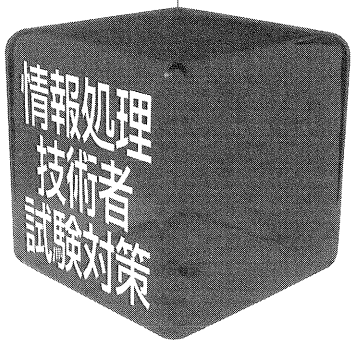
高度共通 午前Ⅰ対策

合格テキスト&トレーニング

TAC情報処理講座



TAC出版

はじめに

本書は，高度情報処理技術者の午前Ⅰ試験対策のために，最少時間で最大効果をあげることができるように作成したもので，次の三つの特徴があります。

○第一に，受験生が抱える事情にでき得るかぎりお応えすることを基本方針としています。
○第二に，過去問題の徹底分析をもとに頻出項目に絞って「知識編」を構成しています。
○第三に，合格に必須な実力を養成できるように「問題編」を構成しています。

第一の受験者が抱える事情とは，午前Ⅱ，午後Ⅰ，午後Ⅱの専門試験の学習時間を確保するために，午前Ⅰ試験の学習にはあまり時間を割きたくないというものです。また，高度情報処理技術者試験は，午前Ⅰ試験に合格しなければ，それ以降の試験は採点されません。そのため，かつて午前Ⅰの試験範囲を学習したことがあり，再学習に時間をかけたくないと考える方が多いと思われます。そこで少ない時間で本試験合格レベルの学力を培う，という基本方針で本書を作成しました。

第二に「知識編」は，本試験の徹底分析に基づき，掲載する項目を厳選しています。さらに，TACが長年受験対策を指導してきた実績をもとに，出題の可能性が高い項目も掲載しています。また，解説は試験に必要な最小限のものとしています。

第三に「問題編」は，出題率が高い過去問題を中心に構成しています。午前Ⅰ試験は再出題率が高く，まったく同じではなくても類似した内容が繰り返し問われることが多いので，過去問題演習が最も効果的で効率的です。

さらに，本書では「知識編」と「問題編」をリンクさせていますので，習得した知識を確認するための問題演習をすぐ行うことができ，また，問題演習で不足だと感じた知識をすぐに補うことができます。

これら三つの特徴をもつ本書を活用し，午前Ⅰ試験に合格されることを願ってやみません。

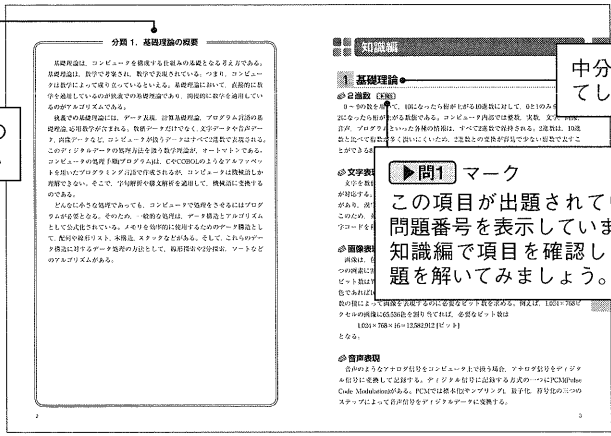
2012年8月　TAC情報処理講座

本書の使い方

知識編と問題編を行き来し，繰り返し演習していただくことで，最も効率の良い午前Ⅰ対策を行うことができます。

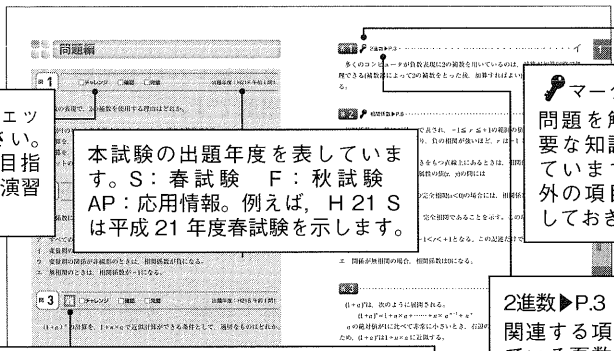
＊本書の「分類」は共通キャリア・スキルフレームワークの「大分類」に相当します。

知識編　頻出項目にポイントをしぼり，わかりやすく解説しています。



分野ごとにサイクル学習

問題編　過去の本試験午前Ⅰ問題と応用情報技術者試験（AP）の午前問題から180問を厳選しています。午前Ⅰ問題は再出題率が高いので，何度も挑戦してみてください。



試験戦略と学習戦略

1　試験戦略

情報処理技術者試験には，午前Ⅰ試験，午前Ⅱ試験，午後Ⅰ試験，午後Ⅱ試験と四つのマイルストーンがあります。そして，これらのマイルストーンを順番に通過して，初めて合格にたどりつくことができます。これは，午前Ⅱ試験に手ごたえがあったとしても，午前Ⅰ試験にパスしていなければ，午前Ⅱ試験は採点の対象にならないということです。

午前Ⅰ試験の攻略を考える際のポイントは，「最小エネルギーで第1のマイルストーンを通過すること」です。午前Ⅰ試験は，60％以上正解すればパスできます。これは，出題30問中，最低18問正解すればいいということです。余裕を持って，19問確実に正解することを目指しましょう。

19問は確実に正解できるという学習をこなし，試験場では，確実に正解できる19問から先に解答しましょう。取りこぼしのないようにじっくり取り組むことが重要です。その後，残りの11問に対して，知識と勘を働かせて解答しましょう。

このようにして試験に臨めば，午前Ⅰ試験は思っているよりも高い点数が獲得できるでしょう。

2　学習戦略

「19問を確実に正解するための学習」は，二つのステップで完成させます。第1ステップは，「試験問題の傾向の把握と学習する分野の選択」で，第2ステップは，「分野ごとのサイクル学習」です。

第1ステップの「試験問題の傾向の把握と学習する分野の選択」では，まず，試験で出題された問題のテーマを「分野」ごとに分類します。ここで「分野」とは，情報処理技術者試験の共通キャリア・スキルフレームワークの「大分類」を示しています。毎回，分野ごとに出題される問題数には大きな変動はありません。次に，自分がどの分野が得意なのかを加味して，合計すれば19問は正解できるという分野をピックアップします。このとき，分野の範囲は広いので，その中に苦手なものが含まれているかもしれません。その場合，一つ下のレベル（中分類）まで出題される問題数をブレイクダウンして把握するといいでしょう。このようにして，この分野を学習しておけば19問は正解できるという学習する分野を決定します。

第2ステップの「分野ごとのサイクル学習」では，第1ステップで選択した分野に対

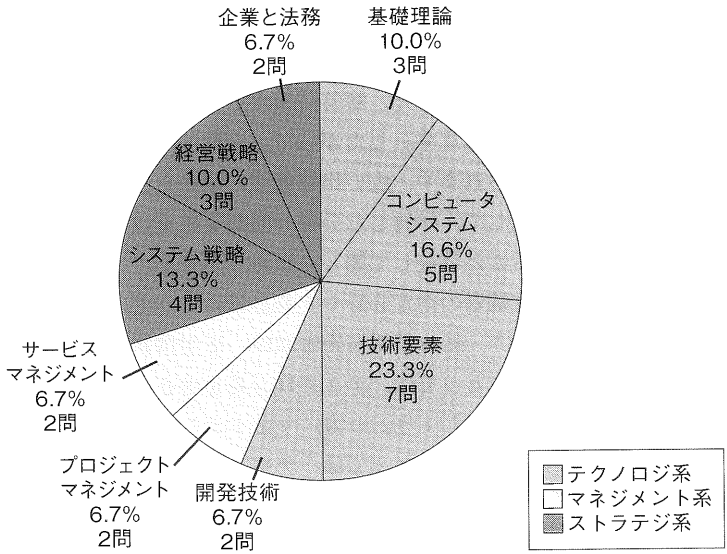
して，サイクル学習をします。サイクル学習とは，すべての分野の知識学習を済ませてから問題演習を行うのではなく，分野ごとに知識学習と問題演習を行い，それを分野の数だけ繰り返すことを意味します。サイクル学習によって，より確実な知識の習得を図ることができます。

2012年度春期試験　傾向分析

1　問題テーマの特徴

今回は，テクノロジ系分野から17問，マネジメント系分野から4問，ストラテジ系分野から9問出題されました。前回と比較すると，マネジメント系分野からの出題が1問減って，ストラテジ系分野からの出題が1問増えました。大分類ごとの出題率と出題数は，次のとおりです。

出題テーマ	出題率	出題数
基礎理論	10.0%	3問
コンピュータシステム	16.6%	5問
技術要素	23.3%	7問
開発技術	6.7%	2問
プロジェクトマネジメント	6.7%	2問
サービスマネジメント	6.7%	2問
システム戦略	13.3%	4問
経営戦略	10.0%	3問
企業と法務	6.7%	2問



どの問題も，それぞれの分野における主要なテーマがとり上げられており，ITに関する総合的な基礎知識を問う試験として適切な問題構成といえるでしょう。最新の技術や用語を問う新しいテーマは，最近はほとんど出題されていません。今回初めて出題された問題も，そのテーマは，タスクスケジューリング，DBMSの媒体障害の対応，インスペクション，アーンドバリュー分析，提案依頼書（RFP），SWOT分析，BCP（事業継続計画）など，これまでに情報処理技術者試験のいずれかの試験区分でとり上げられたことのあるテーマがほとんどでした。

いずれかの試験区分で過去に出題されたことのある問題（過去問題）が17問，新しい問題は13問でした。過去問題は，ほとんどが平成19年度～平成22年度の問題で，新試験制度移行後の過去問題としては，平成22年度の問題が最も多く出題されています。また，試験区分に注目すると，応用情報技術者試験の過去問題が最も多く，高度の各試験区分の午前Ⅱ試験の過去問題からも少しずつ出題されています。

新しい問題は，マネジメント系分野やストラテジ系分野に多く見られます。さらに，マネジメント系分野やストラテジ系分野の過去問題には，平成21年度以降の比較的新しい問題が多いという特徴があります。

2　難易度の特徴

午前Ⅰ試験では，出題範囲のすべての分野において，技術レベル3までの出題が想定されます。今回は，難易度の高い問題は少なく，技術レベル3として適切な内容といえます。技術レベル4までの出題が想定される高度の各試験区分の午前Ⅱ試験とは明確な違いがあります。また，用語や用語の定義を問う問題が多く，計算問題などの時間のかかる問題が少なかったので，解答時間が不足するような，時間的難易度も決して高くはなかったと思われます。したがって，全体的な難易度は，標準，もしくはやや易しいレベルと考えられます。広い出題範囲の中から偏ることなく出題されているので，時間配分に注意して，学習した分野の問題を確実に得点に結び付けることができれば，60点を超えることは決して難しくはありません。

ただし，難易度は，受験者の得意分野や実務経験によって，その感じ方は異なります。システム開発に携わり，基本情報技術者試験，応用情報技術者試験，高度試験と段階的に受験してきた方にとっては，テクノロジ系を中心に出題される午前Ⅰ試験はそれほど難しくは感じないでしょう。一方，マネジメントを専門としていて，段階を踏まずに高度試験を受験した方にとっては，テクノロジ系中心の問題は，難しく感じるかもしれません。

3　問題テーマ難易度一覧表

問	問題テーマ	大分類テーマ	中分類テーマ	難易度
1	論理演算	基礎理論	基礎理論	B
2	パリティビット	基礎理論	基礎理論	B
3	再帰関数	基礎理論	アルゴリズムとプログラミング	C
4	キャッシュメモリの書込み	コンピュータシステム	コンピュータ構成要素	B
5	RAID	コンピュータシステム	システム構成要素	A
6	クライアントサーバ	コンピュータシステム	システム構成要素	C
7	タスクスケジューリング	コンピュータシステム	ソフトウェア	B
8	仮想記憶	コンピュータシステム	ソフトウェア	A
9	テクスチャマッピング	技術要素	マルチメディア	A
10	トランザクションの特性	技術要素	データベース	A
11	媒体障害回復	技術要素	データベース	B
12	LAN間接続装置	技術要素	ネットワーク	A
13	ARP	技術要素	ネットワーク	B
14	ハッシュ関数	技術要素	セキュリティ	A
15	Webサイトのセキュリティ	技術要素	セキュリティ	B
16	安全性／信頼性設計	開発技術	システム開発技術	A
17	レビュー技法	開発技術	システム開発技術	B
18	アーンドバリュー分析	プロジェクトマネジメント	プロジェクトマネジメント	B
19	ファンクションポイント法	プロジェクトマネジメント	プロジェクトマネジメント	A
20	レプリケーション	サービスマネジメント	サービスマネジメント	B
21	システムテスト監査	サービスマネジメント	システム監査	B
22	BCP	システム戦略	システム戦略	C
23	投資対効果指標	システム戦略	システム戦略	B
24	UML	システム戦略	システム戦略	A
25	RFP	システム戦略	システム企画	B
26	SWOT分析	経営戦略	経営戦略マネジメント	B
27	経営管理手法	経営戦略	経営戦略マネジメント	A
28	EDI情報表現規約	経営戦略	ビジネスインダストリ	B
29	在庫管理	企業と法務	企業活動	B
30	著作権	企業と法務	法務	B

注）難易度は3段階評価で，Cが難，Aが易を意味する。

目次

マネジメント系

ストラテジ系

問題文中で共通に使用される表記ルール

各問題文中に注記がない限り，次の表記ルールが適用されているものとする。

図記号	説明
	論理積素子（AND）
	否定論理積素子（NAND）
	論理和素子（OR）
	否定論理和素子（NOR）
	排他的論理和素子（XOR）
	論理一致素子
	バッファ
	論理否定器（NOT）
	スリーステートバッファ

注　入力部又は出力部に示されている○印は，論理状態の反転又は否定を表す。

テクノロジ系

分類1

基礎理論

分類 1．基礎理論の概要

基礎理論は，コンピュータを構成する仕組みの基礎となる考え方である。基礎理論は，数学で考案され，数学で表現されている。つまり，コンピュータは数学によって成り立っているといえる。基礎理論において，直接的に数学を適用しているのが狭義での基礎理論であり，間接的に数学を適用しているのがアルゴリズムである。

狭義での基礎理論には，データ表現，計算基礎理論，プログラム言語の基礎理論，応用数学が含まれる。数値データだけでなく，文字データや音声データ，画像データなど，コンピュータが扱うデータはすべて2進数で表現される。このディジタルデータの処理方法を扱う数学理論が，オートマトンである。コンピュータの処理手順（プログラム）は，CやCOBOLのようなアルファベットを用いたプログラミング言語で作成されるが，コンピュータは機械語しか理解できない。そこで，字句解析や構文解析を適用して，機械語に変換するのである。

どんなに小さな処理であっても，コンピュータで処理をさせるにはプログラムが必要となる。そのため，一般的な処理は，データ構造とアルゴリズムとして公式化されている。メモリを効率的に使用するためのデータ構造として，配列や線形リスト，木構造，スタックなどがある。そして，これらのデータ構造に対するデータ処理の方法として，線形探索や2分探索，ソートなどのアルゴリズムがある。

1 基礎理論

2進数 ▶問1

0～9の数を用いて，10になったら桁が上がる10進数に対して，0と1のみを用いて，2になったら桁が上がる数値である。コンピュータ内部では整数，実数，文字，画像，音声，プログラムといった各種の情報は，すべて2進数で保持される。2進数は，10進数と比べて桁数が多く扱いにくいため，2進数との変換が容易で少ない桁数で表すことができる8進数や16進数で表現されることも多い。

文字表現

文字を数値で表現する。例えば，ASCIIコードであれば，“A”は65が，“j”は106が対応する。この対応する数値を文字コードという。文字コードにはさまざまな種類があり，漢字のように文字数が多い場合は，1バイトでは表現することができない。このため，英数字と記号，制御コード以外の多国語はマルチバイト（複数バイト）の文字コードを利用する場合が多い。

画像表現

画像は，色の付いた画素（ピクセル）を縦横に並べ，色に対応させた数値を一つひとつの画素に割り当てて表現する。表現する色の数によって色を対応させるのに必要なビット数は異なり，白黒の2値画像であれば1ビット，256色であれば8ビット，65,536色であれば16ビットが必要となる。通常，画素数と色を表現するために必要なビット数の積によって画像を表現するのに必要なビット数を求める。例えば，1,024×768ピクセルの画像に65,536色を割り当てれば，必要なビット数は

$$1{,}024 \times 768 \times 16 = 12{,}582{,}912 \text{ [ビット]}$$

となる。

音声表現

音声のようなアナログ信号をコンピュータ上で扱う場合，アナログ信号をディジタル信号に変換して記録する。ディジタル信号に記録する方式の一つにPCM（Pulse Code Modulation）がある。PCMでは標本化（サンプリング），量子化，符号化の三つのステップによって音声信号をディジタルデータに変換する。

・標本化：音声信号を一定の時間間隔で抽出する作業であり，1秒間に抽出する回数をサンプリング周波数（単位はHz）という。音質を劣化させずにディジタル化するためには，一般的に音声信号の最大周波数の2倍以上の周波数でサンプリングする必要がある。
・量子化：サンプリングした音声を離散的な数値に変換する作業である。量子化するビット数が大きければ，より詳細に音声を再現できる。
・符号化：量子化によって得た数値を2進数に変換する作業である。

情報量

情報の大きさを定量化した値である。例えば，コイン投げでは，表か裏かは，1ビットの情報量で表すことができる。情報量は，その事象が発生する生起確率に依存し，事象Eが発生したときに伝達される情報量$I(E)$は，事象Eの生起確率$P(E)$を用いて

$$I(E) = -\log_2 P(E)$$

と表現できる。また，n種類の事象$E_i(1 \leqq i \leqq n)$が発生し得る事象系E_iの平均情報量Hは各事象の情報量と生起確率E_iの加重平均，すなわち，

$$H = -\sum_{i=1}^{n} P(E_i)\log_2 P(E_i)$$

として表すことができる。この平均情報量をエントロピー（entropy）ともいう。

論理

ある事象に対して正しいか正しくないかを規定できる叙述のことを命題と呼ぶ。命題のうち，正しいと規定できることを「真（true）」と呼び，正しいと規定できないことを「偽（false）」と呼ぶ。真と偽のことを真理値といい，真理値表ではそれぞれ1と0で表現する。

命題は，次のような論理演算によって組み合わせることができる。

・論理積　：「かつ（AND）」の意味を持ち，真理値表では“∧”で表す。
・論理和　：「または（OR）」の意味を持ち，真理値表では“∨”で表す。
・論理否定：「でない（NOT）」の意味を持ち，真理値表では“¬”で表す。
・含意　　：「含む」の意味を持ち，真理値表では“⊃”で表す。

グラフ

点と線の集合によって構成される幾何学的な図形である。データ構造をグラフで表したり，構成する線に重みを付加したグラフでネットワークを表現したりする。グラフ（Graph：G）は，節（Node，ノード：N）と枝（Branch：B）の集合から構成され，

$G=(N,\ B)$

と表現することができる。節n_1，n_2を結ぶ枝を$b(n_1,\ n_2)$とすると，n_1，n_2を端点と呼び，n_1とn_2は隣接しているという。2節間の枝に向きが与えられているグラフを有向グラフといい，向きが与えられていないグラフを無向グラフという。$b(n_1,\ n_2)$である（n_1からn_2である）場合，n_1が始点，n_2が終点となる。また，枝が$b(n,\ n)$となるとき，自己ループという。

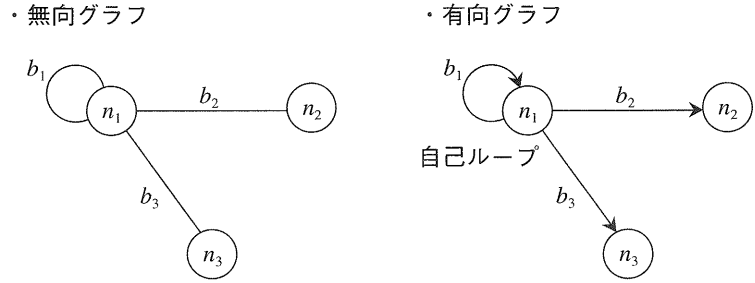


図1　無向グラフと有向グラフと自己ループ

オートマトン ▶問6

数学的な観点からコンピュータそのものをモデル化し，問題解決の処理手順を定式化したものである。得た入力から計算を実行して結果を出力するとともに停止するといった動作をモデル化した，仮想的な装置を指す。アルゴリズムを持つ問題はオートマトンによって計算が可能である。代表的な有限オートマトンとして，状態遷移図の例を示す。

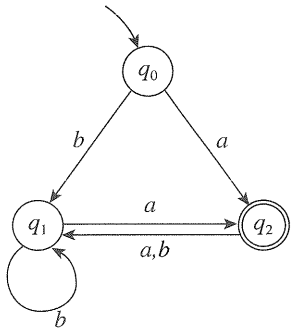


図2　状態遷移図の例

計算量

アルゴリズムの効率性を評価する尺度である。一般には，プログラムの実行が終了するまでに要する時間の量を表す。計算量はプログラムで用いるデータ数に依存するため，データ数Nの関数としてO（order）記法で，

$O(F(N))$

と表現する。ここで，$F(N)$の定数や係数を除外したうえで最も増加率の大きな項に着目して，$F(N)$の増加率を評価する。例えば，$F(N)$が

$F(N)=5N^3+2N^2+3N+5$

となるアルゴリズムの計算量は，N^3のオーダに比例すると考えることができる。Nのオーダに関する増加率の大小関係は，

$O(1)<O(\log_2 N)<O(N)<O(N\log_2 N)<O(N^c)<O(c^N)<O(N!)$　（cは定数）

となる。

構文規則

プログラムを構成する記号や単語の並べ方の規則である。プログラムを構成するときは，句構造文法の枠組みで表現することが一般的である。主な句構造文法として，BNFがある。

BNF（Backus-Naur Form；バッカス記法）

▶問7

プログラム言語の構文を形式的に記述する手段の一つであり，多くのプログラム言語の記述に使われている。「定義」を表す“：：=”，「または」を表す“|”，「非終端記号」を表す“＜　＞”の三つの記号および文字列を用いて構文規則（生成規則）を表す。例えば，SQLにおける表やグループ表を指定する表式（一部抜粋）は，BNFを用いて次のように表される。“[ ]”はこの記号に囲まれた箇所が省略可能であることを表す。また，“{ }”はこの記号に囲まれた箇所を 0 回以上繰り返してもよいことを表す。

＜表式＞：：=
　　＜FROM句＞
　　[＜WHERE句＞]
　　[＜GROUP BY句＞]
　　[＜HAVING句＞]

字句解析

コンパイラは，高水準言語で書かれたソースプログラムをコンパイル（翻訳）し，

ネイティブコードの目的プログラムを生成する言語処理プログラムである。コンパイラは，字句解析，構文解析，意味解析，最適化，コード生成という順に，各フェーズを段階的に処理しながら目的プログラムを生成する。字句解析は，原始プログラムの文字列を読み込み，原始プログラムの中に存在する識別子，定数，演算子などの文字列を，字句の種類に応じてトークンと呼ばれる単位に切り出す作業である。

構文解析

字句解析を行った結果に対して，字句の構成を調べることである。文の分類をするルーチンが文のトークンから文の種類を判別し，各文の処理プログラムへ制御を渡す。例えば，算術式の変数や定数は表に登録して表中の番号に置き換える，式は構文木などに変換するなどの処理が行われる。さらに，文の文法誤りなども検出し，エラーメッセージを出力する。構文解析した結果を表す主な方法に，逆ポーランド記法がある。

逆ポーランド記法（後置記法） ▶問8

字句解析の結果から，字句の構成を導出・表現する方法の一つである。逆ポーランド記法では，演算子を被演算項の後ろ（右側）に示す。例えば，「A＋B」であれば「AB＋」となる。次図に，“A＋B＊(C＋D)－E”を構文解析した結果を，逆ポーランド記法で表したものを示す。この図から，逆ポーランド記法は，構文木を「左から」「深さ優先順の」「後行順に」巡回した結果と一致していることが分かる。

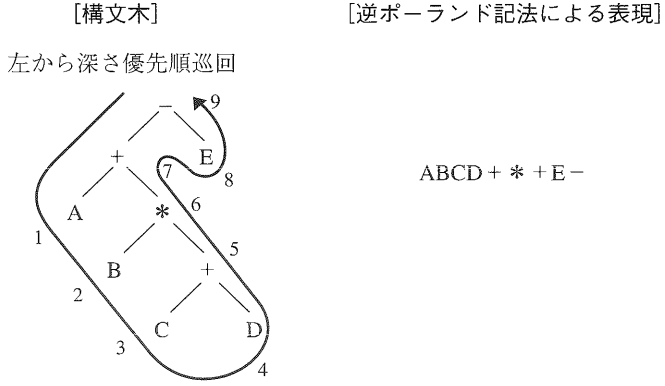


図3　構文木と逆ポーランド記法

相関係数 ▶問2

二つの変数xとyに関して，その観測値の組合せを$(x,\ y)$と表現すると，n個の組合せは次のようになる。

$$(x_1,\ y_1)\quad (x_2,\ y_2)\quad (x_3,\ y_3)\quad \cdots\cdots\quad (x_n,\ y_n)$$

これらの値を用いて，相関係数rを計算する公式は次のようになる。

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

$\bar{x}: x_i \sim x_n$の平均値

$\bar{y}: y_i \sim y_n$の平均値

rのとり得る値は，$-1 \leqq r \leqq +1$となる。正の相関が強いほど，rは$+1$に近くなり，負の相関が強いほど，rは-1に近くなる。相関がない場合は0に近くなる。

回帰分析

2組の観測値の系列間に因果関係を想定し，どちらか一方が原因（独立変数x），他方を結果（従属変数y）として

$$y=f(x)$$

となる関数fを求める手法である。特に$f(x)=ax+b$（a，bは定数）となる場合は線形回帰と呼ばれる。

2 アルゴリズムとプログラミング

抽象データ型

データの内部構造には触れずに，データに対する基本操作だけで定義されたデータ種別である。PUSHとPOPという二つの基本操作だけで定義されるスタックは，抽象データ型に該当する。抽象データ型は，次のような特徴や利点を持つ。

- カプセル化：データとデータを操作する手続きをひとまとめに定義することである。データを扱うモジュールはカプセル化された手続きを利用してデータにアクセスするため，アルゴリズムとデータを明確に分離でき，情報隠ぺいを実現できる。
- 情報隠ぺい：モジュールなどを設計する際に必要最低限の外部仕様のみを公開する概念である。情報を隠ぺいすることによってモジュールの独立性が高まり，変更に対する耐性を高めることができる。

線形リスト ▶問12

要素をポインタでつないだデータ構造である。各要素は，その要素のデータの値を格納する「データ部」と，次または前の要素を指すポインタを格納する「ポインタ部」から構成される。要素の並び順がその物理的な位置関係に依存せず，論理的な位置関係を実現できる。線形リストの最後の要素のポインタ部には，次の要素がないことを示す空値が格納される。一般的に使用される線形リストには，単方向リスト，双方向リスト，環状リストの3種類がある。

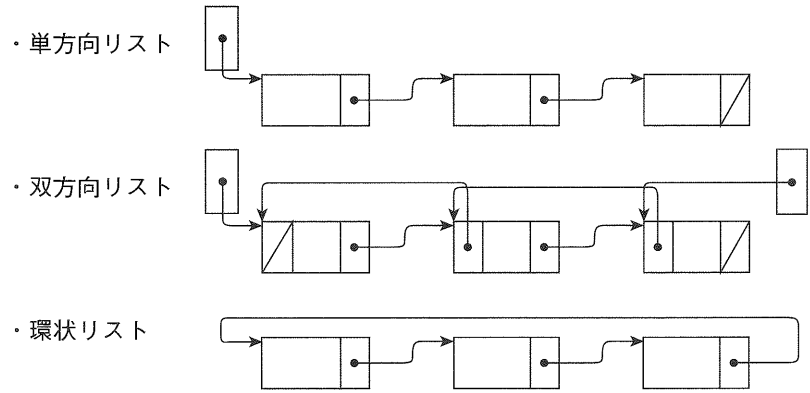


図4　線形リスト

線形リストでは，要素の挿入や削除の操作そのものに要する計算量が要素数に無関係である。単方向リストの場合であれば，挿入の場合には挿入する要素のポインタ部とその前の要素のポインタ部を更新するだけでよく，削除の場合には削除する要素の前の要素のポインタ部を更新するだけでよい。

木構造 ▶問13

複数の要素を階層的に並べたデータ構造である。階層構造を持つデータを表現するのに適している。木構造は，次のような要素で構成される。

- 節（node）：木を構成する要素の集合
- 根（root）：節の中で，特に階層の一番浅い位置にあるもの
- 辺または枝（branch）：節と節を結ぶ経路
- 親（parent）：枝で結ばれた節と節のうち，根から近いほうの節
- 子（child）：枝で結ばれた節と節のうち，根から遠いほうの節
- 葉（leaf）：子を持たない節

次図の点線で囲まれたこれらの節は，節eを根とした木とも考えられる。このような木の一部分を「部分木」という。

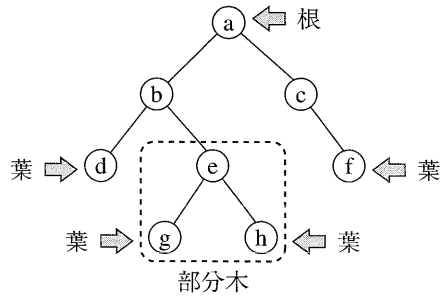


図5　木の構造

2分木（binary tree）は，各節が最大二つまでの子しか持てない木である。さらに，根からの深さの小さい順に，かつ同じ深さのところでは左から順に節を詰めた2分木を，完全2分木（complete binary tree）という。ヒープ（heap）は，すべての親子の節の値に一定の大小関係が成立している完全2分木である。

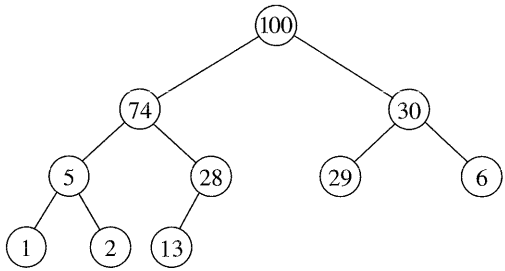


図6　「親の節が持つ値≧子の節が持つ値」というヒープの例

木構造において，節が持つ値を参照したり，検索したりする場合は，すべての節を巡回していく必要がある。

スタック　▶問14

後から入ったデータを先に取り出す後入れ先出し（LIFO：Last-In First-Out）の性質を持つデータ構造である。スタックに対する操作としてスタックにデータを格納するPUSHと，スタックからデータを取り出すPOPがある。

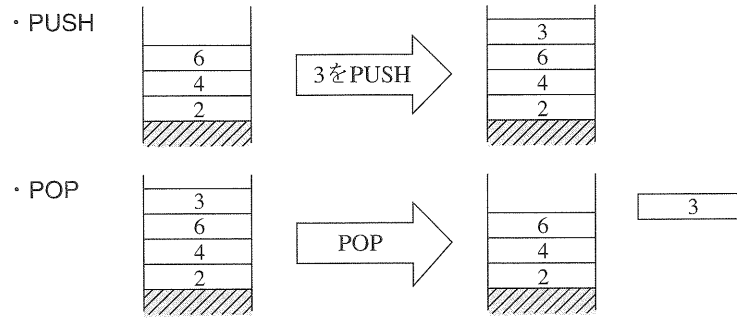


図7　スタックに対する操作

手続き型のプログラムにおいては，副プログラムを呼び出す際に呼出し側プログラムの環境（局所変数，引数，戻り番地など）をスタックにPUSHし，副プログラムの実行終了とともにPOPする。これによって，複数の副プログラムを入れ子で呼び出すような環境においても，呼出し元となるプログラムの副プログラムを呼び出した時点に正しく戻ることができる。

キュー

先に入ったデータを先に取り出す先入れ先出し（FIFO：First-In First-Out）の性質を持つデータ構造である。キューにデータを格納する操作をエンキュー，キューからデータを取り出す操作をデキューという。キューは到着順にサービスを提供するといった待ち行列の仕組みに用いられる。

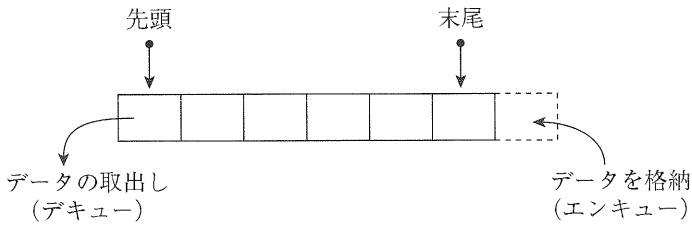


図8　キューに対する操作

線形探索 ▶問15

列の先頭の要素から順に1要素ずつ，目的の要素を探していく探索アルゴリズムである。配列や線形リストなどを使って行う。線形探索は，データ列のi (i=0，1，2，…，N−1)番目の要素を探索キーと比較し，一致しなければ(i+1)番目の要素と比較する手順を，「i ≧要素数となる」か「i 番目の要素と探索キーが一致する」まで繰り返す。

次図は探索キーと一致する要素がデータ列に存在する場合の例であるが，i =要素数(N)になるまで探索した場合は，探索キーと一致する要素がデータ列中に存在しなかったことになる。また，線形探索の平均比較回数は，比較回数の平均より，(N+1)／2となり，線形探索の計算量はデータ数Nに比例する。また，計算量はO(N)となる。

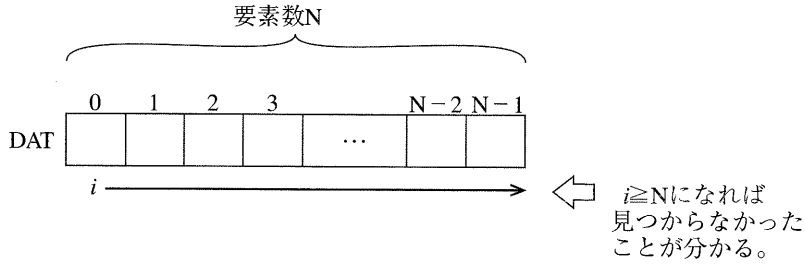


図9　線形探索の概要

2分探索 ▶問15

要素がキー値の昇順または降順に並んでいる配列において，探索範囲を2分しながら指定されたキー値を持つ要素を探し出すアルゴリズムである。探索範囲の中央に位置する値とキー値を比較し，比較結果に応じて探索範囲の半分を切り捨て，探索する範囲を次第に狭めていく。

2分探索開始時は，中央位置の添字（middle）は，探索範囲の先頭となる添字を下限（low），末尾となる添字を上限（high）とすると，(low+high)／2として求めることができる。次図の例では，データ列の中央は8番目の要素（DAT[7]）となる。中央の要素であるDAT[7]とkeyを比較するとDAT[7]>keyとなる。DAT[7]より右側にはDAT[7]よりも小さな値は格納されていないので，新しいhighの値を(middle−1)の6とすることで，探索範囲を狭めることができる。これらの操作を繰り返すことによって，探索値を見つけ出す。2分探索では，最大比較回数は($\log_2$N+1)回となり，計算量はO(logN)となる。

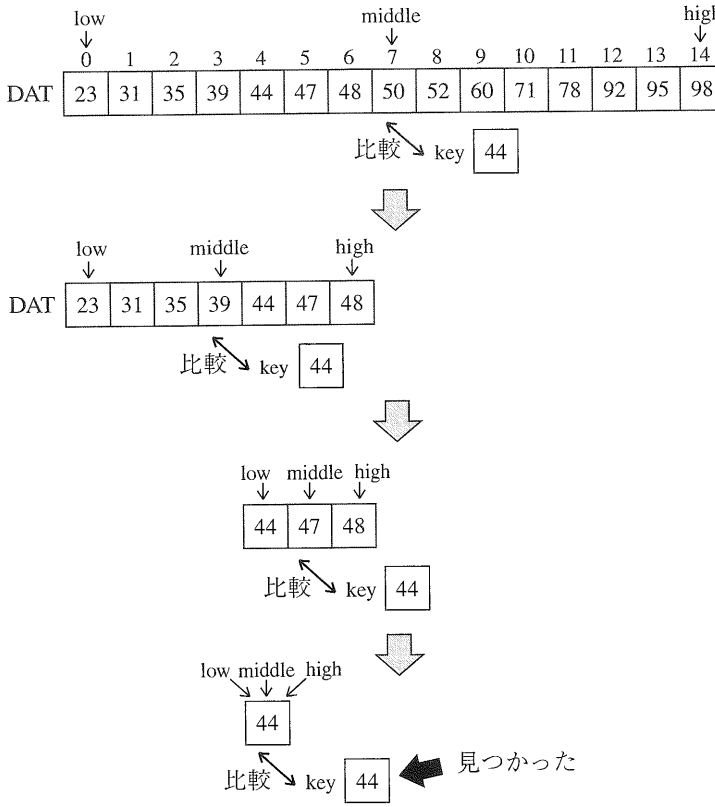

low
middle
high
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
DAT
23 31 35 39 44 47 48 50 52 60 71 78 92 95 98
比較
key
44
low
middle
high
DAT
23 31 35 39 44 47 48
比較
key
44
low middle high
44 47 48
比較
key
44
low middle high
44
比較
key
44
見つかった

図10　2分探索の概要

2分探索木

任意の節について，その左側の子およびその子孫の値はすべてその節の値よりも小さく，その右側の子およびその子孫の値はすべてその節の値よりも大きいという関係が成り立つ木である。節の値を探索キー値と比較し，両者の大小関係によって，

探索キー値＝節の値…探索成功

探索キー値＞節の値…左部分木には探索データが存在しない

探索キー値＜節の値…右部分木には探索データが存在しない

となるので，比較結果からどちらかの枝をたどることによって，配列を用いた2分探索と同様に探索対象を半分に絞り込むことができる。2分探索木の探索アルゴリズムでは，木の根の要素から順に，節の値と探索キー値の比較を行い，目的のデータを探す。探索対象となる2分探索木が完全2分木であれば，最大比較回数は$(\log_2 N + 1)$回となり，計算量は$O(\log N)$となる。

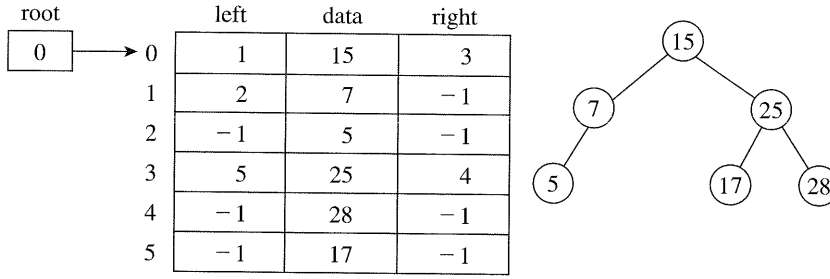


図11　2分探索木

ハッシュ表探索 ▶問15 ▶問16

キー値をハッシュ関数を用いてハッシュ値に変換し，目的のデータを探索する方法である。データへのポインタを格納する配列を「ハッシュ表」という。ハッシュ関数では，異なるキー値が同じハッシュ値に変換されることがある。これを「衝突」という。衝突が発生した（シノニム）場合は，同じハッシュ値を持つデータ同士で線形リストを作成する。探索時はハッシュ関数によってハッシュ値から線形探索によってリストをたどる。

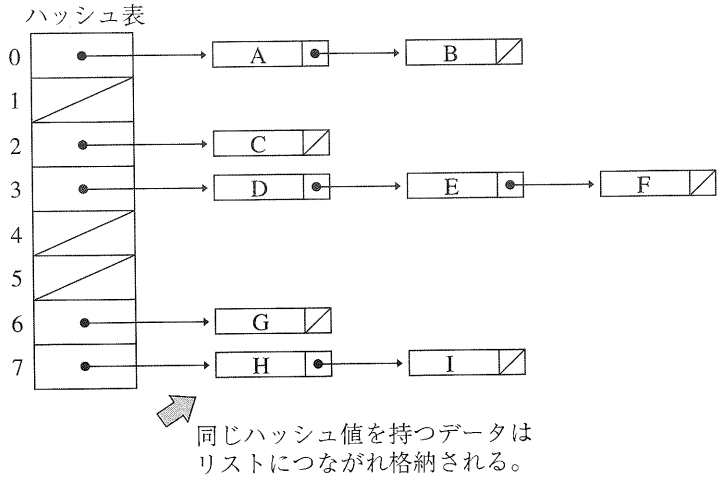


図12　ハッシュ表検索の例

選択法

選択法は，最も基本的な整列アルゴリズムである。整列とは，データをキー項目の値の順に並べ替えることである。整列開始時は配列すべてを整列対象とし，整列対象の中から最も小さな要素を選択し，整列対象の左端となる要素と交換する。これによって最小値が確定するので，整列対象から確定した要素を除いた要素群を新たな整列対象として同じ処理を繰り返す。整列対象の要素数が一つとなった（要素数－1回繰り返した）時点で，整列が完了する。整列アルゴリズムの計算量は比較回数に依存する。選択法の比較回数の合計は$(N(N-1)/2)$回となり，計算量は$O(N^2)$となる。

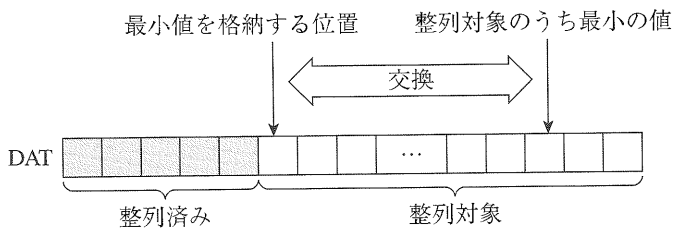


図13　選択法の概念

問題編

問1

□チャレンジ　□確認　□完璧　　出題年度：H21F 午前Ⅰ問1

2進数の表現で，2の補数を使用する理由はどれか。

ア　値が1のビット数を数えることで，ビット誤りを検出できる。
イ　減算を，負数の作成と加算処理で行うことができる。
ウ　除算を，減算の組合せで行うことができる。
エ　ビットの反転だけで，負数を求めることができる。

問2

□チャレンジ　□確認　□完璧　　出題年度：H23S 午前Ⅰ問1

相関係数に関する記述のうち，適切なものはどれか。

ア　すべての標本点が正の傾きをもつ直線上にあるときは，相関係数が＋1になる。
イ　変量間の関係が線形のときは，相関係数が0になる。
ウ　変量間の関係が非線形のときは，相関係数が負になる。
エ　無相関のときは，相関係数が－1になる。

問3

難　□チャレンジ　□確認　□完璧　　出題年度：H21S 午前Ⅰ問1

$(1+\alpha)^n$ の計算を，$1+n\times\alpha$ で近似計算ができる条件として，適切なものはどれか。

ア　$|\alpha|$ が1に比べて非常に小さい。
イ　$|\alpha|$ が n に比べて非常に大きい。
ウ　$|\alpha \div n|$ が1より大きい。
エ　$|n\times\alpha|$ が1より大きい。

答 1 2進数▶P.3 …… イ

多くのコンピュータが負数表現に2の補数を用いているのは，減算が加算回路で処理できる（補数器によって2の補数をとった後，加算すればよい）ようになるからである。

答 2 相関係数▶P.8 …… ア

相関係数は一般に記号 r で表され，$-1 \leqq r \leqq +1$の範囲の値をとる。正の相関が強いほど，r は$+1$に近くなり，負の相関が強いほど，r は-1 に近くなる（相関がない場合は0に近くなる）。

「すべての標本点が正の傾きをもつ直線上にあるときは，相関係数が$+1$になる」関係を正の完全相関といい，2属性の値(x, y)の間には

$$y = ax + b \quad (a > 0)$$

の関係が成立する。逆に負の完全相関$(a < 0)$の場合には，相関係数が-1となる。

イ　関係が線形であるとは，完全相関であることを示す。この場合，相関係数は$+1$か-1である。

ウ　関係が非線形の場合，$-1 < r < +1$となる。この記述だけでは，正負は特定できない。

エ　関係が無相関の場合，相関係数は0になる。

答 3 …… ア

$(1+a)^n$は，次のように展開される。

$$(1+a)^n = 1 + n \times a + \cdots\cdots + n \times a^{n-1} + a^n$$

a の絶対値が1に比べて非常に小さいとき，右辺の $a^2 \sim a^n$の各項の和は0に近づくため，$(1+a)^n$は$1 + n \times a$ に近似する。

問4 難 □チャレンジ □確認 □完璧

出題年度：H22S 午前Ⅰ問1

多数のクライアントが，LANに接続された1台のプリンタを共同利用するときの印刷要求から印刷完了までの所要時間を，待ち行列理論を適用して見積もる場合について考える。プリンタの運用方法や利用状況に関する記述のうち，M/M/1の待ち行列モデルの条件に**反しないもの**はどれか。

ア　一部のクライアントは，プリンタの空き具合を見ながら印刷要求をする。
イ　印刷の緊急性や印刷量の多少にかかわらず，先着順に印刷する。
ウ　印刷待ち文書の総量がプリンタのバッファサイズを超えるときは，一時的に受付を中断する。
エ　一つの印刷要求から印刷完了までの所要時間は，印刷の準備に要する一定時間と，印刷量に比例する時間の合計である。

問5 難 □チャレンジ □確認 □完璧

出題年度：H22F 午前Ⅰ問2

a，b，c，dの4文字からなるメッセージを符号化してビット列にする方法として表のア～エの4通りを考えた。この表はa，b，c，dの各1文字を符号化するときのビット列を表している。メッセージ中でのa，b，c，dの出現頻度は，それぞれ50%，30%，10%，10%であることが分かっている。符号化されたビット列から元のメッセージが一意に復号可能であって，ビット列の長さが最も短くなるものはどれか。

	a	b	c	d
ア	0	1	00	11
イ	0	01	10	11
ウ	0	10	110	111
エ	00	01	10	11

答 4 待ち行列理論▶P.47　M/M/1モデル▶P.47 …… イ

M/M/1の待ち行列モデルに従うためには，次のような条件を満たす必要がある。

・サービスを提供する窓口が一つだけである。
・待ち行列の長さに制限がない。
・同時にサービスを受けることができる処理要求は，一つである。
・先着順にサービスが提供され，順番が入れ替わることはない。
・待ち行列に並んだ処理要求が，途中で待ち行列から抜けることはない。
・サービス時間の分布が指数分布に従う（ランダムである）。
・到着がポアソン分布に従う（ランダムである）。

これらより，印刷における緊急性や量の多少に関係なく，先着順に印刷することは，M/M/1の待ち行列モデルの条件に反しない。

ア　印刷要求の発生タイミングがプリンタの状態に影響を受けたのでは，ランダム（ポアソン分布）にならない。
ウ　印刷要求の受付を中断する場合，待ち行列の長さに制限があることになる。
エ　M/M/1モデルの場合，一つの印刷要求から印刷完了までの所要時間は，印刷開始までの待ち時間と印刷に要する時間の合計である。待ち時間は待ち行列中の要求数に依存するため，一定とはならない。

答 5 …… ウ

まず，各選択肢が「一意に復号可能」という条件を満たすか否かを調べる。“ア”では，“00”というビット列を“aa”と“c”の2通りに解釈でき，“イ”では，“0110”というビット列を“ada”と“bc”の2通りに解釈できるため，一意に復号することができない。したがって，“ア”と“イ”は条件を満たさない。

“ウ”と“エ”は，元のメッセージを一意に復号できる。この二つについて，出現頻度をもとにビット列の平均長（期待値）を求めてみると，次のようになる。

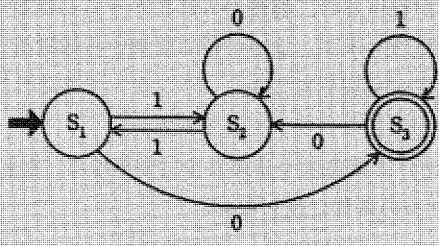
ウ　$0.5\times1+0.3\times2+0.1\times3+0.1\times3 = 1.7$ [ビット]
エ　$0.5\times2+0.3\times2+0.1\times2+0.1\times2 = 2$ [ビット]

よって，答えは“ウ”となる。

問 6

□チャレンジ　□確認　□完璧　　出題年度：H21S 午前Ⅰ問2

次に示す有限オートマトンが受理する入力列はどれか。ここで，S_1は初期状態を，S_3は受理状態を表している。



ア　1011　　イ　1100　　ウ　1101　　エ　1110

問 7

難 □チャレンジ　□確認　□完璧　　出題年度：H23S 午前Ⅰ問2

あるプログラム言語において，識別子（identifier）は，先頭が英字で始まり，それ以降に任意個の英数字が続く文字列である。これをBNFで定義したとき，aに入るものはどれか。

＜digit＞ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
＜letter＞ ::= A | B | C | … | X | Y | Z | a | b | c | … | x | y | z
＜identifier＞ ::= [a]

ア　＜letter＞ | ＜digit＞ | ＜identifier＞＜letter＞ | ＜identifier＞＜digit＞
イ　＜letter＞ | ＜digit＞ | ＜letter＞＜identifier＞ | ＜identifier＞＜digit＞
ウ　＜letter＞ | ＜identifier＞＜digit＞
エ　＜letter＞ | ＜identifier＞＜digit＞ | ＜identifier＞＜letter＞

答6 オートマトン▶P.5 …… ウ

有限オートマトンは，初期状態，有限個の状態，受理状態および有限個の記号と状態遷移関数で構成される数学的モデルで，状態遷移図などで表記される。

提示された有限オートマトンの状態遷移図では，S_1が初期状態，S_2が有限個の状態，S_3が受理状態を表し，0と1が有限個の記号，矢線が状態遷移関数による遷移先を示している。したがって，最初に初期状態であるS_1から入力列の記号である0または1を入力し，最後の記号を入力した段階で受理状態であるS_3に遷移して処理を終えれば，この有限オートマトンで受理されたことになる。

ア　1011の場合，S_1，S_2，S_2，S_1，S_2と遷移し，S_2で処理を終えるので受理されない。
イ　1100の場合，S_1，S_2，S_1，S_3，S_2と遷移し，S_2で処理を終えるので受理されない。
ウ　1101の場合，S_1，S_2，S_1，S_3，S_3と遷移し，S_3で処理を終えるので受理される。
エ　1110の場合，S_1，S_2，S_1，S_2，S_2と遷移し，S_2で処理を終えるので受理されない。

よって，この有限オートマトンで受理される入力列は“ウ”となる。

答7 BNF▶P.6 …… エ

<digit> は任意の数字を，<letter> は任意の英字を表している。

識別子は，「先頭が英字」の必要があるので，定義中に

<letter>

は単独で含まれるが，

<digit>

は単独で含まれてはならない。また，「任意個の英数字が続く」ためには，

識別子として成立している文字列 ＋ 数字
識別子として成立している文字列 ＋ 英字

が，どちらも識別子となるように定義すればよい。したがって，

<identifier> <digit>
<identifier> <letter>

が，どちらも定義中に含まれることになる。

以上が <identifier> の条件なので，これらを“|”によってつなげた

<letter> | <identifier> <digit> | <identifier> <letter>

が答えとなる。

問 8

□チャレンジ　□確認　□完璧　　出題年度：H23F 午前Ⅰ問1

式A＋B×Cの逆ポーランド表記法による表現として，適切なものはどれか。

ア　＋×CBA　　イ　×＋ABC　　ウ　ABC×＋　　エ　CBA＋×

問 9

□チャレンジ　□確認　□完璧　　出題年度：H21F 午前Ⅰ問2

誤り検出方式であるCRCに関する記述として，適切なものはどれか。

ア　検査用のデータは，検査対象のデータを生成多項式で処理して得られる1ビットの値である。

イ　受信側では，付加されてきた検査用のデータで検査対象のデータを割り，余りがなければ送信が正しかったと判断する。

ウ　送信側では，生成多項式を用いて検査対象のデータから検査用のデータを作り，これを検査対象のデータに付けて送信する。

エ　送信側と受信側では，異なる生成多項式が用いられる。

答 8 逆ポーランド記法▶P.7 …… ウ

逆ポーランド記法は，演算子を被演算子の右側に記述する後置表記法である。

提示された式では，“＋”よりも“×”のほうが演算の優先順位が高いので，次のように演算子を記述していけばよい。

提示式：A＋B×C

① B×C → BC×

② A＋① → A① ＋ → ABC×＋

答 9 …… ウ

CRC（Cyclic Redundancy Check；巡回冗長検査）とは，送信データを定められた生成多項式で割った余りを検査用のデータとして付加して送信し，受信側では余り（検査用のデータ）を含めた受信データを生成多項式で割り，余りが0になれば誤りがないと判断する誤り制御方式である。ビットのランダム誤りやバースト誤りを検出できるので，HDLC手順やCSMA/CD方式などのフレーム伝送における誤りチェック方式として採用されている。ITU-Tでは，CRC符号導出のための生成多項式として，次式を勧告している。なお，X^nはn番目のビットを表す。

生成多項式＝$X^{16}+X^{12}+X^5+1$

ア　1ビットの冗長ビット（検査用のデータ）を付加して誤り検出をするのは，パリティチェック方式である。

イ　検査対象のデータに検査用のデータを加えたものを生成多項式で割り，余りがなければ送信が正しかったと判断する。

エ　送信側と受信側では，同じ生成多項式が用いられる。

問10 難 □チャレンジ □確認 □完璧

出題年度：H23F 午前Ⅰ問2

符号長7ビット，情報ビット数4ビットのハミング符号による誤り訂正の方法を，次のとおりとする。

受信した7ビットの符号語 $x_1x_2x_3x_4x_5x_6x_7$（x_k=0又は1）に対して

$c_0 = x_1 + x_3 + x_5 + x_7$

$c_1 = x_2 + x_3 + x_6 + x_7$

$c_2 = x_4 + x_5 + x_6 + x_7$

（いずれもmod2での計算）

を計算し，c_0，c_1，c_2の中に少なくとも一つは0でないものがある場合には，

$i = c_0 + c_1 \times 2 + c_2 \times 4$

を求めて，左からiビット目を反転することによって誤りを訂正する。

受信した符号語が1000101であった場合，誤り訂正後の符号語はどれか。

ア　1000001　　イ　1000101　　ウ　1001101　　エ　1010101

問11 □チャレンジ □確認 □完璧

出題年度：H23F AP午前問4

サンプリング周波数40kHz，量子化ビット数16ビットでA/D変換したモノラル音声の1秒間のデータ量は，何kバイトとなるか。ここで，1kバイトは1,000バイトとする。

ア　20　　イ　40　　ウ　80　　エ　640

答10 ハミング符号▶P.45 …… エ

受信した符号語が1000101であるから，c_0 ～ c_2は

$c_0 = (1+0+1+1) \bmod 2 = 1$

$c_1 = (0+0+0+1) \bmod 2 = 1$

$c_2 = (0+1+0+1) \bmod 2 = 0$

となる。したがって，

$i = 1+1\times2+0\times4 = 3$

となり，3 ビット目を反転させて誤りを訂正すればよい。よって，誤りを訂正した符号語は

1010101

である。

答11 …… ウ

サンプリングしたデータの1秒当たりのデータ量（サイズ）は，

サンプリング周波数×量子化ビット数×チャネル数

で求められる。提示されている情報を整理すると

サンプリング周波数：40 kHz

量子化ビット数：16 ビット

チャネル数：1 ch（モノラルなので）

となるので，これらを用いて次のように計算すればよい。

40 × 1,000 × 16 × 1

= 640 × 1,000 [ビット]

= 80 × 1,000 [バイト]

= 80 [kバイト]

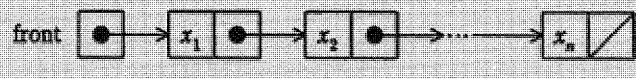
問12 基礎 □チャレンジ □確認 □完璧

出題年度：H21F 午前Ⅰ問3

n個の要素 x_1, x_2, …, x_n から成る連結リストに対して，新たな要素 x_{n+1} の末尾への追加に要する時間を$f(n)$とし，末尾の要素x_nの削除に要する時間を$g(n)$とする。nが非常に大きいとき，実装方法1と実装方法2における$\frac{g(n)}{f(n)}$の挙動として，適切なものはどれか。

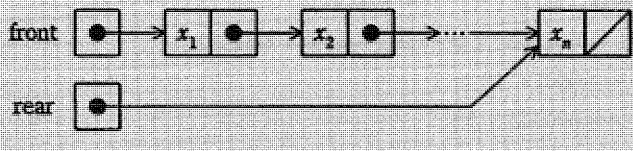
〔実装方法1〕

先頭のセルを指すポインタ型の変数frontだけをもつ。



〔実装方法2〕

先頭のセルを指すポインタ型の変数frontと，末尾のセルを指すポインタ型の変数rearを併せもつ。



	実装方法1	実装方法2
ア	ほぼ1になる。	ほぼ1になる。
イ	ほぼ1になる。	ほぼnに比例する。
ウ	ほぼnに比例する。	ほぼ1になる。
エ	ほぼnに比例する。	ほぼnに比例する。

答12 線形リスト ▶P.9 …… イ

各実装方法について評価してみると，次のようになる。

(実装方法1)

末尾への追加においても末尾の削除においても，どちらも先頭（front）から順にn個の要素をたどり，末尾に到達する必要がある。よって，処理時間はどちらもnに依存し，オーダは$O(n)$と表せる。

以上を用いると，

「$\dfrac{g(n)}{f(n)}$は，ほぼ1になる」

と評価できる。

(実装方法2)

末尾への追加は，rearによって末尾要素x_nに直接アクセスし，x_nのポインタとrearがともに新しい要素x_{n+1}を指すように変更すればよい。よって，処理時間はnに依存せず一定で，オーダは$O(1)$と表せる。

これに対し末尾の削除では，末尾要素x_nにアクセスするだけでは不十分で，その一つ前にある要素x_{n-1}にアクセスし，x_{n-1}のポインタを空値にし，rearをx_{n-1}を指すように変更しなければならない。x_{n-1}を得るためには，実装方法1と同様，先頭から順に末尾まで要素をたどる必要があるので，処理時間はnに依存し，オーダは$O(n)$と表せる。

以上を用いると，

「$\dfrac{g(n)}{f(n)}$は，ほぼ n に比例する」

といえる。

問13 難 □チャレンジ □確認 □完璧

出題年度：H23S AP午前問6

葉以外の節点はすべて二つの子をもち，根から葉までの深さがすべて等しい木を考える。この木に関する記述のうち，適切なものはどれか。ここで，深さとは根から葉に至るまでの枝の個数を表す。

ア　枝の個数がnならば，葉を含む節点の個数もnである。

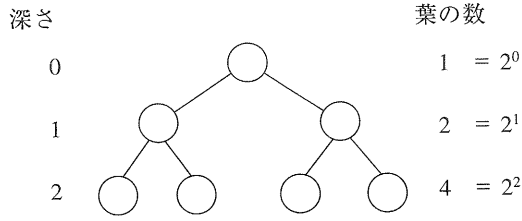
イ　木の深さがnならば，葉の個数は2^{n-1}である。

ウ　節点の個数がnならば，深さは$\log_2 n$である。

エ　葉の個数がnならば，葉以外の節点の個数は$n-1$である。

答13 木構造▶P.9 ..... エ

問題文で示されているのは，次のような木である。



問題で提示される木のイメージ

深さがdのときの葉の個数は2^dである。また，このときの「葉以外の節点の個数」は，「深さが$d-1$のときの節点の総数」に等しく，

$$1+2+4+\cdots+2^{d-1}=2^d-1$$

となる。

ここで$2^d=n$と置き換えると，「葉の個数がnのとき，葉以外の節点の個数は$n-1$」が成立する。

ア　枝の数がnならば，節点の個数は$n+1$となる。

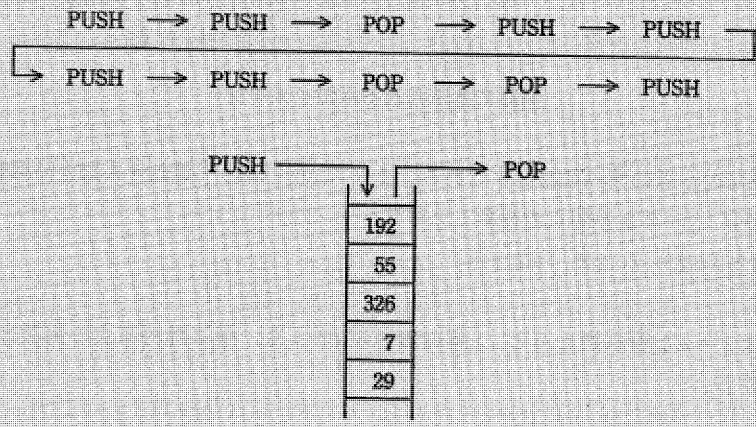
イ　木の深さがnのとき，葉の個数は$2n$となる。

ウ　深さがdの木の場合，節点の個数nは，$2^{d+1}-1$となる。これを書き換えると$2^{d+1}=n+1$となり，両辺について2の対数をとると，$d=\log_2(n+1)-1$のような関係が得られる。

問14

□チャレンジ　□確認　□完璧　　出題年度：H23S AP午前問7

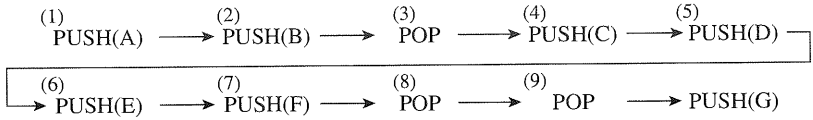
PUSH命令でスタックにデータを入れ，POP命令でスタックからデータを取り出す。動作中のプログラムにおいて，ある状態から次の順で10個の命令を実行したとき，スタックの中のデータは図のようになった。1番目のPUSH命令でスタックに入れたデータはどれか。



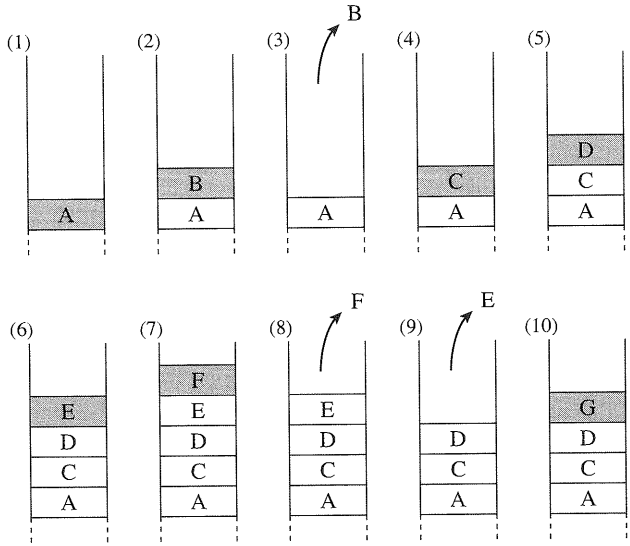
ア　29　　イ　7　　ウ　326　　エ　55

答14 スタック▶P.10 ……イ

各PUSH命令によって入れられたデータを，先頭のPUSH命令から仮にA，B，C，D，E，F，Gとし，一連のPUSH命令，POP命令によってスタックの内容が変化する様子を図に示す。



※ PUSH(x)は，データxをプッシュすることを表す。



スタックの変化の様子

PUSH(G)が終了した時点で，最初のPUSH命令（PUSH(A)）によってスタックに入れられたデータAは，スタックトップ（スタック内の最上段）から4番目にある。問題文の図でスタックトップから4番目のデータは“7”である。なお，“7”の下にある“29”は，PUSH(A)が行われる前に，すでにスタックに存在していた要素であると判断できる。

問15

□チャレンジ　□確認　□完璧　出題年度：H22F 午前Ⅰ問3

探索表の構成法を例とともにa～cに示す。探索の平均計算量が最も小さい探索手法の組合せはどれか。ここで，探索表のコードの空欄は表の空きを示す。

a　コード順に格納した探索表

コード	データ
120380	……
120381	……
120520	……
140140	……

b　コードの使用頻度順に格納した探索表

コード	データ
120381	……
140140	……
120520	……
120380	……

c　コードから一意に決まる場所に格納した探索表

コード	データ
120381	……
120520	……
140140	……
120380	……

	a	b	c
ア	2分探索	線形探索	ハッシュ表探索
イ	2分探索	ハッシュ表探索	線形探索
ウ	線形探索	2分探索	ハッシュ表探索
エ	線形探索	ハッシュ表探索	2分探索

問16

解説　□チャレンジ　□確認　□完璧　出題年度：H21S 午前Ⅰ問3，H23F 午前Ⅰ問3

自然数をキーとするデータを，ハッシュ表を用いて管理する。キーxのハッシュ関数$h(x)$を

$h(x)=x \bmod n$

とすると，キーaとbが衝突する条件はどれか。ここで，nはハッシュ表の大きさであり，$x \bmod n$はxをnで割った余りを表す。

ア　$a+b$がnの倍数　　イ　$a-b$がnの倍数

ウ　nが$a+b$の倍数　　エ　nが$a-b$の倍数

答15 線形探索▶P.12　2分探索▶P.12　ハッシュ表探索▶P.14 ····· ア

選択肢に挙げられている各探索手法の特徴は，次のようになる。

2分探索：表の中央の要素と探索キーを比較し，探索範囲を2分しながら探索する。計算量はデータ数nに対し$O(\log n)$と効率は良いが，探索する表はキー値の昇順または降順で整列されていなければならない

線形探索：表の先頭から順に比較照合しながら目的のデータを探索する。平均的な計算量は$O(n)$である。先頭に近いデータほど探索時間が短くなり，末尾に格納されたデータほど探索時間が長くなる

ハッシュ表探索：ハッシュ関数によってキー値からハッシュ値を得て，それをもとにデータの格納位置を決定する。（衝突を考えなければ）計算量はデータ数に依存しない，すなわち$O(1)$である

aのようにコード順に格納された探索表の場合，2分探索が適用できる。また，cの探索表は各データをコードから一意に決まる場所に格納しているので，ハッシュ表探索が適用できる。bの探索表は2分探索やハッシュ表探索を適用することはできないが，使用頻度の高いものほど先頭近くに格納されているので，線形探索の効率が良いと判断できる。

答16 ハッシュ表探索▶P.14 ····· イ

関数 h によるハッシュ値は「xをnで割った余り」で求められるので，ハッシュ値がtとなる場合，xの値は「nの倍数＋t」と表すことができる。

aとbが衝突するということは，ハッシュ値が等しくなるということなので，aは$(P \times n + t)$，bは$(Q \times n + t)$と表すことができる（P，Qは任意の整数）。よって，aとbの差は

$$(P \times n + t) - (Q \times n + t) = (P - Q) \times n$$

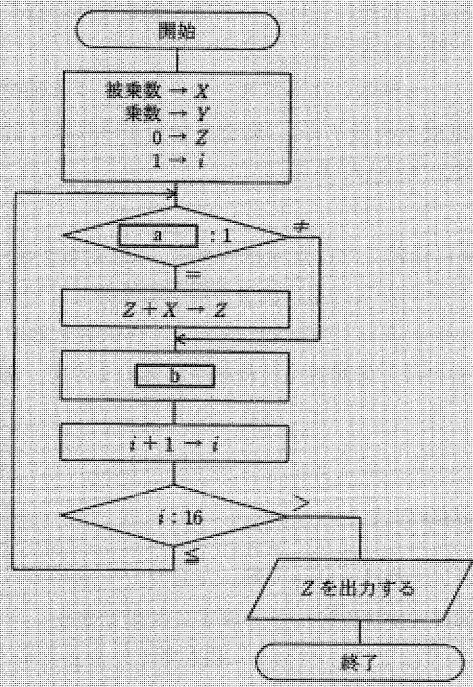
となり，結果は必ずnの倍数となる。

問17

難 □チャレンジ □確認 □完璧

出題年度：H22S 午前Ⅰ問2

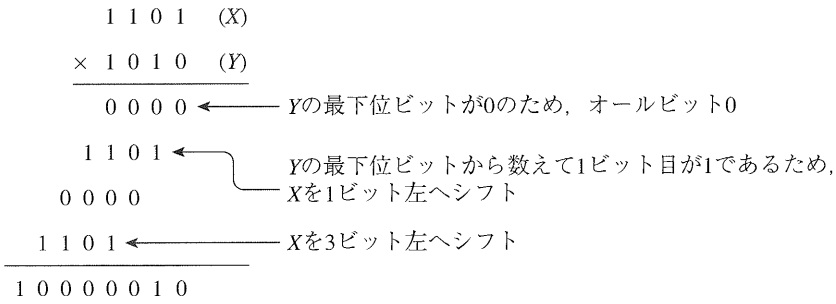
流れ図は，シフト演算と加算の繰返しによって2進整数の乗算を行う手順を表したものである。この流れ図中のa，bの組合せとして，適切なものはどれか。ここで，乗数と被乗数は符号なしの16ビットで表される。X，Y，Z は32ビットのレジスタであり，けた送りには論理シフトを用いる。最下位ビットを第0ビットと記す。



	a	b
ア	Y の第0ビット	X を1ビット左シフト，Y を1ビット右シフト
イ	Y の第0ビット	X を1ビット右シフト，Y を1ビット左シフト
ウ	Y の第15ビット	X を1ビット左シフト，Y を1ビット右シフト
エ	Y の第15ビット	X を1ビット右シフト，Y を1ビット左シフト

答17 ア

シフト演算を用いた乗算処理は，筆算のイメージで考えると分かりやすい。X（被乗数）を“1101”，Y（乗数）を“1010”とした筆算の例を示す。



筆算の例

図中の加算が行われる場合は，

“Yの第kビットが1の場合，Xを左にkビットシフトした数値を加算する”

と一般化できる。したがって，ループ中の空欄bでは，Yの最下位からkビット目が最下位ビットにくるように，Yを1ビット右シフトし，同時に，Xを左にkビットシフトした数値が求まるように，Xを1ビット左シフトすればよい。また，空欄aでは，Yの最下位ビット（第0ビット）が1であるか否かを検査すればよい。

問18

□チャレンジ　□確認　□完璧　　出題年度：H22S 午前Ⅰ問3

整形式（well-formed）のXML文書が妥当（valid）なXML文書である条件はどれか。

ア　DTDに適合している。
イ　XML宣言が完全に記述されている。
ウ　XMLデータを記述するための文法に従っている。
エ　エンティティ参照ができる。

答18 ア

ルートがただ一つである，開始タグと終了タグが対応している，といったように，XMLとしての最低限の形式（文法ルール）を満たしているものを，整形式（well-formed）のXML文書という。

整形式のXML文書のうち，DTDやXML Schemaなどのスキーマ言語によって記述された“文書型定義”の要件を満たしているものを，妥当（valid）なXML文書という。

イ　整形式のXML文書，妥当なXML文書のいずれにおいても，XML宣言は省略可能である。

ウ　整形式のXML文書であるための条件である。

エ　エンティティ参照（実体参照）は，文書内に文字を直接記述するのではなく，別に定義された内容を参照して埋め込む手段である。例えば“<”と記述することで，普通に記述したのではタグの先頭とみなされてしまう“<”を文字データ内に含めることができる。