

基本情報技術者 公開セミナー「午後試験対策学習法」

1. 基本情報技術者試験の性格と概要

試験の概要

情報処理技術者試験は、「情報処理の促進に関する法律」に基づき経済産業省が、情報処理技術者としての「知識・技能」の水準がある程度以上であることを認定している国家試験です。

情報システムを構築・運用する「技術者」から情報システムを利用する「エンドユーザ（利用者）」まで、IT に関係するすべての人に活用いただける試験として実施しています。特定の製品やソフトウェアに関する試験ではなく、情報技術の背景として知るべき原理や基礎となる技能について、幅広い知識を総合的に評価しています。

試験の目的

- 情報処理技術者に目標を示し、刺激を与えることによって、その技術の向上に資すること。
- 情報処理技術者として備えるべき能力についての水準を示すことにより、学校教育、職業教育、企業内教育等における教育の水準の確保に資すること。
- 情報技術を利用する企業、官庁などが情報処理技術者の採用を行う際に役立つよう客観的な評価の尺度を提供し、これを通じて情報処理技術者の社会的地位の確立を図ること。

注) 上記は <http://www.jitec.ipa.go.jp> からの引用です。

2. 評価と資格取得のメリット

対外的な評価

“システムの仕事に役立つ資格”を調査した「2012年版いる資格、いない資格」(ITpro 掲載)によると、基本情報技術者はITベンダーの技術者が取得すべき資格の3位に(1位はプロジェクトマネージャ)、ITベンダーの営業担当者では2位に(1位はITパスポート)、また、ユーザ企業のシステム担当者が取得すべき資格の第1位にランキングされています。ITベンダーの人事担当者の多くが、技術者はもちろんのこと、営業担当者にも基本情報技術者を取得させたいと回答しており、システムを発注する側のユーザ企業においても、取得の必要性が高い資格であることがうかがえます。

資格取得のメリット

- (1) 情報処理技術者としてのスキルを有していることが公的に証明される。
- (2) IT業界で活躍する方が、現在どのレベルの専門知識を有するかが立証される。
- (3) 上級資格受験のための基礎を習得できる。
- (4) 就職活動を行う上でのアピール材料となる。
- (5) 資格手当や一時金など報奨金制度、昇級条件、(学生さんなら)履修の単位や特待生など待遇面で優遇

3. 基本情報技術者試験の出題と形式 ～何が問われるか～

(1) 午前試験

試験時間：2時間 30分

出題形式：マークシートで四択 80 問，全問必須(1.25 点×80＝100 点)

合格基準：100 点満点の 60% 以上の正解で合格

出題比率

テクノロジー	マネジメント	ストラテジ
50	: 10	: 20

(2) 午後試験

試験時間：2時間 30分

出題形式：マークシートで多肢選択式(複数個の選択肢から 1 個以上を選択)

13 問中 7 問を解答(次表を参照)

合格基準：100 点満点の 60% 以上の正解で合格

出題形式：長文問題形式

出題内容

問番号	テーマ		配点	解答数・出題数
1～4	ハードウェア	5分野から 4問を出題	各12点	5問選択/7問出題 (12×5＝60点)
	ソフトウェア			
	データベース			
	ネットワーク			
	情報セキュリティ			
5	ソフトウェア設計			
6	マネジメント			
7	ストラテジ			
8	データ構造とアルゴリズム		20点	必須 (20×1＝20点)
9	C		各20点	1問選択/5問出題 (20×1＝20点)
10	COBOL			
11	Java			
12	アセンブラ			
13	表計算			

4. 平成 23 年秋期の午後試験の出題内容・傾向

(1) 出題内容

問番号	分野	テーマ	難易度
1	ハードウェア	A/D 変換	難
2	データベース	従業員データベースの 設計と運用	易
3	ネットワーク	ネットワークの構築	易
4	情報セキュリティ	情報セキュリティに おけるリスク	標準
5	ソフトウェア設計	書籍の受注業者の 受注システム	易
6	プロジェクトマネジメント	プロジェクトの要員計画	難
7	システム戦略	販売ルート別の売上 及び市場の分析	標準
8	データ構造と アルゴリズム	代入文の処理	標準
9	C	循環小数の出力	標準
10	COBOL	受験申込ファイルの 集計及び整列	易
11	Java	住所録管理プログラム	標準
12	アセンブラ	除算と 2 進 10 進数文字列変換	標準
13	表計算	社員の基本給及び賞与計算	標準

(2) 出題傾向

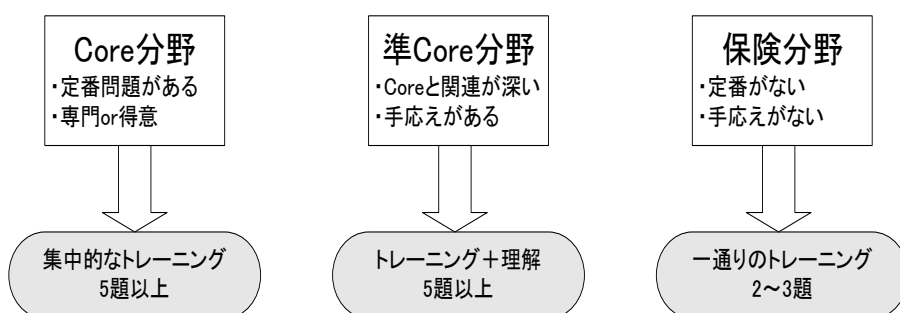
23 年秋の午後試験は、選択問題のブロックで難易度にやや差がありました。中でも問 1、問 6 が難しかった反面、テクノロジー系では前回(23 春)よりも手間のかかる問題が減ったため、いかに柔軟に対応できたかが鍵になります。必須問題(問 8)のアルゴリズムについては、問題文がやや難解ですが、最大値アルゴリズムなどが使われており難易度は標準でした。プログラム言語では、COBOL を除き、他のプログラム言語で難易度の差はそれほど大きくはありません。ただし、表計算では、試験前に発表されていた「表計算の機能・用語」の改訂により、「関数」「マクロ」など、新しい仕様で出題が行われました。今後は表計算でも「マクロ」によるロジックが問われることになるといえます。

午後全体でみると、前回よりもバランスが良く、基本情報技術者として理想的な難易度であったといえます。

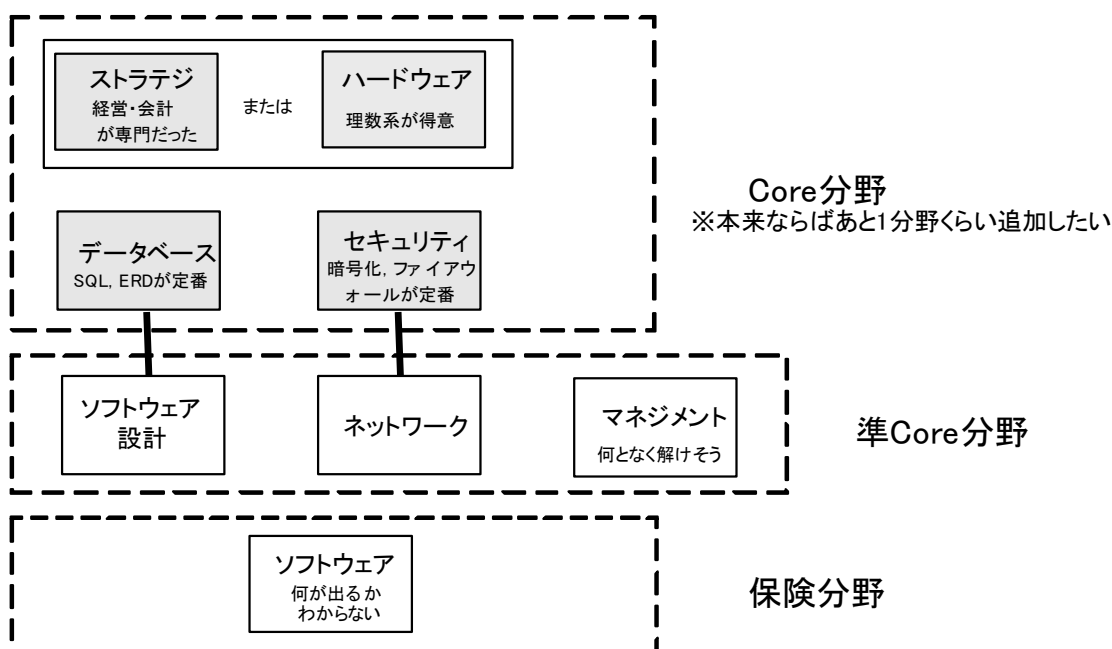
5. 午後対策 分野別学習法

(1) 午後選択問題対策

- ・トレーニング中心。間違えてしまった設問は解説で理解する。
場合によってはテキストに戻って復習し直す。
- ・原則として苦手な分野は選ばない。(午前と同様)分野にメリハリを付ける。
- ・Core 分野, 準 Core 分野, 保険分野に分ける
 - Core 分野は強力な得点源にする。
 - 準 Core 分野は Core と同じくらいに育てる
 - 保険分野はそのもの保険, 余力を入れない

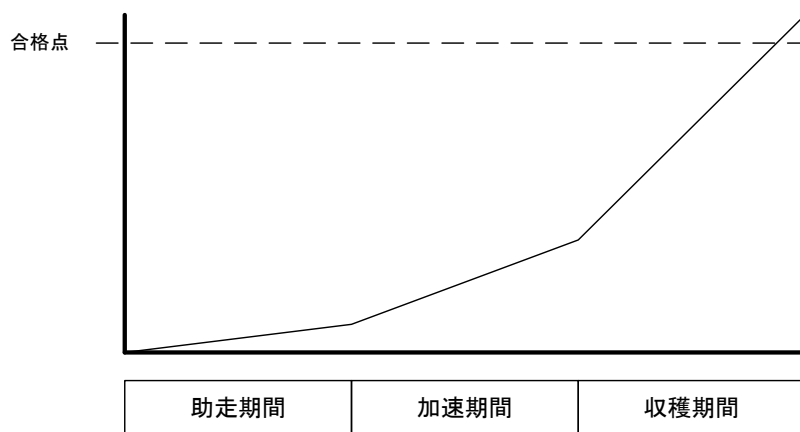


【選択分野の例】



(2) 午後 アルゴリズム系の学習期間

- ・問題を解くための「スキル」が必要な分野(アルゴリズム, プログラム言語)
- ・長期的な学習でスキルを高める
 - スキルを合格レベル(50~80%)にあげるには, 3ヶ月~(場合によっては)6ヶ月近く必要



★助走期間(0.5~2ヶ月)

- ・とにかく, プログラムの全体像や流れをつかむようにする。
この期間に集中して文法を覚えようとしても, 難しく感じる部分が多くなってしまう。
- ・わからない部分を気にするよりも, わかるところをつまみ食いする。
- ・アルゴリズムのテキストを眺めてみる。アルゴリズムの読解を「試みる」。

★加速期間(1~2ヶ月)

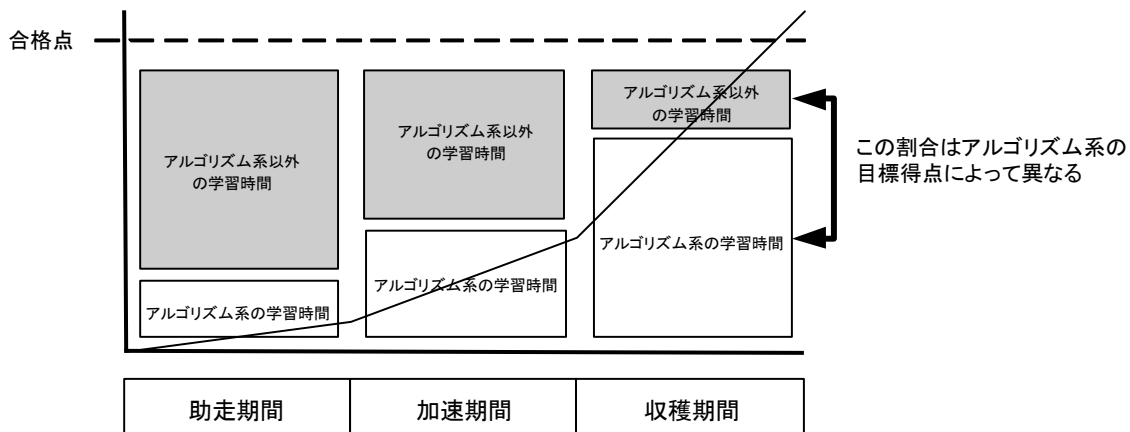
- ・テキストをしっかりと読んで, 疑問点をつぶしてゆく。
→ 文法や記述法などを確実に覚えてゆく。
- ・わからなければ, 誰か(講師, 先輩, 友達, etc)に聞く。
→ それでもわからなければ先に進む(先に進むことで, 後から解決することもある)。

★収穫期間(1~2ヶ月)

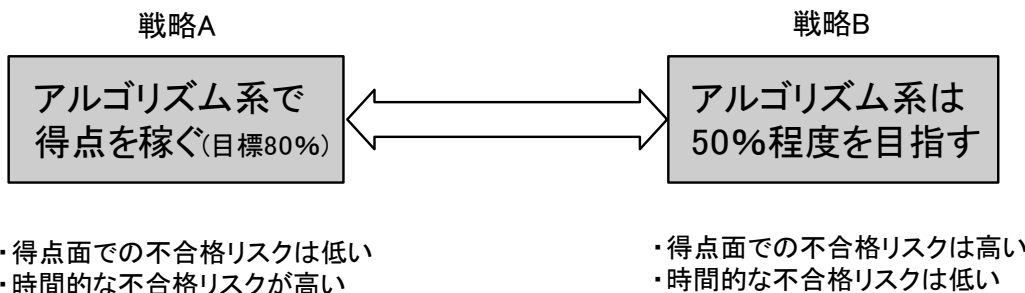
- ・過去問題 or 問題集などに収録された問題をとにかく解く。
- ・わからないところは, 解説などを読んで理解する。
→ とにかく「理解する」まではプログラムに取り組む。
- ・初めのうちは実際の解答時間(1題あたり 20~25分)にはこだわらない。
- ・終盤では解答時間を意識したトレーニングを行う。
- ・この期間ではできるだけプログラミングに集中する。
→ 他分野の学習は気晴らし or 実力維持トレーニング程度。

6. 午後対策の学習スケジュール

午後対策の学習スケジュールは、アルゴリズム系の対策が鍵！



(計画しておきたいこと)



- ・午前対策は「今すぐ」始める → アルゴリズム系の助走期間が最大のチャンス
- ・午前対策を一通り終えてから午後選択問題トレーニングに移った方が、成功率は高い
- ・アルゴリズム系を「50%でよい」とするならば、アルゴリズム系の学習時間を減らし、その分、選択分野の学習時間に充てる。

7. 午後の解法テクニックを伝授

～ アルゴリズム問題の攻略法 ～

「アルゴリズム(擬似言語)」は、基本情報技術者試験の午後問 8 に出題される必須問題です。難易度、配点ともに高く、午後試験をクリアする上での「関門」といってもよいでしょう。アルゴリズム問題を苦手とする多くの方が「どのように解けばよいのかわからない」と感じておられるようです。本セミナーでは、TAC 講師によるアルゴリズム問題を解くときのアプローチや解答テクニックを一部公開したものです。もちろん、“これらがすべてではありません”が、アルゴリズム問題が苦手という方は、ここにご紹介する方法を試してみることを、ぜひお勧めします。

(1) 擬似言語のアプローチ

擬似言語を解くとき、講師などの経験者は、一体どのようなアプローチをとっているのでしょうか。実は、経験者がとるアプローチは 1 つです。それは「分割して考える」ことなのです。

プログラム全体を一度に考えようとすると、プログラム読解が難しく感じるでしょう。プログラムをいくつかのブロックに分割してみると、それぞれのブロック単位では、比較的構造が単純であることに気がつきます。プログラムを分割することにより、読解を容易にできるのです。これを積み上げていくことで、プログラム全体を把握でき、問題を解くのに要する時間を短縮できる場合が多いのです。

(2) ブロック分割

プログラムは分割して考える、とはいえ、でたらめに分割しては効果がありません。意味のあるブロックに分割することが重要なのです。慣れないうちは、プログラムの制御構造に従って分割すればよいでしょう。(午後対策講義・演習では、より具体的なテクニックをお教えします)

(3) 変数の役割を明らかにする

プログラムで用いる変数の役割を明らかにしましょう。ブロック内だけで明らかにならない場合は、プログラム全体に視点を広げましょう。プログラムからその変数を用いている命令を抜き出せば、だいたいの想像がつくはずです。

特に変数の多いプログラムでは、多少時間がかかっても、変数の役割をまとめておくことが必要です。また、添字などは図示するとよいでしょう。

(4) トレースは最小限に抑える

アルゴリズムを理解するためには「トレース」がかかせません。丁寧にトレースすれば、プログラムの動作が理解できるのです。ただし、本試験に限っては、トレースは

「必要最小限」に抑えなければなりません。なぜなら、トレースは時間のかかる作業だからです。ここでは、思い切って、

“練習ではベタトレース推奨、ただし、本試験ではトレースは必要最小限”
と言いたいと思います。こうすることで、本試験でのトレースによる時間のロスを防ぎ、効率的に解くことができるようになります。(具体的なテクニックについては、講義・演習でお教えいたします)

(5) 基本部品を蓄積する

プログラムの定型的な処理パターン（基本部品）を覚えておきましょう。基本部品を蓄積しておくことで、同じパターンを使ったアルゴリズム問題が出題された場合に時間を大幅に節約することができます。基本アルゴリズムやデータ構造についてはしっかり理解しておきましょう。

8. TAC の午後対策 アルゴリズム学習の流れ

● 講義

アルゴリズムの知識習得には時間がかかります。なるべく、早い時期から体系的に学習していくことが大切です。

- ・ アルゴリズム・データ構造の知識を習得。
- ・ 基本問題の演習（問題を解くことで知識のモレ・弱点を確認）を行う。
- ・ 弱点箇所、未履修のテーマを必ず再確認する → 苦手分野が少なくなる

「インプット → アウトプット → 再インプット」

の学習サイクルを繰り返す。

● 演習

午後対策の専用教材を使って、答案練習を行います。

- ・ 広い範囲の過去・模擬問題の演習
- ・ 擬似言語・プログラム言語など午後の応用演習
- ・ 選択問題対策

} TAC式 解法テクニック

● 公開模試で腕試し

試験直前期の実力診断・弱点分野の把握に最適です。

※ 基本情報技術者の午後試験では「プログラミング言語」が出題されます。
 ご自身の学習目的にあったプログラミング言語を選択しましょう。

【プログラミング言語の種類・特徴】

名 称	特 徴	学習時間	過去の難易度
C 言語	システム記述からアプリケーション作成まで広く用いられる開発者向きの言語です。	やや多め	標準～やや難
COBOL	英文形式で命令を記述するため、理解しやすい事務処理用の言語ですが、現在ではやや古さを感じる面もあります。	普通	やや易～標準
Java	最新のオブジェクト指向型言語で広く利用されており、人気の高い言語です。多機能な反面、学習内容は多くなります。	多め	標準～やや難
アセンブラ CASL II	試験用のアセンブラ言語です。命令数が少なく、理解し易い言語ですが、試験用言語なので実務向きではありません。 初学者(開発系)向き。	やや少なめ	やや易～標準
表計算	試験唯一のユーザ向き言語です。身近で理解し易く、実務にも活かれます。 初学者(ユーザ系/文系出身者)向き。	少なめ	やや易～標準

Memo

問2 次のプログラムの説明及びプログラムを読んで、設問1, 2に答えよ。

〔プログラムの説明〕

長さが `Textlen` の文字列 `SourceText` の中で、長さが `Patlen` の文字列 `Pattern` と一致する部分の文字列（以下、部分文字列という）の出現回数を数える関数 `MatchCounter` である。ここで、 $0 < \text{Patlen} \leq \text{Textlen}$ である。

(1) `MatchCounter` の処理手順は、次のとおりである。

- ① 一致する部分文字列の出現回数を数える変数 `Counter` の値を0に初期化する。
- ② `SourceText` の比較開始位置を先頭から順に1文字ずつ後ろにずらしながら、その比較開始位置から始まる長さ `Patlen` の文字列と `Pattern` が一致するかどうかを調べ、一致したら出現回数 `Counter` の値に1を加算する。
- ③ `Counter` の値を返す。

(2) `MatchCounter` の引数と返却値の仕様を表に示す。文字列は文字型配列の各要素に1文字ずつ格納されている。また、各配列の添字は0から始まる。

表 `MatchCounter` の引数と返却値の仕様

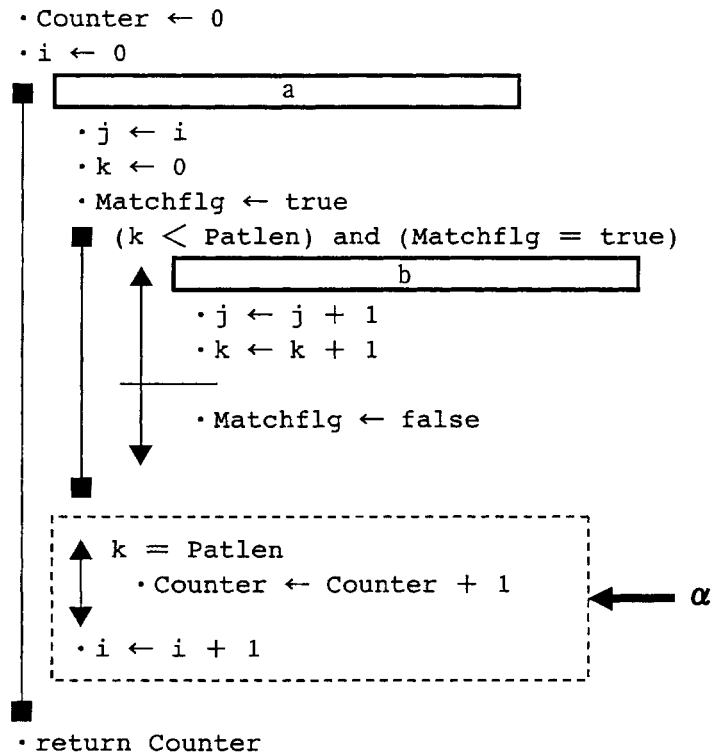
引数／返却値	データ型	入力／出力	意味
<code>SourceText[]</code>	文字型	入力	検索される文字列が格納されている1次元配列
<code>Textlen</code>	整数型	入力	検索される文字列の長さ
<code>Pattern[]</code>	文字型	入力	検索する文字列が格納されている1次元配列
<code>Patlen</code>	整数型	入力	検索する文字列の長さ
返却値	整数型	出力	<code>SourceText</code> の中で <code>Pattern</code> と一致した部分文字列の出現回数

[プログラム]

○整数型: MatchCounter(文字型: SourceText[], 整数型: Textlen,
文字型: Pattern[], 整数型: Patlen)

○整数型: Counter, i, j, k

○論理型: Matchflg



設問1 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。

aに関する解答群

ア $i + \text{Patlen} \leq \text{Textlen}$

イ $i + \text{Patlen} < \text{Textlen}$

ウ $i + \text{Textlen} \leq \text{Patlen}$

エ $i + \text{Textlen} < \text{Patlen}$

bに関する解答群

ア $\text{SourceText}[i] = \text{Pattern}[k]$

イ $\text{SourceText}[j] = \text{Pattern}[k]$

ウ $\text{SourceText}[k] = \text{Pattern}[i]$

エ $\text{SourceText}[k] = \text{Pattern}[j]$

設問2 Patternと一致した部分文字列は以降の検索対象から外すように、このプログラムを変更する。

例えば、SourceTextとPatternが図に示す文字列であるとき、プログラムでは、比較開始位置を①、②、③、…と移動して、一致する部分文字列の出現回数を求めるので、下線の部分文字列と二重下線の部分文字列がともに数えられる。これに対して、一致した部分文字列を検索対象から外す場合は、比較開始位置を①、②、⑤、…と移動するので、二重下線の部分文字列は数えられない。

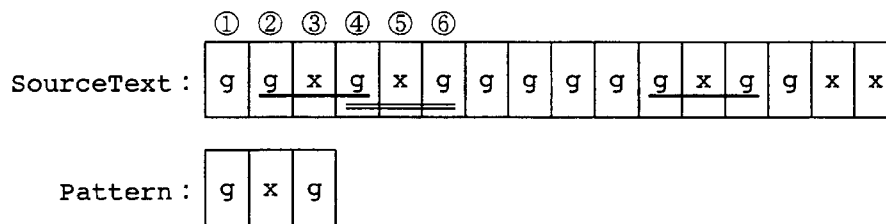
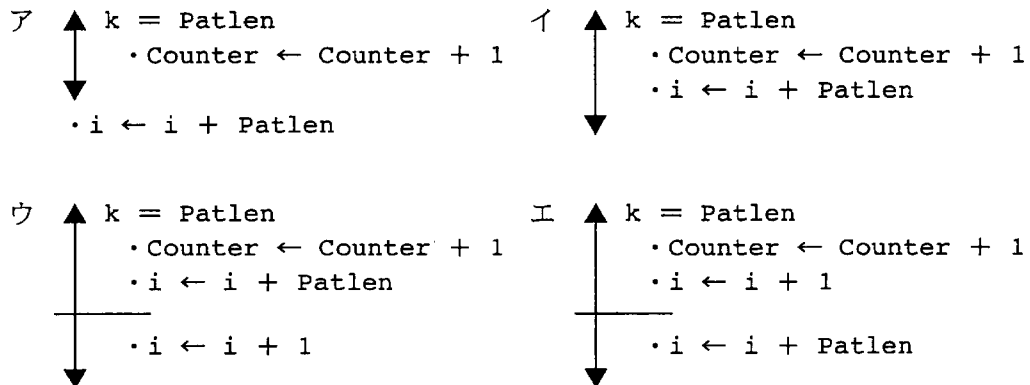


図 比較開始位置を変更する場合の例

このとき、プログラム中の α 部分の変更内容として正しい答えを、解答群の中から選べ。

解答群



【20春FE本試験 午後問2】