

基本情報技術者 無料公開セミナー

～得点力を高めるための問題演習法～



1. 試験日までのスケジュール例

	9月	10月	11月	12月	1月	2月	3月	4月
午前対策		午前 基本講義			午前 問題演習			
午後対策		アルゴリズム プログラム言語 基本講義				午後共通問題 問題演習		
						アルゴリズム プログラム言語 問題演習		

TAC 本科生コースのスケジュール

2. なぜ問題演習が必要なのか？（問題演習を行う目的）

① 解法の習得

解答を導くまでのプロセスや考え方を定着させる

② 知識強化・応用

今までに得た知識を午後試験の問題に適用できるよう確立させる

自分の苦手、理解不十分なテーマを把握して補強する

③ 出題傾向を把握

頻出テーマや重要テーマを把握する

- ・学習済みの知識を速く・正確に・効率良く引き出し、問題文に適用させる
 - ・論理的な思考力を強化し、解答を導けるようにする
- ⇒得点力を高める（問題や答えを覚えることが目的ではない！）

3. 今回の公開セミナーの内容

効果的に得点力を上げるための問題演習法をご紹介します

問題演習はやればやるほど力が付く
しかし、単純に「解くだけ」では効果的とは言えない



- ・どうすれば効率よく得点力を高めることができるか？
- ・問題演習を実施する場合は何に注意すべきか？

4. どのように問題演習を実施すればよいか？

問題演習の方法は問題の種類ごとに異なる

① 午前

② 午後 共通問題（問 1～問 7）

③ 午後 アルゴリズム（問 8）、プログラム言語（問 9～問 13）

⇒問題演習の種類ごとにご紹介

午前問題演習

前提：午前試験では、過去問題が再出題される

- ・午前試験の 80 問のうち、過去問題の比率は約 _____ % (約 _____ 問)
- ・過去 7 回の試験 (H22 年) からの出題は約 _____ % (約 _____ 問)
- ・過去 3 回の試験 (H25 年) からの出題は約 _____ % (約 _____ 問)
- ・3~5 回前の試験から流用されやすく、約 _____ % (約 _____ 問)

過去 7 回の過去問題を全部覚えてもほとんどは初めて見る問題

同じ問題「も」出題されるが、新規問題の方が圧倒的に多い

⇒問題文と答えを覚えても意味がない

しかし…同じテーマは繰り返し出題される！

⇒よく出るテーマと何を知らなくてはいけなさを把握

- 2 分探索の問題例 平成 26 年秋 午前問 6

問 6 2 分探索に関する記述のうち、適切なものはどれか。

- ア 2 分探索するデータ列は整列されている必要がある。
- イ 2 分探索は線形探索よりも常に速く探索できる。
- ウ 2 分探索は探索をデータ列の先頭から開始する。
- エ n 個のデータの 2 分探索に要する比較回数は、 $n \log_2 n$ に比例する。

まったく同じ問題が出題されたのは _____ 年前

ただし、2 分探索というテーマは何回も出題

- 仮想記憶の問題例 (1) 平成 25 年春 午前問 20

問 20 仮想記憶管理におけるページ置換えの方式のうち、LRU 制御方式はどれか。

- ア 各ページに参照フラグと変更フラグを付加して管理し、参照なしかつ変更なしのページを優先して置き換える。
- イ 主記憶にある全てのページを同一の確率でランダムに選択し、置き換える。
- ウ 最も長い間参照されていないページを置き換える。
- エ 最も長い間主記憶にあったページを置き換える。

こちらも _____ 年前の問題を再利用

同じく、仮想管理というテーマは何回も出題

● 仮想記憶の問題例（２） 平成 23 年春 午前問 20

問20 ページング方式の説明として、適切なものはどれか。

- ア 仮想記憶空間と実記憶空間を、固定長の領域に区切り、対応づけて管理する方式
- イ 主記憶装置の異なった領域で実行できるように、プログラムを再配置する方式
- ウ 主記憶装置を、同時に並行して読み書き可能な複数の領域に分ける方式
- エ 補助記憶装置に、複数のレコードをまとめて読み書きする方式

● 仮想記憶の問題例（３） 平成 23 年春 午前問 21

問21 LRU アルゴリズムで、ページ置換えの判断基準に用いられる項目はどれか。

- | | |
|----------------|-------------|
| ア 最後に参照した時刻 | イ 最初に参照した時刻 |
| ウ 単位時間当たりの参照頻度 | エ 累積の参照回数 |

● 仮想記憶の問題例（４） 平成 20 年秋 午前問 27

問27 ページング方式の仮想記憶において、主記憶に存在しないページをアクセスした場合の処理や状態の順番として、適切なものはどれか。ここで、主記憶には現在、空きのページ枠はないものとする。

- ア 置換え対象ページの決定 → ページイン → ページフォールト → ページアウト
- イ 置換え対象ページの決定 → ページフォールト → ページアウト → ページイン
- ウ ページフォールト → 置換え対象ページの決定 → ページアウト → ページイン
- エ ページフォールト → 置換え対象ページの決定 → ページイン → ページアウト

● 仮想記憶の問題例（５） 平成 20 年春 午前問 20

問27 ページング方式の仮想記憶において、ページ置換えアルゴリズムに LRU 方式を採用する。主記憶に割り当てられるページ枠が 4 のとき、ページ 1, 2, 3, 4, 5, 2, 1, 3, 2, 6 の順にアクセスすると、ページ 6 をアクセスする時点で置き換えられるページはどれか。ここで、初期状態では主記憶にどのページも存在しないものとする。

- | | | | |
|-----|-----|-----|-----|
| ア 1 | イ 2 | ウ 4 | エ 5 |
|-----|-----|-----|-----|

いずれも仮想記憶に関する問題 ⇒ 何を知っていなければならないか？

- ・仮想記憶システムでは、仮想記憶領域のアドレス（仮想アドレス）と実記憶領域のアドレス（実アドレス空間）を対応付けて管理する。
- ・ページング方式では、必要なページが実記憶（主記憶）になればページフォールトが発生する。
- ・ページフォールトが発生したら、必要なページを主記憶に読み込む（ページイン）。実記憶に空きがない場合は、最も必要としないであろうページを決め、仮想記憶にページアウトしてからページインする（ページ置換え）。
- ・どのページをページアウトするかを決めるアルゴリズムをページ置換えアルゴリズムという。
- ・ページ置換えアルゴリズムには、FIFO 方式、LRU 方式、などがある。
- ・LRU 方式では、最後に参照されてからの時間が最も長い（最も長く参照されていない）ページを選ぶ。

基本を抑えていれば、どんなに形の違う問題が出題されても解答可能

⇒基本を抑えずに問題の答えを覚えても意味がない

⇒もちろん覚えていれば、より速く確実に解ける

●午前問題の演習目的

- ・理解度の確認（午前試験に合格できるだけの知識があるか）
- ・どのようなことが問われやすいのか（頻出論点，重要ポイント）を把握
- ・どのような形式で出題されるか（出題形式）を把握

●午前問題の問題演習法

① まずは問題を解く

- ・何が正解か，を考える（なぜ正解か？もあわせて考える）
- ・他の選択肢の誤りである根拠も「解説を見る前に」，「自分で」考える
 - ⇒どのような理由で正解ではないのか？
 - ⇒その文章は何に関する記述か？
 - ⇒各用語はどのようなものなのか？

② 解説を読んで，自分が考えた根拠と同じかどうかをチェック

③ 関連する知識，理解不十分な知識，知らない（説明できない）用語などがあれば，それも確認

午前の問題演習は正解／不正解にこだわらないこと

技術や仕組みの理解，解いた後の知識の再確認が重要

誤り選択肢の確認は関連する知識への展開に有用

● 関連する問題の例（１）

平成 24 年春 午前問 11

問11 外部割込みに分類されるものはどれか。

- ア インターバルタイマによって，指定時間経過時に生じる割込み
- イ 演算結果のオーバフローやゼロによる除算で生じる割込み
- ウ 仮想記憶管理において，存在しないページへのアクセスによって生じる割込み
- エ ソフトウェア割込み命令の実行によって生じる割込み

● 関連する問題例（２）

平成 26 年春 午前問 9

問 9 主記憶に記憶されたプログラムを，CPU が順に読み出しながら実行する方式はどれか。

- | | |
|-----------|-------------|
| ア DMA制御方式 | イ アドレス指定方式 |
| ウ 仮想記憶方式 | エ プログラム格納方式 |

午後問題演習

午後試験の出題構成

問 1	情報セキュリティ	12 点	必須	共通問題 (選択問題)
問 2～問 4	ハードウェア，ソフトウェア，ネットワーク，データベースの 4 分野から 3 問出題	各 12 点	4 問選択／6 問出題	
問 5	システム開発	12 点		
問 6	マネジメント	12 点		
問 7	ストラテジ	12 点		
問 8	データ構造とアルゴリズム	20 点	必須	アルゴリズム
問 9	C	20 点	1 問選択／5 問出題	プログラム言語
問 10	COBOL	20 点		
問 11	Java	20 点		
問 12	CASL II	20 点		
問 13	表計算	20 点		

1. 午後の共通問題（問 1 ～問 7）

- ・ 出題される分野，テーマ，出題範囲などは午前試験と大きく変わらない
⇒ 必要な知識は午前問題の延長線上（午前試験の知識範囲を超えるなら問題文で説明されるはず）
⇒ 事前に午前対策をきちんと行うことも重要
- ・ ただし，午前問題が解ければ午後問題も解けるとは限らない
⇒ 問題文が長く複雑になるので，午前知識「だけ」あれば十分とはいえない
⇒ 前提となる原理や仕組みをきちんと理解・イメージできているか？が重要
- ・ では，午後問題と午前問題は何が違うか？
午後問題では，問題文章（問題事例）が長く複雑＋複数の知識を複合して要求する
⇒ 午前よりも深い知識（原理や仕組みの理解），長い問題文や図表の読み取り力も必要



このために問題演習を活用！！

2. 午後の選択問題（問 1 ～問 7）の出題パターン

- ・ 知識重視：午前の知識を活用して答えるパターン
⇒ 午前の問題演習を通じて，知識を広く習得することが重要
- ・ 読解力＋思考力重視：問題文に従って解答を導くパターン
⇒ 午前知識をベースに問題文の読解力や図表の解析力，論理思考力が重要
⇒ 「問題文の意味を理解する」ためには知識も必要

●午前知識を重視した設問の例（平成 25 年春 問 2 設問 1&設問 3）

設問 1 次の記述中の に入れる正しい答えを、解答群の中から選べ。

プログラムの実行過程で存在ビットを調べ、プログラムの実行に必要なページが a に存在していないときには、ページフォールトという割込みが発生する。ページフォールトが発生すると、ページアウトやページインなどのページ置換え処理が実行される。ページ置換え処理のアルゴリズムには、ページインしてから最も時間が経過しているページを置換え対象とする FIFO アルゴリズムや、参照されていない時間が最も長いページを置換え対象とする b アルゴリズムなどがある。

解答群

- | | | |
|------------|------------|-------|
| ア LFU | イ LIFO | ウ LRU |
| エ 仮想アドレス空間 | オ 物理アドレス空間 | |

設問 3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

ページ置換えアルゴリズムとして FIFO アルゴリズムを採用する。プログラムの実行過程で仮想ページが次の順で参照されるとき、物理ページの個数が 3 の場合のページフォールトの回数は c 回である。そして、物理ページの個数を 4 に増やした場合のページフォールトの回数は d 回である。ここで、プログラムの実行開始時点では、物理アドレス空間にはどのページも存在していないものとする。

【仮想ページの参照順を示す仮想ページ番号の並び】

1 → 4 → 3 → 2 → 1 → 4 → 5 → 1 → 4 → 3 → 2 → 5 → 1

解答群

- | | | | | |
|-----|-----|------|------|------|
| ア 8 | イ 9 | ウ 10 | エ 11 | オ 12 |
|-----|-----|------|------|------|

何を知っていなければならないか？

- ・仮想記憶システムでは、仮想記憶領域のアドレス（仮想アドレス）と実記憶領域のアドレス（実アドレス空間）を対応付ける。
- ・ページング方式では、必要なページが実記憶（主記憶）になればページフォールトが発生する。
- ・ページフォールトが発生したら、必要なページを主記憶に読み込む（ページイン）。実記憶に空きがない場合は、最も必要としないであろうページを仮想記憶にページアウトしてからページインする。
- ・どのページをページアウトするかを決めるアルゴリズムをページ置き換えアルゴリズムという。
- ・ページ置き換えアルゴリズムには、FIFO 方式、LRU 方式などがある。
- ・LRU 方式では、最後に参照されてからの時間が最も長い（最も長く参照されていない）ページを選ぶ。

⇒午前知識がそのまま活きる！

ジと物理アドレス空間のページの対応例を，図 1 に示す。図 1 では，補助記憶装置に格納されているプログラム A は a1, a2, a3, a4, a5 に分割されて，仮想ページ番号 1～5 のページに格納されている。

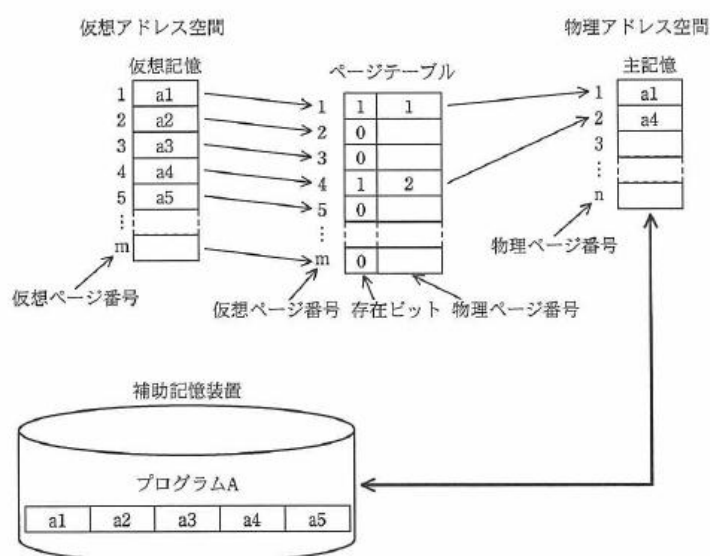


図 1 仮想アドレス空間のページと物理アドレス空間のページの対応例

設問 2 プログラム A を実行するために割り当てられた物理アドレス空間の物理ページの個数が 3 の場合を考える。プログラム A の実行過程において，物理アドレス空間に a1, a2, a3 が存在している状態で a4 を参照するとページフォールトが発生する。このページフォールトが発生した後の処理の流れとして適切な答えを，解答群の中から選べ。ここで，解答群中の処理は左から右に向かって行なうものとする。

【処理の単位】

- ① 退避させるページをページアウトする。
- ② ページ置換えアルゴリズムによって，物理アドレス空間からページアウトするページを決定する。
- ③ 実行に必要なページをページインする。
- ④ ページアウトしたページに対応するページテーブルの要素の存在ビットを 0 にする。
- ⑤ ページインしたページに対応するページテーブルの要素の存在ビットを 1 にする。
- ⑥ ページアウトしたページに対応するページテーブルの要素の物理ページ番号を設定する。
- ⑦ ページインしたページに対応するページテーブルの要素の物理ページ番号を設定する。

解答群

- | | |
|---------------|---------------|
| ア ①→②→③→④→⑤→⑥ | イ ①→③→②→④→⑦→⑤ |
| ウ ②→①→④→③→⑤→⑥ | エ ②→①→④→③→⑦→⑤ |
| オ ②→③→①→④→⑤→⑥ | カ ②→③→⑤→⑥→①→④ |

最低限の知識を得た上で，問題文章や図表を読んで解答を導く

午後問題演習（共通問題）の問題演習法

- ① まずは問題を解く
 - ・何が正解か？を考える ⇒ 「なぜ正解なのか」の根拠を重視
 - ・可能であれば、他の選択肢が誤りである根拠も考える
- ② 解説を読んで、解答導出プロセス（どの文章や図に注目したか、どのように計算したか、など）が自分の考えと同じかどうかをチェック
- ③ 同じでなければ解説に沿ってチェック
- ④ 知識不足が原因の場合はテキストで再確認（仕組みからきちんと）。関連する知識があれば確認

午後の問題演習では解答を導出するプロセスの確認が重要
知識不足がわかったらテキストに戻って体系的な知識確認を



同じような問題が出題されたときのために解法をパターン化

●午後では「同じ問題」は出題されないが、「同じテーマ」は何回も出題

平成 26 年秋 問 2

問 2 機械語命令に関する次の記述を読んで、設問 1、2 に答えよ。

命令語の形式を、図 1 に示す。

(ビット番号) 15 12 11 10 9 8 7 6 5 0

op	r 又は m	x	b	d
----	--------	---	---	---

図 1 命令語の形式

図 1 で使用している記号の説明を、表 1 に示す。数字の末尾は 16 進数表記である。

記号	説明
op	命令コード
r	レジスタ番号
m	分岐命令で分岐の判定に使用する値
x	指標レジスタとして使用するレジスタの番号
b	ベースレジスタとして使用するレジスタの番号
d	00h～3Fh で示されるアドレスの変位

平成 23 年春 問 1

問 1 機械語命令に関する次の記述を読んで、設問 1、2 に答えよ。

この機械語が実行される CPU の 1 語は 16 ビットで、CPU には 1 語長の汎用レジスタが四つ（レジスタ番号 0～3）ある。主記憶容量は 1,000 語（番地 0～999）あり、命令語は 700 番地以降に格納される。命令語の形式は図 1 に示すとおりである。命令には 1 語命令と 2 語命令があり、1 語命令の場合は 1 語目だけで構成される。

1語目 15 8 7 6 5 4 3 2 1 0 (ビット番号)

op	r ₁	r ₂ 又は X ¹⁾	0 0	D	I
----	----------------	-----------------------------------	-----	---	---

2語目 15 0 (ビット番号)

adr

注¹⁾ 1語命令のときにはr₂、2語命令のときにはX

図 1 命令語の形式

図 1 で使用している記号の説明を表 1 に、命令の実効アドレスの算出方法を表 2 に、命令の対象となるデータが設定されているレジスタ（以下、ソースレジスタという）の指定方法を表 3 に、命令の仕様（一部）を表 4 に示す。数字の末尾に h が付いているものは 16 進数表記である。

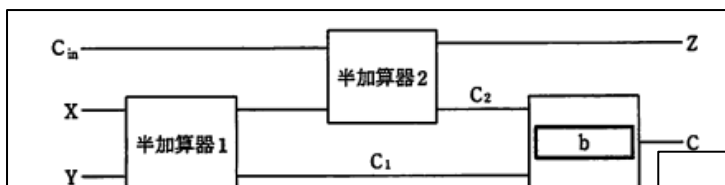


図2 全加算器を実現する論理回路

表3 X, Y, C_{in} と, C_1 , C_2 との関係

C_{in}	X	Y	C_1	C_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	c	

表3 全加算器の真理値表

C_{in}	X	Y	C	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

} β

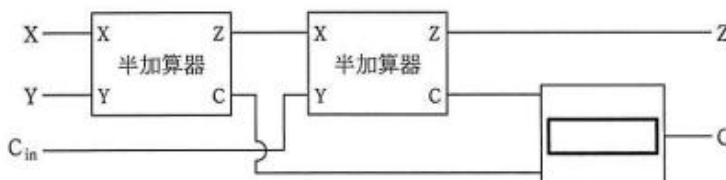


図2 全加算器の論理回路

なるべく出題率の高いテーマを優先的に演習できればベター

⇒ 同じテーマの問題が出たら非常に有利

問題文や図表の読み取り方はすべての幅広く応用できる

⇒ 出題率が高くななくても、問題演習は「やればやるほど」得点力になる

3. 午後問題（選択問題）の時間配分対策

午後試験では時間配分が重要

・ 時間配分の例

問 1：情報セキュリティ → 15 分

問 2～7 の 4 問 → 各 15 分

問 8（アルゴリズム） → 30 分

問 9～13 の 1 問

（言語または表計算） → 30 分

合計 → 2 時間 15 分

} 概ね 15 分

アルゴリズムが苦手な方はもっと短く

・ 午後試験が苦手な方は、問題演習を 3 ステップで

ステップ 1：解答導出プロセスの確認と知識の補強，問題文や図表の読解力を養成

ステップ 2：正確に解答を導けるように

ステップ 3：時間（10 分や 15 分など）を決め，制限時間内に問題を解く

※ステップ 1 とステップ 2 を統合して 2 ステップ or 1 ステップでも ok

午後問題演習（アルゴリズム・プログラム言語）の問題演習法

- ・午前試験と同じ知識も要求されるが、プログラミング力（論理的思考力）をより強く要求
⇒必要最低限の知識は持っておきたいが、プログラムを読んで考えることが重要
- ・問われやすい設問のパターンは2つ

- ・プログラムの処理設計（空欄補充、複数行をまとめたプログラムの改造や機能追加など）
- ・プログラムの実行結果（出力されるデータ、●●の処理が何回実行されるか？など）

- ⇒ どのような処理（条件）が必要か？処理結果はどのようなになるか？を「考える力」が必要
- ⇒ これ以外のパターン(時間計算量など) も考えられるが出題率は高くない

「考える力」を付けるために問題演習を活用！

※ただし、やみくもに問題を解いても力はつかない ⇒ 得点力につながる問題演習を！！

1. 問題演習を始める前に知っておいて欲しいこと

●問題演習を行う前提として、基本アルゴリズムとデータ構造は理解

【基本アルゴリズム】

最大値(最小値)アルゴリズム

線形探索, 2 分探索

基本整列アルゴリズム(選択法, 交換法, 挿入法)

高速整列アルゴリズム(クイックソート, マージソートなど)

文字列照合, 文字列置換

【データ構造】

配列

リスト

スタック, キュー

木

グラフ

●問題演習はアルゴリズムから

プログラム言語 = アルゴリズムの論理思考力+文法+言語特有の処理

●「考える力」を得るためには、プログラムを正確に「追う」ことが重要

- ・プログラムを追う（トレース） ⇔ プログラムを頭の中で実行
- ・複雑なプログラムを頭の中だけで考えるのは困難 ⇒ 必ずメモしながら（手を動かす）

●短期間では力がつかないので焦らずに継続的に

暗記ではないので「覚えれば解ける」というものではない

2. 問題演習は3ステップで

ステップ1：まずはプログラムを正確に追うことができるように

ステップ2：プログラムの一部をパターン化して考えることができるように

ステップ3：時間内（30分～）に6割以上の得点をとれるように

※得意な方はステップ1とステップ2を統合して2ステップでも ok

●ステップ1：プログラムを正確に追うための演習

① まずは何をやるプログラムなのかを把握

特に入力するデータと出力するデータの把握は重要

平成 25 年秋 問 8

〔プログラムの説明〕

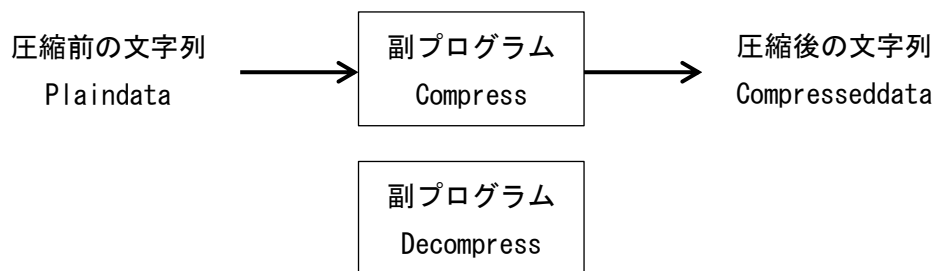
英字 A～Z から構成される文字列を圧縮する副プログラム Compress、及び圧縮された文字列を復元する副プログラム Decompress である。

〔圧縮処理の説明〕

副プログラム Compress では、配列 Plaindata に格納されている圧縮前の文字列を受け取り、圧縮後の文字列を配列 Compresseddata に格納する。図 1 に示す文字列 “ABCDEF ABCD ABCDEF” を例として、圧縮処理の内容を説明する。

圧縮前の文字列 (Plaindata)	:	A B C D E F	A B C D	A B C D E F
			(s1)	(s2)
圧縮後の文字列 (Compresseddata)	:	A B C D E F	\$ F D	\$ J F
			(s1')	(s2')

図 1 文字列の圧縮例



② 設問を見て何が問われるかを把握

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。

設問 2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

次の文字列を圧縮した文字列を副プログラム Decompress を使って復元する場合、プログラム 2 の α の部分は f 回実行される。

文字列 : ABCDEFGABCDEABCDFEFGABCD

- ・この問題では、プログラムの穴埋めをしてからプログラムの実行回数を答える
- ・問題によっては、最初の設問でデータや処理の理解を深めるための簡単な設問がある場合も
⇒基本的には「設問 1 から順に」が解きやすく作られている
設問 1 が処理やデータを把握するヒントとなることもある

③ 問題文や図から、処理の概要やデータどのように実現（処理）するのかを把握

(4) 圧縮する文字並びの検索処理の内容を次に示す。

- ① 圧縮処理の対象となる文字並びを圧縮文字並び、圧縮文字並びの先頭位置を圧縮文字位置という。圧縮列の文字数は3なので、一致する文字数が4以上の圧縮文字並びの場合に圧縮列に置き換える。したがって、最初の圧縮文字位置は、図2に示すように圧縮前の文字列の先頭から5文字目となる。この文字位置から圧縮文字位置を文字列の後方に向かって移動させながら検索処理②、③を行う。

最初の圧縮文字位置
↓
圧縮前の文字列 : A B C D E F A B C D A B C D E F

図2 最初の圧縮文字位置

- ② 圧縮文字並びの比較対象とする文字並びを比較文字並び、比較文字並びの先頭位置を比較文字位置という。最初の比較文字位置は、図3に示すように圧縮文字位置の4文字前とする。

(1) 図1の圧縮前の文字列“ABCDEFABCD…”のように、“ABCD”という同じ文字の並びが複数回出現する場合、二つ目以降の“ABCD”を圧縮列に置き換えることによって、文字列の長さを短くする。図1では、Plaindataに格納されている圧縮前の文字並び(s1)、(s2)を圧縮列(s1')、(s2')に置き換えてCompresseddataに格納している。

(2) 圧縮列は、制御記号、距離、文字数の三つの部分から成る。

- ① 制御記号は、“\$”であり、圧縮列の先頭を表す。
② 距離と文字数は、圧縮前の文字列において、“この場所の文字並びは、何文字前(距離)の文字から始まる何文字の長さ(文字数)の文字並びと同じである”という内容を意味する。距離と文字数の最大値は26とし、1, 2, …, 26をA, B, …, Zで表す。図1の例では、圧縮前の文字並び(s1)は、(s1)の先頭文字の

A B C D E F \$ F D \$ J F (図1の Compresseddata)

文字数 : 4
距離 : 6文字前
制御記号 (圧縮されている)

ここから推測できること

- ・文字列が一致する部分を探す処理が必要ではないか？
- ・何文字前かを表す情報（変数）とそれを求める処理が必要ではないか？
- ・何文字一致したかを表す情報（変数）それを求める処理が必要ではないか？

⇒間違ってもよい（プログラムが何をやっているのか？を把握するのに有効）

- ④ プログラムを読まなくても説明や図から答えられる設問があれば、答えてもよい
⇒処理の概要や入出力データの把握に役立つことが多い
- ⑤ 問題文、図表などから処理の概要や使用する変数、引数（プログラムに与えるデータ）を把握
引数は問題文中や図表で提示されることが多い（本問は表を用いたパターン）

〔副プログラム Compress の引数の仕様〕			
Compress の引数の仕様は、次のとおりである。各配列の添字は、0 から始まる。			
引数名	データ型	入力／出力	意味
Plaindata[]	文字型	入力	圧縮前の文字列が格納されている 1 次元配列
Plength	整数型	入力	圧縮前の文字列の長さ（1 以上）
Compresseddata[]	文字型	出力	圧縮後の文字列が格納される 1 次元配列
Clength	整数型	出力	圧縮後の文字列の長さ

プログラムの冒頭には使用する変数が

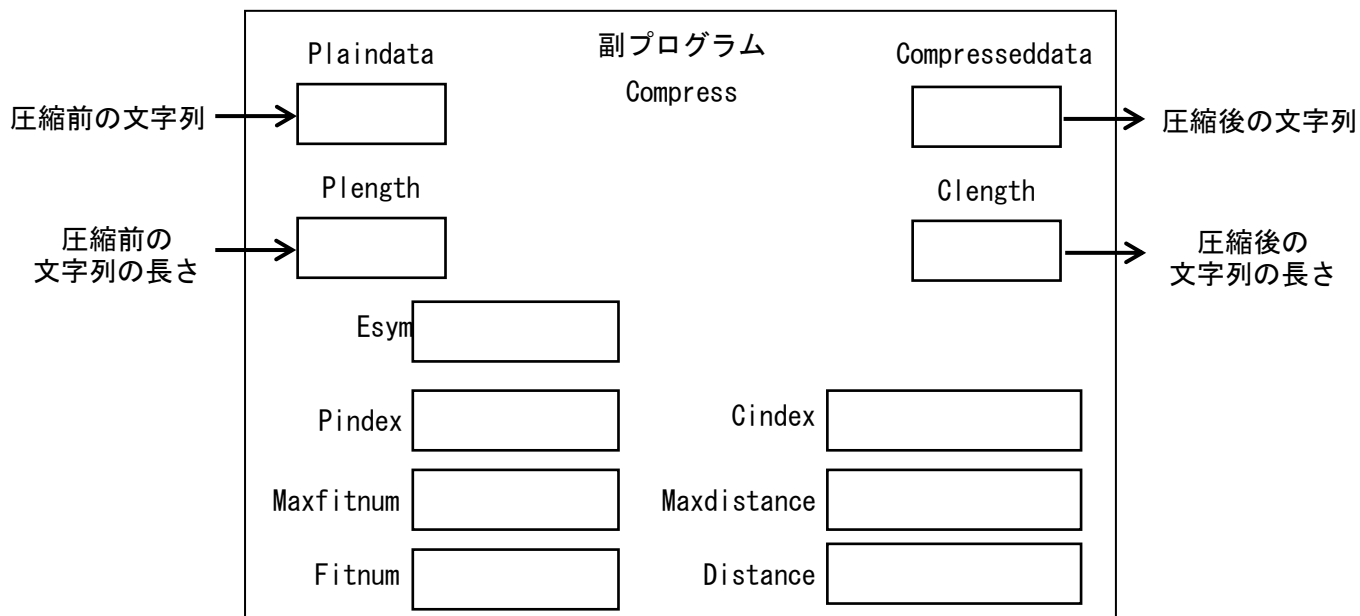
〔プログラム 1〕

○副プログラム : Compress(文字型 : Plaindata[], 整数型 : Plength, 文字型 : Compresseddata[], 整数型 : Clength)

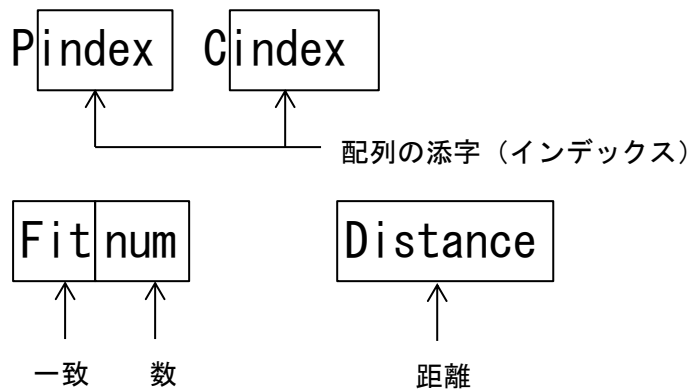
○文字型 : Esym

○整数型 : Pindex, Cindex

○整数型 : Maxfitnum, Maxdistance, Distance, Fitnum



⑥ 各変数（データ）の意味をイメージする



ということは Maxfitnum や Maxdistance は・・・

ここから推測できること

- ・何文字前か？を求めるために **Distance** を使うのではないか？
- ・何文字一致したか？を求めるために **Fitnum** を使うのではないか？
- ・**Maxfitnum** や **Maxdistance** を求めるために最大値のアルゴリズムを使うのではないか？

※ この段階では間違っても構わない（プログラムを読む前に考える癖を付けることが重要）

⑦ プログラムをトレースしながら、実際に問題を解く

- ・空欄補充の問題なら、プログラムを先頭から読みながら空欄に入れる条件や処理を考える
⇒わからなければ、空欄をとばして、不具合が出たときにそこに戻ってみるのも一手
- ・実行結果の問題なら、プログラムをトレースした結果がそのまま解答となる

- ・各変数の値を「必ず」メモしながら（消さないこと）

```

▲ Plength ≥ 1
• Esym ← "$"          /* 制御記号の設定 */
• Pindex ← 0           /* 圧縮前の文字列の文字位置を初期化 */
• Cindex ← 0           /* 圧縮後の文字列の文字位置を初期化 */
■ (Pindex < Plength) and (Pindex < 4)
  • Compresseddata[Cindex] ← Plaindata[Pindex]
  • Cindex ← Cindex + 1
  • Pindex ← Pindex + 1
■
■ Pindex < Plength
  • Maxfitnum ← -1
  • Maxdistance ← -1
  • Distance ← 4
  ■ (Distance ≤ 26) and ( a )

```

- ・プログラム「だけ」を見るのではなく、③でチェックした問題文とも照らし合わせる

- ⑧ ひとつの問題を解き終わったら解説を確認
- ・解説と自分の考えが一致しているかをチェック

アルゴリズム・言語でも解答導出プロセスの確認が重要

- ・変数の値や状態が図示されていたら、自分の書いた図と一緒にチェック
 - ⇒値が違っていれば「必ず」トレースしなおす
 - ⇒値が同じでも解説の図の方がわかりやすそうなら、次から書式をまねてみる

●アルゴリズム&プログラム言語の対策は「慣れ」が重要

はじめのうちは空欄に入れる処理を考えるのは大変
まったくわからなかったら、答えを見てもよい ⇒ その代わり最後までトレースする
慣れてきたら、選択肢を見て「これかな？」と思うものを入れてトレースしてみる
最終的には論理的に解答を導けるように

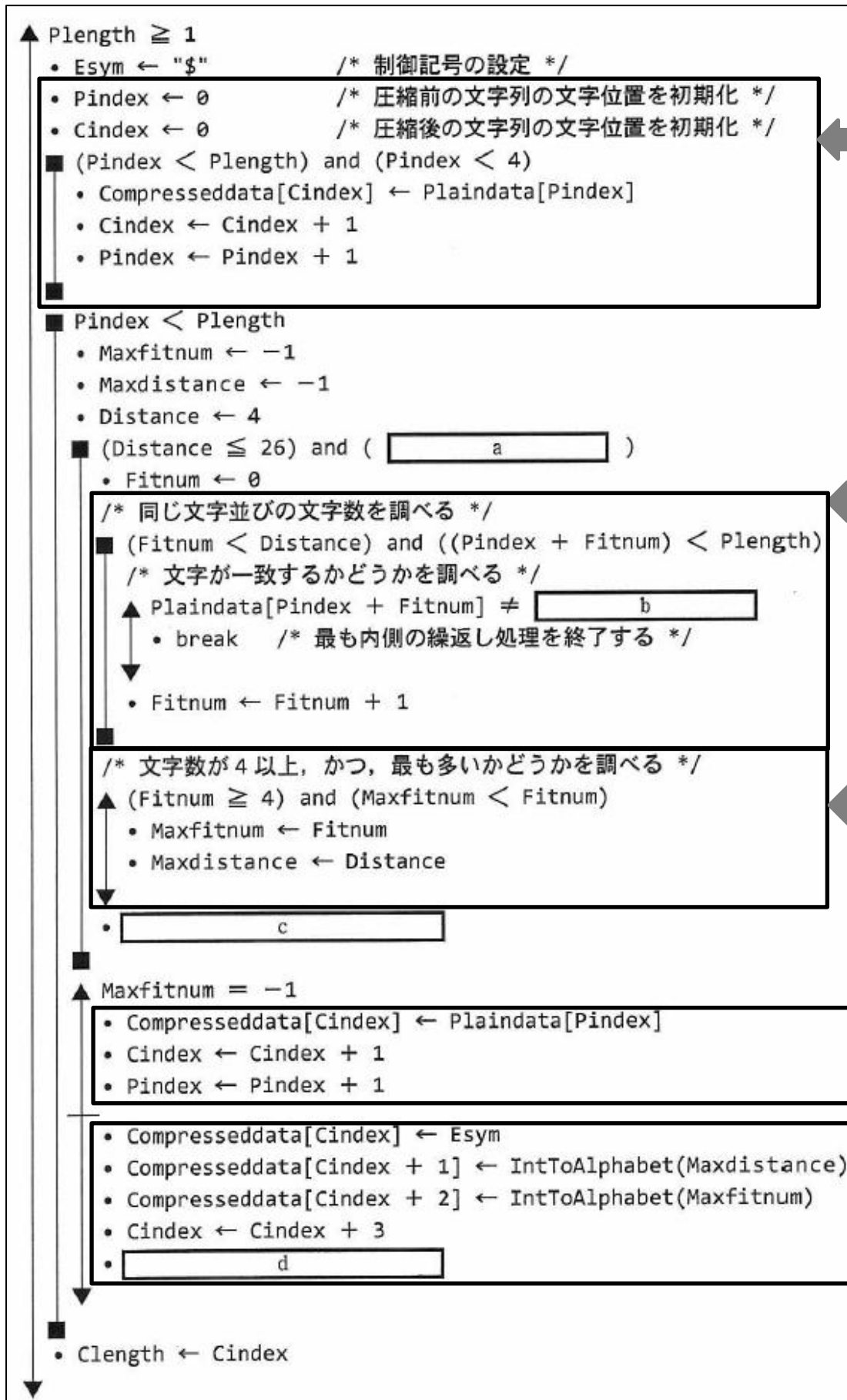
時間がかかるので焦らずに！

●ステップ2：プログラムの一部をパターン化して考えるための演習

同じ問題は出題されないが、いくつも問題を解いていくと、同じような処理（パターン）が出てくる

⇒ 慣れてくると、同じようなパターンは1行ずつ追わなくてもわかるようになる

⇒ プログラムを1行ずつ追わなくても、どこで何をやっているかの概要を把握する練習を



①
Plaindata の先頭 4 文字を
Compresseddata に複写
⇒ 文字列の複写

②
・ループ中で文字が一致し
ない(≠)場合にループを
抜ける
・一致している場合はループを
続ける
・ループの中で1ずつ増える
変数 Fitnum が文字数
⇒ 個数を数える

③
・fitnum が4以上で
Maxfitnum より大きければ
その Fitnum で Maxfitnum
を更新
⇒ 最大値のアルゴリズム

⑤
Plaindata の1文字を
Compresseddata に複写
⇒ 圧縮しない

④
制御文字, 距離, 文字
数を設定
⇒ 圧縮

●ステップ3：時間内に6割以上の得点をとるための演習

時間を決めて（30分など）、問題を解く

- ・問題文の重要箇所にマークする
- ・必要な変数の絵を描く
- ・処理と問題文章をリンクさせる
- ・プログラムをパーツに分解し、肝となる部分に力を入れる

必要な手順を最低限にしつつ、短時間で高い正答率を

●プログラム言語の問題演習

基本的な流れや考え方はアルゴリズムと同じ

この段階ではプログラム言語に特化した対策を
（言語特有の概念、文法、処理など）

C

- ・ポインタ、再帰的プログラム、ライブラリ関数、数値処理

Java

- ・オブジェクト指向（クラス、コンストラクタ、継承、抽象クラス、インタフェースなど）、API

COBOL

- ・各種ファイル処理、事務処理（コントロールブレイク処理、マッチングなど）、SORT文、索引ファイル

CASL II

- ・アセンブラ（レジスタ、アドレス）
- ・ビット操作、シフト、マスクパターン、JUMP命令、乗算・除算処理

表計算

- ・表計算（ワークシート、セル）、関数、セル参照（絶対参照、相対参照）
- ・マクロ（書式はアルゴリズムと同様だが変数だけでなくセルの値を扱う）

※特にCASL IIと表計算は、アルゴリズムと異なる概念が多いので準備期間を十分に

プログラム言語も、2ステップで

- ・ステップ1：時間を気にせず正答を得られるように
- ・ステップ2：時間を決め（30分など）時間内に6割以上正解できるように

TAC 講座のご案内

TAC では、午後試験の合格を目指すコースをご用意いたしております。

●午後演習パック

午後試験の問題演習に特化したコース、すでに知識はあるが結果が出ていない方向け

- ・問題を実際に解き、問題ごとのアプローチ方法や重要ポイントを講師が解説
- ・どこが重要なのか、何が問われやすいか、など独学では気づき難いポイントがわかる

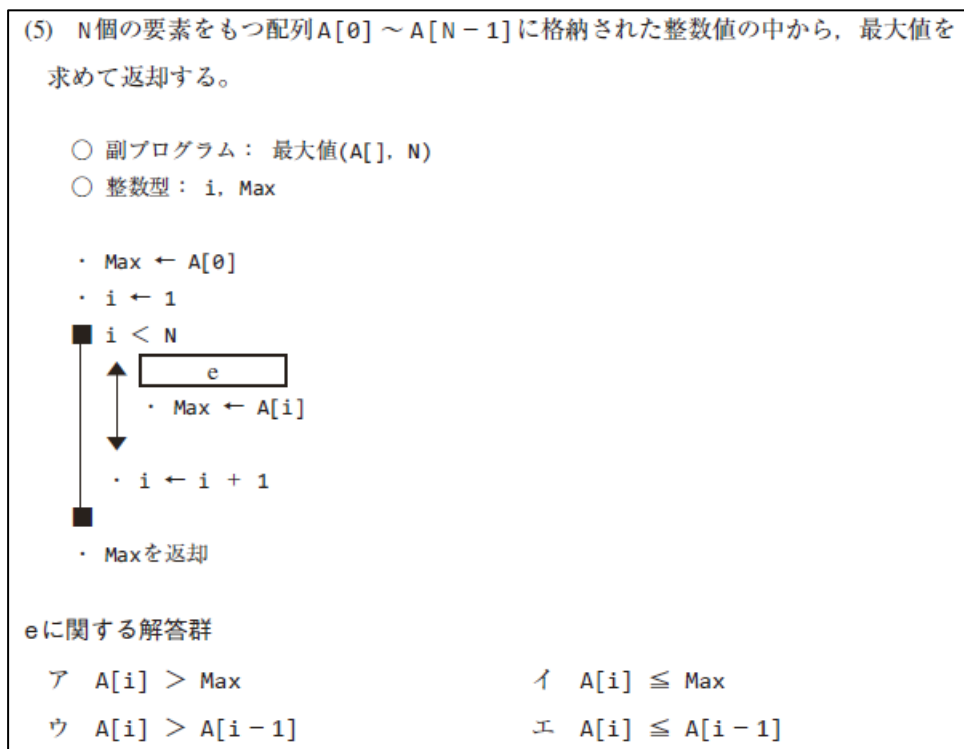
ご興味ございましたら、ぜひ体験コース（1/24）にご参加ください

●午後マスターコース（開講済）

午後試験対策用のコース、問題演習の前提となる基本知識に不安のある方向け

- ・ 午後試験対策用講義＋午後演習パック
- ・ 午後試験対策で重要となる前提知識を重点的に講師が解説
- ・ 午後試験対策問題集で簡単なアルゴリズムの演習

午後対策問題集（基礎編）の問題例



※講義は DVD または Web での視聴となります。1/24 より午後演習パックに合流します

皆様のご参加をお待ち申し上げます

