**f**' changes from 100 to 400.

3. Evaluating Other Pointers:

- **b:** Employee A's PreviousPointer remains 0. (No change)
- **c & d:** Employee T's pointers (Next: 0, Prev: 300) do not change because the insertion happened earlier in the list.
- **e:** Employee K's NextPointer still points to Employee T (Addr: 200). (No change)

Therefore, only pointers '**a**' and '**f**' need to be updated in the existing table rows.

Author's Comment

When tackling pointer problems, don't just look at the numbers — visualize the "chains." I highly recommend drawing arrows between the employees on your scratch paper. In this problem, the "tracing ability" we discussed earlier is key. Trace the path from A to K: if you insert G in the middle, you must "break" the existing link and re-route the arrows. Notice that pointers *x* and *y* (for the new Employee G) will also have values, but the question specifically asks which *existing* pointers (*a* through *f*) change.

Column: Linked Lists vs. Arrays

Why do we bother with pointers and linked lists instead of just using a simple array? It comes down to **efficiency during modification**.

- **Array:** To insert an element in the middle, you have to "shift" every subsequent element one by one to make space. This is very slow for large datasets.
- **Linked List:** You don't need to move any data. You only need to update a few pointers (like 'a' and 'f' in this question) to point to the new memory address.

This makes linked lists ideal for systems where data is frequently added or removed, such as managing a queue of print jobs or implementing "Undo" functionality in software (where each action is a node in a bidirectional list).

This book is sample edition.

Notes and Disclaimers

Source Note:

The questions in this book are based on the past questions from the October 2025 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

Copyright © 2026 Takashi Narita

All rights reserved.

This book contains past exam questions from the **FE (Fundamentals of Information Technology Engineer) Examination** published by ITPEC, reproduced in accordance with ITPEC's public usage policy for educational purposes. The copyright of the original questions remains with ITPEC.

Source: <https://itpec.org/>