

基本情報技術者試験

変更のポイントと新試験攻略のための学習法

●基本情報技術者 (FE) 試験の位置づけ

情報処理技術者試験の枠組みを以下に記します。FE 試験は、知識・技能のレベルに応じて3段階になっている“情報処理技術者”向けの試験区分構造のうち、最も「基本的」な機能・技能を問うものになっています。



(IPA IT人材育成センターHP <https://www.ipa.go.jp/shiken/kubun/list.html> より)

典型的な人材イメージとしては「システム開発チームや、システム戦略チームのメンバー」に近いでしょう。皆さんがIT関連のキャリアを進めていく道筋の、ちょうど入り口に該当する資格といえます。

●従来の FE 試験

従来の(2022 年下期までの)基本情報技術者試験は、以下のような形で行われていました。

- ・実施時期

上期(春期)・下期(秋期)の年 2 回実施 (CBT 方式)

- ・出題形式

午前試験：四肢択一方式 80 問が出題，全 80 問を解答 (試験時間 150 分)

午後試験：多肢選択式 11 問が出題，うち 5 問を解答 (試験時間 150 分)

午前試験は，数行～1 ページ程度の分量からなる，比較的単純な四肢択一式の問題です。「〇〇の説明はどれか」や「〇〇の特徴として正しいものはどれか」といったように，基礎的な知識や，簡単な事例に対する解決能力が求められます。

午後試験は，事例を用いた長文形式の問題です。数ページ程度で文章や図表などが提示され，文章中の空欄を埋める設問や，適切な改善策を選ばせる設問など，いくつかの設問が並ぶ形態となります。

問番号	テーマ		配点	解答数・出題数
1	情報セキュリティ		20 点	必須（20 点）
2～5	ソフトウェア・ハードウェア	4 分野から 3 題を出題	各 15 点	2 問選択／4 問出題 (30 点)
	データベース			
	ネットワーク			
	ソフトウェア設計			
	マネジメント・ストラテジ			
6	データ構造とアルゴリズム		25 点	必須（25 点）
7～11	C		各 25 点	1 問選択／5 問出題 (25 点)
	Java			
	Python			
	アセンブラ (CASL II)			
	表計算			

●これからの FE 試験

2023 年 4 月から、基本情報技術者試験は実施方式や出題形式を大きく変えて行われます。概要は以下のとおりです。

・実施方式、採点方式

決められた日時・期間だけではなく、受験者が日時を選んで随時受験できるよう“通年試験化”されます。

採点は、IRT (Item Response Theory : 項目応答理論) に基づく方式に変更されます。

・出題形式

小問形式への変更による午後問題のコンパクト化、出題数・解答数の変更などによって、試験時間が 30～40%短縮されます。具体的な内訳は次のようになります。

試験区分	変更前		変更後	
FE	午前試験 (小問)	試験時間：150分 出題数：80問 解答数：80問	科目A試験 (小問)	試験時間：90分 出題数：60問 解答数：60問
	午後試験 (大問)	試験時間：150分 出題数：11問 解答数：5問 ※選択問題あり	科目B試験 <u>(小問)</u>	試験時間：100分 出題数：20問 解答数：20問 <u>※選択問題なし（全問必須）</u>

(IPA Web サイトより https://www.jitec.ipa.go.jp/1_00topic/topic_20220425.html)

出題範囲については以下のような情報が提示されています。

試験区分	科目	出題範囲の変更概要
FE	科目A試験	現在の午前試験に準じます。
	科目B試験	これまで必須解答としていた「情報セキュリティ」と「データ構造及びアルゴリズム（擬似言語）」の二つの分野を中心にした構成に変更します。 また、個別プログラム言語（C、Java、Python、アセンブラ言語、表計算ソフト）による出題は、普遍的・本質的なプログラミング的思考力を問う擬似言語による出題に統一します。

(IPA Web サイトより https://www.jitec.ipa.go.jp/1_00topic/topic_20220425.html)

●科目 A 試験(旧：午前試験)の変更ポイント

科目 A 試験については，出題範囲は“現在の午前試験に準ずる”とされていますので，大きな変化はないでしょう。従来の午前試験と同様，

テクノロジ系(理系)：基礎理論，コンピュータシステム(ハードウェア，ソフトウェア，システム構成など)，技術要素(データベース，ネットワーク，セキュリティ)，システム開発

マネジメント系(文系)：プロジェクトマネジメント，サービスマネジメント，システム監査

ストラテジ系(文系)：システム戦略，経営戦略，企業活動(会計を含む)，法務(著作権や派遣法など)

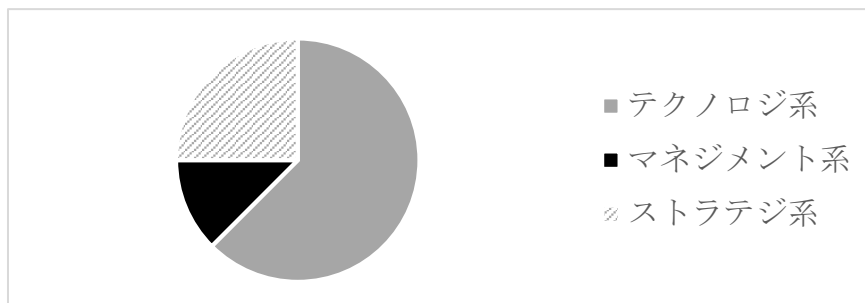
といった分野から，それぞれ基本的な用語知識などが問われると思われます。従来の午前試験では，計 80 問のうち

テクノロジ系：約 50 問 マネジメント系：約 10 問 ストラテジ系：約 20 問

という内訳でしたので，変更後の計 60 問でもその割合が続くと仮定すると，

テクノロジ系：37～38 問 マネジメント系：7～8 問 ストラテジ系：15 問

程度の問題数になります。変更に伴ってこの割合が変化する可能性もありますが，現時点ではおおよそこのような想定で対策に臨むとよいでしょう。



加えて注目したいのは，1 問当たりの解答時間が変化することです。従来は 150 分で 80 問，変更後は 90 分で 60 問となるので，

従来： $150 \div 80 = 1.875$ 分/1 問

変更後： $90 \div 60 = 1.5$ 分/1 問

となり，1 問あたりに費やすことのできる時間は，むしろ減少します。もしかすると，従来の試験よりも手間のかかる計算問題や事例問題の割合が減って，短い時間で答えられる単純な問題の比率が多くなるのかもしれませんが，しかし，従来と同様の難易度レベルを保ったまま，1 問当たりの時間が短くなるという可能性もあります。学習を進めるにあたっては，油断せず「もし従来と同様の問題構成が出てきても，ちゃんと制限時間内に答えられる」ように，スピード感を磨いておく必要があるでしょう。

●科目 B 試験(旧：午後試験)の変更ポイント

・20 問／100 分になり，コンパクト化

従来の午後試験は一つ一つの問題のボリュームが大きい，「大問」形式でした。問題のページ数は多くが 3 ページ以上であり，内容によっては 7～8 ページになることもありました。

これに対し，変更後の科目 B 試験は「小問」形式が 20 問並ぶ形式となります。IPA が公開しているサンプル問題を見る限りでは，1 問のボリュームは 1～2 ページ程度で，従来の午前問題を少し拡張したような印象を受けます。

問2 次のプログラム中の と に入れる正しい答えの組合せを，解答群の中から選べ。ここで，配列の要素番号は 1 から始まる。

次のプログラムは，整数型の配列 array の要素の並びを逆順にする。

〔プログラム〕

```
整数型の配列: array ← {1, 2, 3, 4, 5}
整数型: right, left
整数型: tmp

for (left を 1 から (arrayの要素数 ÷ 2 の商) まで 1 ずつ増やす)
  right ← 
  tmp ← array[right]
  array[right] ← array[left]
   ← tmp
endfor
```

解答群

	a	b
ア	array の要素数 - left	array[left]
イ	array の要素数 - left	array[right]
ウ	array の要素数 - left + 1	array[left]
エ	array の要素数 - left + 1	array[right]

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

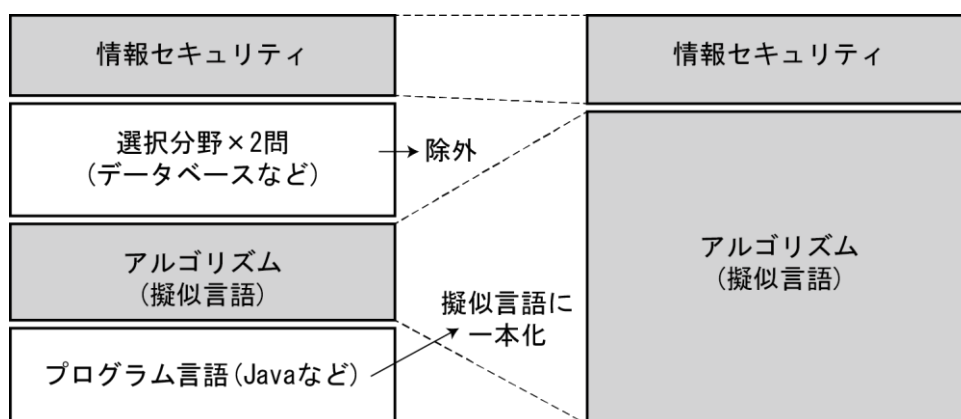
単純計算すると，1 問当たり 100 分÷20 問＝5 分で解くスピードが求められることになります。サンプル問題の内容を考慮すると，5 分あればある程度余裕をもって解ける人が多いと思われますが，実際の試験において各問のボリューム・難易度がサンプル問題と異なってくる可能性もあるので，油断は禁物と言えるでしょう。

・出題範囲が「アルゴリズム＋情報セキュリティ」に特化

従来の午後試験では、情報セキュリティとアルゴリズム(擬似言語)の必須問題のほかに、

- ・データベースなど、各分野の問題から2問
- ・Java や表計算など、プログラム言語の問題から1問

を選択し、合計で大問を5問解く形になっていました。これが変更後の科目B試験では、アルゴリズムと情報セキュリティの2分野に集約されます。全20問中、約8割がアルゴリズム、2割が情報セキュリティの問題になる予定です。データベースなどの分野は姿を消し、Javaなどの言語の問題は「擬似言語による問題で十分にプログラミングのスキルは判定可能」ということで、擬似言語を用いた出題に一本化される形になります。



図を見てもわかるように、科目B試験では「擬似言語で記述されたプログラムを読み解く力」が非常に重要となることがわかります。

従来の午後試験では、アルゴリズムが苦手な人でも、データベースやマネジメント系などの選択問題で高得点を取れば、総合的に合格点を得ることが可能でした。変更後はそのようなアプローチは不可能となり、アルゴリズムが苦手だからといって避けては通れない試験になったといえるでしょう。

また、従来は情報セキュリティもアルゴリズムも大問が一つという出題でしたが、変更後は小規模な問題がいくつも出る形になるので、「たまたま知っていた知識だったので運よく高得点が取れた」や「たまたま知らないジャンルだったので、実力よりも点数が出なかった」ということが少なくなり、「基礎的なことを漏れなくマスターできている」人がしっかりその力に応じた点数を取れる試験になると思われます。

・科目 B のサンプル問題

では実際に、さきほど紹介したものも含め、IPA が公開しているサンプル問題を紹介してみしましょう。

[アルゴリズムとプログラミング(擬似言語)]

問 1 次のプログラム中の に入れる正しい答えを、解答群の中から選べ。

ある施設の入場料は、0 歳から 3 歳までは 100 円、4 歳から 9 歳までは 300 円、10 歳以上は 500 円である。関数 fee は、年齢を表す 0 以上の整数を引数として受け取り、入場料を返す。

[プログラム]

```
○整数型: fee(整数型: age)
  整数型: ret
  if (age が 3 以下)
    ret ← 100
  elseif (  )
    ret ← 300
  else
    ret ← 500
  endif
  return ret
```

解答群

- ア (age が 4 以上) and (age が 9 より小さい)
- イ (age が 4 と等しい) or (age が 9 と等しい)
- ウ (age が 4 より大きい) and (age が 9 以下)
- エ age が 4 以上
- オ age が 4 より大きい
- カ age が 9 以下
- キ age が 9 より小さい

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

if と else の場合分けや、and／or を用いた複合条件など、条件分岐に関する基礎的な理解を問う問題です。これまで午後試験対策でアルゴリズムを学習してきた人であれば、容易に感じる内容でしょう。

問2 次のプログラム中の と に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

次のプログラムは、整数型の配列arrayの要素の並びを逆順にする。

〔プログラム〕

```
整数型の配列: array ← {1, 2, 3, 4, 5}
整数型: right, left
整数型: tmp

for (left を 1 から (arrayの要素数 ÷ 2 の商) まで 1 ずつ増やす)
  right ← 
  tmp ← array[right]
  array[right] ← array[left]
   ← tmp
endfor
```

解答群

	a	b
ア	array の要素数 - left	array[left]
イ	array の要素数 - left	array[right]
ウ	array の要素数 - left + 1	array[left]
エ	array の要素数 - left + 1	array[right]

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

配列要素を順に処理していく、典型的なループ(繰返し)制御のプログラムです。これも問1同様、従来の午後試験対策学習の観点から見れば、ごく基礎的なレベルに入ります。

問3 次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。

手続 append は、引数で与えられた文字を単方向リストに追加する手続である。単方向リストの各要素は、クラス ListElement を用いて表現する。クラス ListElement の説明を図に示す。ListElement 型の変数はクラス ListElement のインスタンスの参照を格納するものとする。大域変数 listHead は、単方向リストの先頭の要素の参照を格納する。リストが空のときは、listHead は未定義である。

メンバ変数	型	説明
val	文字型	リストに格納する文字。
next	ListElement	リストの次の文字を保持するインスタンスの参照。初期状態は未定義である。

コンストラクタ	説明
ListElement(文字型: qVal)	引数 qVal でメンバ変数 val を初期化する。

図 クラス ListElement の説明

〔プログラム〕

大域: ListElement: listHead ← 未定義の値

```

○append(文字型: qVal)
  ListElement: prev, curr
  curr ← ListElement(qVal)
  if (listHead が a)
    listHead ← curr
  else
    prev ← listHead
    while (prev.next が 未定義でない)
      prev ← prev.next
    endwhile
    prev.next ← b
  endif

```

解答群

	a	b
ア	未定義	curr
イ	未定義	curr.next
ウ	未定義	listHead
エ	未定義でない	curr
オ	未定義でない	curr.next
カ	未定義でない	listHead

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

オブジェクト指向を取り入れたリスト構造操作のプログラムです。リスト構造は基本的なデータ構造の一つであり、この問題で行っている処理も典型的なものですが、今まで擬似言語になかったオブジェクト指向の概念が用いられているため、文法面でやや解きづらさを感じる人も多いかと思われます。

問4 次の記述中の a ～ c に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

要素の多くが0の行列を疎行列という。次のプログラムは、二次元配列に格納された行列のデータ量を削減するために、疎行列の格納に適したデータ構造に変換する。

関数 transformSparseMatrix は、引数 matrix で二次元配列として与えられた行列を、整数型配列の配列に変換して返す。関数 transformSparseMatrix を transformSparseMatrix({{3, 0, 0, 0, 0}, {0, 2, 2, 0, 0}, {0, 0, 0, 1, 3}, {0, 0, 0, 2, 0}, {0, 0, 0, 0, 1}})として呼び出したときの戻り値は、{{ a }, { b }, { c }}である。

〔プログラム〕

```
○整数型配列の配列: transformSparseMatrix(整数型の二次元配列: matrix)
  整数型: i, j
  整数型配列の配列: sparseMatrix
  sparseMatrix ← {{}, {}, {}} /* 要素数0の配列を三つ要素にもつ配列 */
  for (i を 1 から matrixの行数 まで 1 ずつ増やす)
    for (j を 1 から matrixの列数 まで 1 ずつ増やす)
      if (matrix[i, j] が 0 でない)
        sparseMatrix[1]の末尾 に iの値 を追加する
        sparseMatrix[2]の末尾 に jの値 を追加する
        sparseMatrix[3]の末尾 に matrix[i, j]の値 を追加する
      endif
    endfor
  endfor
  return sparseMatrix
```

解答群

	a	b	c
ア	1, 2, 2, 3, 3, 4, 5	1, 2, 3, 4, 5, 4, 5	3, 2, 2, 1, 2, 3, 1
イ	1, 2, 2, 3, 3, 4, 5	1, 2, 3, 4, 5, 4, 5	3, 2, 2, 1, 3, 2, 1
ウ	1, 2, 3, 4, 5, 4, 5	1, 2, 2, 3, 3, 4, 5	3, 2, 2, 1, 2, 3, 1
エ	1, 2, 3, 4, 5, 4, 5	1, 2, 2, 3, 3, 4, 5	3, 2, 2, 1, 3, 2, 1

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

2次元配列(配列の配列)を用いて、多数の数値データを一定の規則に従い処理していくプログラムの問題です。従来の午後試験と同様に、プログラムの空欄埋めだけでなく、処理結果を考える(追跡する)内容も出題されることが示されているといえるでしょう。時間がかかりやすいタイプの問題ですので、要注意といえます。

また、行列データの処理は AI や機械学習で重要となるテーマであり、今後の試験でも、同様に「多数のデータをルールに沿って変換や集計する」といったアルゴリズムは出題頻度が高くなるものと推測されます。

問5 次のプログラム中の に入れる正しい答えを、解答群の中から選べ。

任意の異なる2文字を c_1 , c_2 とするとき、英単語群に含まれる英単語において、 c_1 の次に c_2 が出現する割合を求めるプログラムである。英単語は、英小文字だけから成る。英単語の末尾の文字が c_1 である場合、その箇所は割合の計算に含めない。例えば、図に示す4語の英単語 “importance”, “inflation”, “information”, “innovation” から成る英単語群において、 c_1 を “n”, c_2 を “f” とする。英単語の末尾の文字以外に “n” は五つあり、そのうち次の文字が “f” であるものは二つである。したがって、求める割合は、 $2 \div 5 = 0.4$ である。 c_1 と c_2 の並びが一度も出現しない場合、 c_1 の出現回数によらず割合を 0 と定義する。

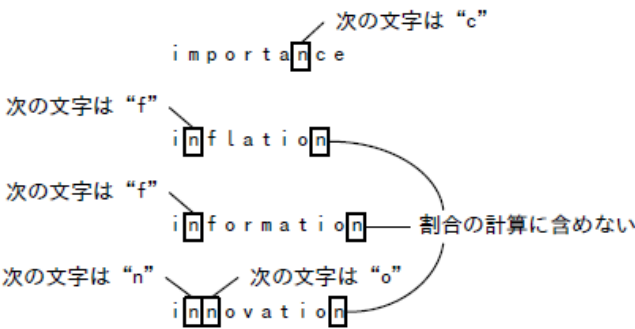


図 4 語から成る英単語群の例

プログラムにおいて、英単語群は Words 型の大域変数 words に格納されている。クラス Words のメソッドの説明を、表に示す。本問において、文字列に対する演算子 “+” は、文字列の連結を表す。また、整数に対する演算子 “÷” は、実数として計算する。

表 クラス Words のメソッドの説明

メソッド	戻り値	説明
freq(文字列型: str)	整数型	英単語群中の文字列 str の出現回数を返す。
freqE(文字列型: str)	整数型	英単語群の中で、文字列 str で終わる英単語の数を返す。

〔プログラム〕

```
大域: Words: words /* 英単語群が格納されている */
```

```
/* c1の次にc2が出現する割合を返す */
```

```
○実数型: prob(文字型: c1, 文字型: c2)
```

```
文字列型: s1 ← c1の1文字だけから成る文字列
```

```
文字列型: s2 ← c2の1文字だけから成る文字列
```

```
if (words.freq(s1 + s2) が 0 より大きい)
```

```
    return 
```

```
else
```

```
    return 0
```

```
endif
```

解答群

ア $(\text{words.freq}(s1) - \text{words.freqE}(s1)) \div \text{words.freq}(s1 + s2)$

イ $(\text{words.freq}(s2) - \text{words.freqE}(s2)) \div \text{words.freq}(s1 + s2)$

ウ $\text{words.freq}(s1 + s2) \div (\text{words.freq}(s1) - \text{words.freqE}(s1))$

エ $\text{words.freq}(s1 + s2) \div (\text{words.freq}(s2) - \text{words.freqE}(s2))$

(出典：基本情報技術者試験 科目 B 試験サンプル問題)

文字列操作のアルゴリズムに関する問題です。文字列操作は従来から重要テーマの一つでした。この問題では、文字列を表すクラスを用意し、メソッド呼出しによって便利な機能を実現する仕様になっています。

〔情報セキュリティ〕

問6 製造業のA社では、ECサイト（以下、A社のECサイトをAサイトという）を使用し、個人向けの製品販売を行っている。Aサイトは、A社の製品やサービスが検索可能で、ログイン機能を有しており、あらかじめAサイトに利用登録した個人（以下、会員という）の氏名やメールアドレスといった情報（以下、会員情報という）を管理している。Aサイトは、B社のPaaSで稼働しており、PaaS上のDBMSとアプリケーションサーバを利用している。

A社は、Aサイトの開発、運用をC社に委託している。A社とC社との間の委託契約では、Webアプリケーションプログラムの脆弱性対策は、C社が実施するとしている。

最近、A社の同業他社が運営しているWebサイトで脆弱性が悪用され、個人情報が漏えいするという事件が発生した。そこでA社は、セキュリティ診断サービスを行っているD社に、Aサイトの脆弱性診断を依頼した。脆弱性診断の結果、対策が必要なセキュリティ上の脆弱性が複数指摘された。図1にD社からの指摘事項を示す。

- (一) Aサイトで利用しているDBMSに既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。

(二) Aサイトで利用しているアプリケーションサーバのOSに既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。

(三) ログイン機能に脆弱性があり、Aサイトのデータベースに蓄積された情報のうち、会員には非公開の情報を閲覧されるおそれがある。

図1 D社からの指摘事項

設問 図1中の項番(一)～(三)それぞれに対処する組織の適切な組合せを、解答群の中から選べ。

解答群

	(一)	(二)	(三)
ア	A社	A社	A社
イ	A社	A社	C社
ウ	A社	B社	B社
エ	B社	B社	B社
オ	B社	B社	C社
カ	B社	C社	B社
キ	B社	C社	C社
ク	C社	B社	B社
ケ	C社	B社	C社
コ	C社	C社	B社

(出典：基本情報技術者試験 科目B 試験サンプル問題)

情報セキュリティ分野のサンプルです。技術的な観点よりはむしろ、組織におけるセキュリティマネジメントの色合いが強い内容になっています。今後に変化する可能性もありますが、現状ではこういったマネジメント的な問題が主流で構成されると予想して取り組むのがよさそうです。

●変更後の試験の学習ポイント

・科目 A 対策は、従来の午前試験対策と同様に

科目 A 試験は出題数と試験時間に変更になりましたが、内容は従来の午前試験に準ずる形になるので、対策学習も従来の午前試験対策のアプローチがそのまま使えます。

インプット：テキスト(教科書)を用いた各分野の基礎知識の習得

アウトプット：問題演習で学習の成果を確認、弱点を把握

という二つの学習サイクルを、

インプット → アウトプット → 再インプット(再学習) → …

という形で回していき、広く知識を身に付けましょう。

アウトプットの際には、過去の基本情報技術者の午前問題も、そのまま材料として活用できるでしょう。

・科目 B 対策は、2 分野に集中して演習(ケーススタディ)の積み重ね

従来と異なり、情報セキュリティとアルゴリズム(擬似言語)のみの分野構成となるので、それらに集中した対策学習を行うことになります。問題数やサンプル問題のバリエーションをみてもいろいろなタイプの問題が幅広く出るであろうことは確実ですので、

「整列や探索は十分理解しているが、文字列操作アルゴリズムはよく知らない」

「リスク分析のところはかなり勉強したが、利用者認証のことは分からない」

といったような、理解の範囲に偏りを生じさせないように心がけましょう。

そのためには、できるだけ沢山のプログラムやセキュリティ事例に触れておくことが大事です。サンプル問題や TAC での問題演習などを介して、さまざまなプログラム、多様なシチュエーションを目にしておくことで、本番でのリスクを下げるすることができます。

また、昨今ではインターネットや書籍で、様々なアルゴリズムや情報セキュリティ事例に関する情報が入手できます。問題演習の形となっていなくとも、それらに広く目を通すことで、より漏れのないスキルを身に付けることが期待できます。単純に知識を暗記するというよりは、「こういう目的の場合はこのような手順が有効だ」といったように、ケーススタディという意識で学習を進めるのがよいと思われます。

過去の基本情報技術者の午後問題はボリュームが大きいため、そのままでは科目 B 用のアウトプット(問題演習)の材料としては使いづらいかもしれません。ただしプログラムの基本的な考え方や、部分部分における処理の工夫、セキュリティ事例における論点

などは、そのまま独立して科目 B の題材となることもあり得ます。参考資料として目を通して(時間に余裕があれば挑戦して)みるのも、一つの有効な対策学習法といえます。

出題数の割合を考えれば、

- ・まず、アルゴリズム(擬似言語)の対策学習をしっかりと
- ・その上で、情報セキュリティについても学習

という形で、アルゴリズムに重点をおいて科目 B 試験対策を進めるのがよいでしょう。学習時間についても、実際の出題数割合と同様の 8 割 : 2 割程度、あるいはそれ以上の重み付けでアルゴリズム対策に重点をおく、といった心構えでよいかと思われます。

●コースへのご案内

これまで述べたような試験対策学習を進めるにあたっては、科目 A、科目 B ごとにしっかりとスケジュールを立て、ペースを守って漏れなく学習していくことが重要になりますので、独学でゼロから始めるのはなかなか難しいかと考えます。特に科目 B 試験については未知の部分も多いため、どう学習してよいか分からない部分も多いでしょう。

TAC は今までの情報の蓄積と講師の経験を活用し、新しい試験に対しても最善の対策となるようなカリキュラムをご提供します。講座をご利用いただくことで、前述のような不安を廃し、スムーズに学習を進めることができますでしょう。

また、講義や教材で分かりづらいところがあれば、そのつど講師に質問するという「コミュニケーションしながらの学習」ができるというのも、講座受講の強みです。

ぜひご利用をご検討ください。皆様のご参加を心よりお待ちしております！