6. アルゴリズム

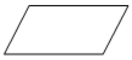
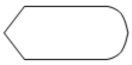
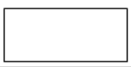
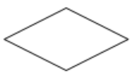
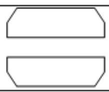
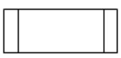
5 章で定義したデータ構造の中に格納されたデータに対して処理を行うことを考える。データ構造の構成方法と同様に、処理方法も適切に構成できれば、計算機資源の利用効率を向上し、ひいては、単位時間あたりの処理能力を向上させることができる。本章では、アルゴリズムの構築法や代表的なアルゴリズムについて説明する。

6.1 フローチャート

データを処理する手順をアルゴリズムという。このアルゴリズムを表現するには、フローチャート、擬似言語、各種プログラム言語などが用いられる。フローチャートと擬似言語は、プログラム言語に直接依存しないので、アルゴリズムをフローチャートで表しておけば、それを基にして高級言語（C 言語、COBOL 言語など）でプログラミングすることが可能となる。

(1) フローチャートで使用される記号

以下は、JIS X0121 規格で定義されているフローチャートである。

記号	名称	意味
	流れ線	データの制御の流れを示す。必要に応じて矢印をつける。
	端子	プログラムの始まりと終わり。
	データ	データの入出力やファイルに格納されているデータを示す。
	表示	画面にデータを表示
	処理	データの代入や計算の処理
	判断	条件によって処理を分木させる。
	ループ	ループの始まりと終わり。
	定義済み処理	サブルーチンなど、別のフローチャートによって定義された処理。

フローチャートの表記法を例 A-D により説明する。

変数

データを一時的に保管しておく領域。例(A)-(D)では、A が変数である。

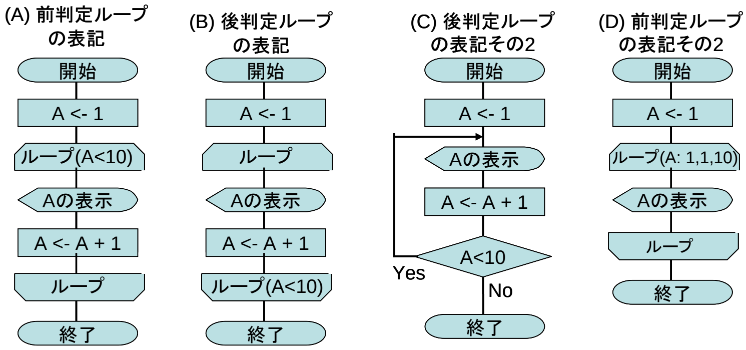
ループの前判定と後判定

(A)のように繰り返し条件の前(A>10)に書く場合と、(B)のように後ろに書く場合がある。条件を前に書いた場合、この条件を満たさないときにはループを抜けることになる。条件

を後ろに書いた場合は、ループ内の処理を一回は必ず実行される。

初期値、増分、終値

(D)のようにループにおいて A:1,1,10 は A の初期値=1、増分=1、終値=10 でループを実行することを表す。A=1 において、A を表示し、A に 1 を加え A=2 として A を表示し、...A=10 までこれを繰り返す。



7.2 計算量／メモリ使用量とオーダ記法

以降の節で、ソートとデータ探索について、いくつかのアルゴリズムを紹介するが、当然、それらのアルゴリズムには優劣がある。一般的に、アルゴリズムに対して最優先されるものは処理速度に直結する計算量であり、次点として、メモリ空間の使用量がある。これらについては、よく**オーダ記法(O 記法)**を用いて表される。この表記法では、データ数 N に比例して処理時間やメモリが必要となる場合 O(N)と記述する。通常、大規模データに対して考えるので、定数は取るに足らない小さな値として扱い、O(2N), O(N+100)も O(N)と書く。ただし、同じオーダ(例： O(N))のアルゴリズム同士の優劣を述べる場合、定数を用いることもある(例： O(aN)と O(bN)で a<b)。また、支配的な項のみを残して表記するので、N^2 の項と N の項があった場合、N^2 の項のみで表す。

以降で説明するソートアルゴリズムの計算量をまとめると、以下のようになる。、

$O(N^2)$	隣接交換法(バブルソート)
$O(N \log_2 N)$	クイックソート
$O(N \log_2 N)$	マージソート
$O(N^2)$	挿入法
$O(N^2)$	選択法
$O(N^{3/2})$	シェルソート
$O(N \log_2 N)$	ヒープソート

同様に、データ探索法のアルゴリズムについては、以下のようになる。

O(N)	線形探索

O(log₂ N)　　　2 分探索

7.3 整列(ソート)アルゴリズム

整列(ソート)は、データを昇順または降順に並べ替えることをいう。隣接交換法(バブルソート)、クイックソート、マージソート、挿入法、選択法、シェルソート、などがある。

(1) 隣接交換法（バブルソート）

[0] N 個の要素からなる一次元配列 A(j), j=1,2,...,N を用意する。

[1] p←N-1 とする。

[2] j=1,2,...,p それぞれについて A(j)>A(j+1)のとき、

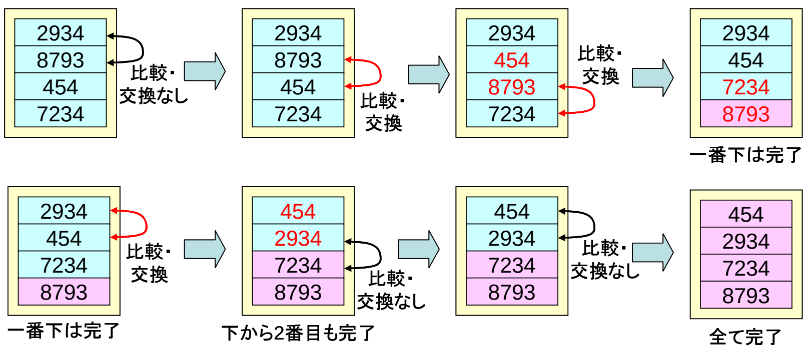
　b←A(j)

　A(j)←A(j+1)

　A(j+1)←b

[3] p←p-1

[4] p>0 のとき[2]へ行く、その他の場合終了する。

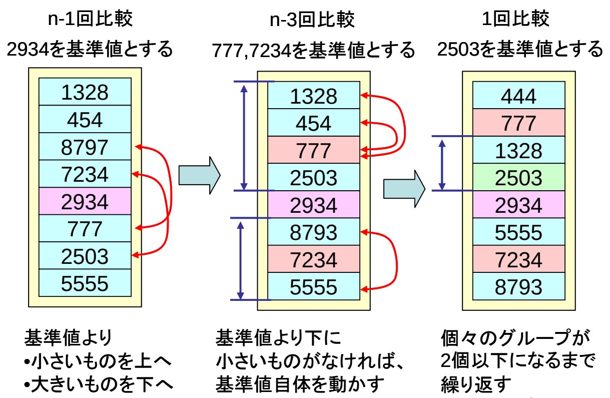


泡(＝小さい数値)が順番に浮かび上がっていく様子に例えて、バブルソート法とも呼ばれる。また、完了した部分までの順次の比較において交換が発生しない場合、すでに整列しているので、ソートを打ち切ることができる。

隣接交換法の比較回数は、1+2+...+N-1 = N(N-1)/2 となり、オーダ記法では $O(N^2)$となる。もともとの配列中のみでソートは可能なので、作業スペースとする追加のメモリ空間は不要となる。

(2) クイックソート

はじめに基準となる値を決め、基準値より小さい値のグループと大きいグループの二つに分ける。分割されたグループそれぞれに、この処理を繰り返し、個々のグループが 2 個以下になるまで繰り返す。

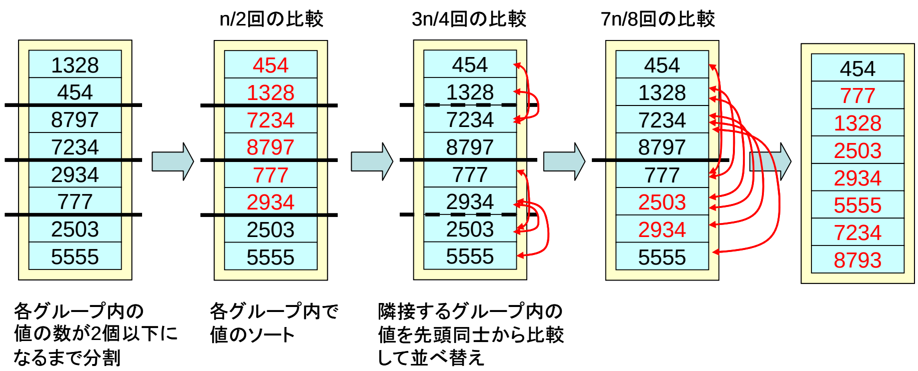


クイックソートでは、ソートする値のグループを分割して、それぞれのグループで基準値を求めることにより、次ステップで必要となる比較回数を減らしたり、総ステップ数を減らしたりしている。このように、問題を小さな問題に分割してゆくことで、処理の複雑度を低減させていく手法を**分割統治法**と呼ぶ。この分割統治法は、上手にプログラミングを行えば、関数の再帰呼び出しで実現できるため、プログラムのコードが簡潔になるという利点もある。

クイックソートの比較回数は、$x\, N\log_2 N + y$ となり(導出過程は複雑になるので省略)、オーダ記法では $O(N\log_2 N)$となる。もともとの配列中のみでソートは可能なので、作業スペースとする追加のメモリ空間は不要となる。基準値に最小／最大の数を非常に効率が悪くなるため、基準値を数個の候補の中央値を取るなどして、最小／最大の数を避けるようにすることもある。なお、常に最小／最大の値を基準値にした最悪時には、比較回数のオーダ記法は $O(N^2)$となる。

(3) マージソート

クイックソート同様、分割統治法を利用したソートアルゴリズム。ソートする値を 2 つのグループに分割し、それらのグループの値をソートした後、マージしてゆくというアルゴリズムである。



(4) その他のソートアルゴリズム

・挿入法： 未ソートのデータに対して、すでにソート済みのデータの中から挿入すべき箇所を特定し、そこに挿入するという作業を続けてゆく方法。

・選択法： 未ソートのデータの中から最大値／最小値を求めた後、それを対応する場所に格納してゆく方法。

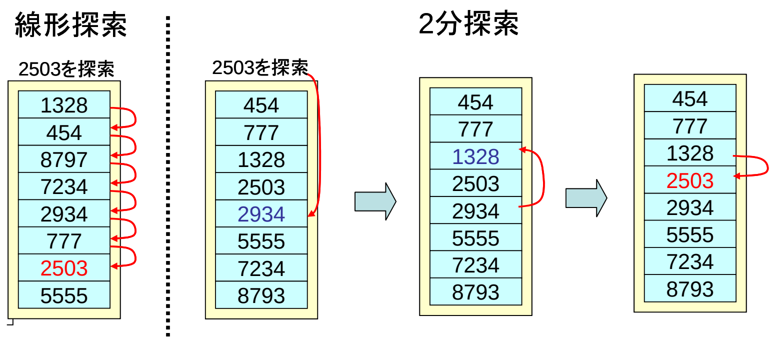
・シェルソート： 挿入法の改良。適当な間隔 h ごとのデータを別グループとし、そのグループごとのデータに対して挿入法を適用し、h をせばめてゆくという方法。

・ヒープソート： 最大値／最小値を求めるのに適したデータ構造であるヒープを利用して改良した選択法。最初に未ソートのデータをヒープに登録し、最大値／最小値を取り出して選択法を行う。

7.4 データ探索アルゴリズム

(1) 線形探索

　データが見つかるまで、配列の 1 番目の要素から順にデータを探索していく方法。データが見つかるか、配列の末尾まで探索を行い、データが存在しないことを確認したら終了する。



(2) 2 分探索

　あらかじめデータをソートしておいて、以下のように、配列の未探索部を 2 分割しながらの探索を行う。要は分割統治法の応用である。

[0]目的のデータキーを X とする。

[1]配列の中央 m を求める。m ← N/2

[2]要素 A(m)と x を比較する。

　　X = A(m)のとき、探索終了（ビンゴ！）[3]へ

　　X > A(m)のとき、

　　　探索範囲を決める：A(m+1) – A(N)

　　　m←(m+N)/2

　　　N-(m+1) が 1 のとき[3]へ、それ以外のとき[2]へ

X < A(m)のとき、

探索範囲を決める：A(1) – A(m-1)

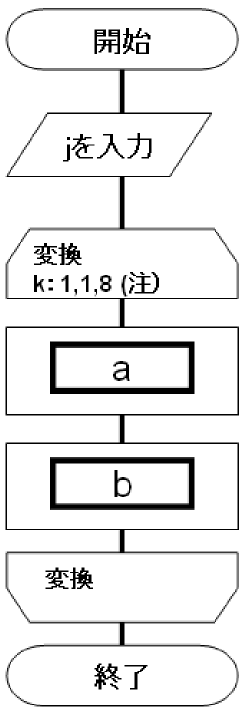
m←(m-1)/2

m-1 が 1 のとき[3]へ、それ以外のとき[2]へ

[3]終了

第 7 章の問題集

問題 7.1 次の流れ図は、10 進整数 j(0<j<100)を 2 進数に変換する処理を表している。2 進数は下位けたから順に、配列の要素 NISHIN(1)から NISHIN(8)に格納される。流れ図の a 及び b に入る処理はどれか。ここで、j div 2 は j を 2 で割った商の整数部分を、j mod 2 は j を 2 で割った余りを表す。



注）ループ端の繰り返し指定は、変数名：初期値、増分、終値を示す。

	a	B
ア	j div 2→j	j mod 2→NISHIN(k)
イ	j div 2→NISHIN(k)	j mod 2→j
ウ	j mod 2→j	j div 2 →NISNIH(k)
エ	j mod 2→NISHIN(k)	j div 2→j

(平 17 春午前問 1)

問題 7.2　配列Ａが図２の状態のとき、図１の流れ図を実行すると、配列Ｂが図の状態になった。図１の a に入るべき操作はどれか。ここで、配列 A,B の要素をそれぞれ A(i,j)、B(i,j) とする。

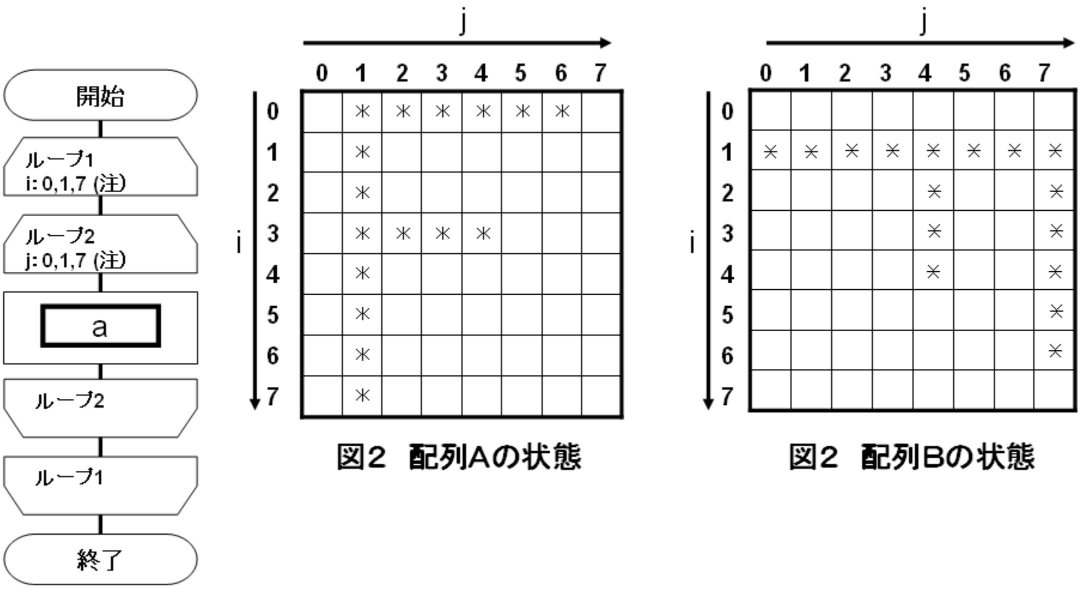


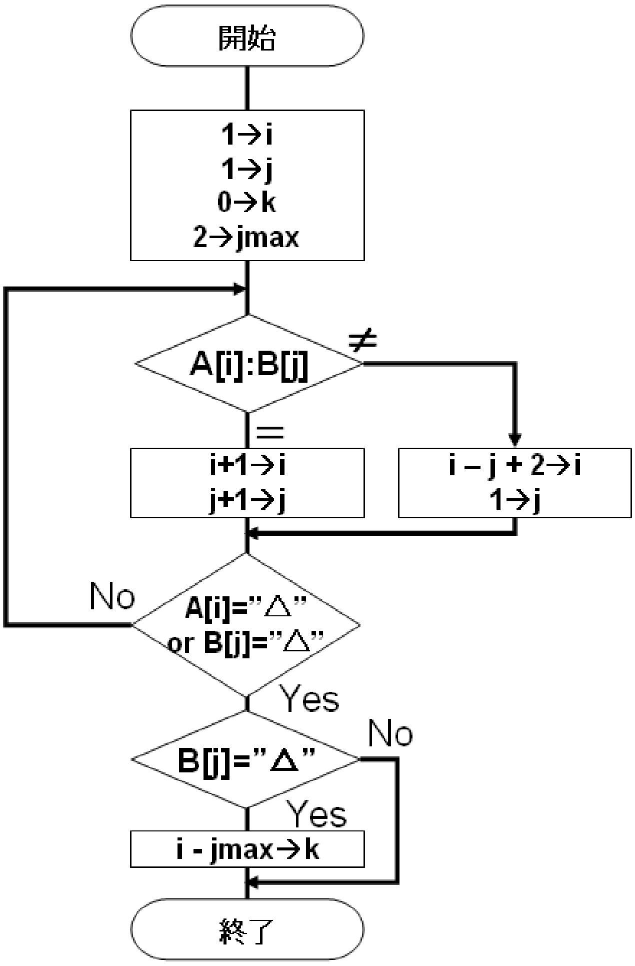
図2　配列Aの状態　　図2　配列Bの状態

注）ループ端の繰り返し指定は、変数名：初期値、増分、終値を示す。

ア　A(i,j)→B(i,7-j)　イ　A(i,j)→B(j,7-i)　ウ　A(i,j)→B(7-j,i)　エ　A(i,j)→B(7-i,7-j)

（平 15 春午前問 15）

問題 7.3　文字列 A が”aababx△”、文字列 B が”ab△”であるとき、流れ図の終了時点の k は幾らか。ここで、文字列の先頭の文字を１番目と数えるものとし、A[i]は A の i 番目の文字を、B[j]は B の j 番目の文字を、”△”は終端を示す文字を表す。



ア 0 イ　1　ウ 2 エ 4　　(平 19 春午前問 13)

問題 7.4　整列アルゴリズムの一つであるクイックソートの記述として、適切なものはどれか。

ア　対象集合から基準となる要素を選び、これよりも大きい要素の集合と小さい要素の集合を分割する。この操作を繰り返すことで、整列を行う。

イ　対象集合から最も小さい要素を順次とり出して、整列を行う。

ウ　対象集合から要素を順次取り出し、それまでに取り出した要素の集合の順序関係を保つよう挿入して、整列を行う。

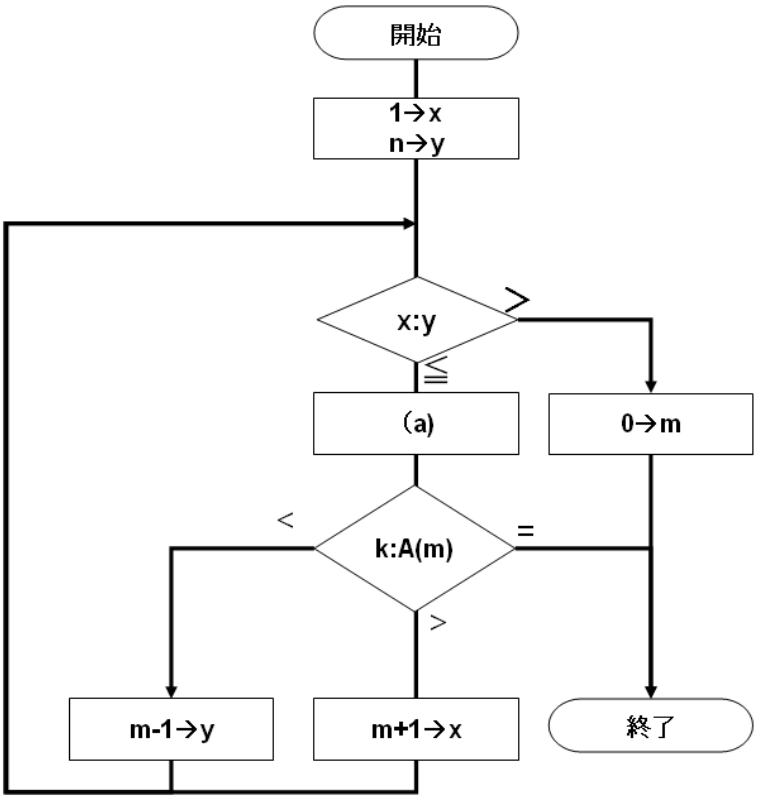
エ　隣り合う要素を比較し、逆順であれば交換して、整列を行う。

(平 14 秋午前問 12)

問題 7.5　6 個の数値 180、315、282、410、645、525 を並べ替える。手順 1 － 4 は途中までの手順を示したものである。手順 4 まで終わった時の結果はどれか。

手順 1：並びの左側から順に、数値の 1 の位の値によって 0-9 のグループに分ける。
手順 2:次の 0 のグループの数値を左側から順に取り出して並べ、その右側に 1 のグループ、以下順に 2-9 のグループの数値を並べていく。
手順 3：手順 2 で得られた数値の並びの左側から順に、数値の 10 の位によって 0-9 のグループに分ける。
手順 4：手順 2 と同様に、0 のグループの数値から順に並べる。
ここで、グループ内では、処理が行われた数値の左側から順に並べるものとする。
ア　180,282,315, 410, 525, 645　イ　315, 410, 525, 180, 282, 645
ウ　410, 315, 525, 645, 180, 282　　エ　645, 525, 410, 315, 282, 180
(平 18 春午前問 13)

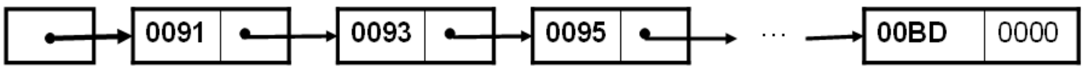
問題 7.6　昇順に整列済の配列要素 A(1), A(2), ..., A(n)から A(m)=k となる配列要素 A(m)の添字 m を 2 分探索法によってみつける処理を図に示す。終了時点で m=0 の場合は、A(m)＝k となる要素は存在しない。図中の a に入る式はどれか。ここで、/は小数点以下を切り捨てる除算を表す。



ア　(x+y)→m　イ　(x+y)/2→m　ウ　(x-y)/2→m　エ　(y-x)/2→m
(平 18 春午前問 14)

問題 7.7　リストの構造を以下の通りとする。

ROOT



(1) ROOT は、リストの先頭をさす。

(2) リストの要素は連続する 2 語からなる。第 1 語には値が第 2 語には次の要素へのポインタが格納されている。

(3) リストの各要素は値の昇順に連結されていて、値は全て異なる。最後の要素の第 2 語にはポインタとして 0000 が格納されている。

(4) 上の図の構造をもつリストが以下の図の通り主記憶の 00FF 番地から 0117 番地まで格納されている。00FF 番地は ROOT である。

(5) 1 語は 16 ビットからなり、語単位で番地がついている。

番地	内容	番地	内容	番地	内容	番地	内容
	…	0106	00A0	010E	00B0	0116	0099
00FF	0100	0107	010C	010F	0000	0117	0110
0100	0091	0108	00A9	0110	009B	0118	00A7
0101	010A	0109	0112	0111	0102	0119	0000
0102	009F	010A	0093	0112	00AB	011A	009C
0103	0106	010B	0104	0113	010E	011B	011C
0104	0095	010C		0114	00A2	011C	00A5
0105	0116	010D	0114	0115	0108	011D	0118

設問 1　010C 番地の内容は何か。

解答群　ア　0099　イ　00A1　ウ　00A3　エ　00A4　オ　00A5

設問 2　0110 番地及び 0111 番地からなる要素をリストから削除するには、[a]番地の内容を[b]番地に変えればよい。

解答群　ア　0101　イ　0102　ウ　0103　エ　0104　オ　0105　カ　0113　キ　0114
　　　　ク　0115　ケ　0116　コ　0117

設問 3 0118 番地から 011D 番地に格納されている 3 要素からなるサブリストと、設問 2 で要素を削除する前のリストを併合するには、[c]番地の内容の 011A に、[d]番地の内容を 0102 に、[e]番地の内容を 011C に、[f]番地の内容を 0108 にそれぞれかえればよい。

解答群　ア　0109　イ　010B　ウ　010D　エ　010F　オ　0111　カ　0113　キ　0115
　　　　ク　0117　ケ　0119　コ　011B

(平 17 春午後問 1)