【１】データ構造　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

コンピュータは、メモリ上に存在するデータを処理する。データの並べ方をデータ構造といい、適切なデータ構造にすることで効率良く処理することができる。

１．配列

同じ種類（基本データ型）のデータを並べて、多くの要素をまとめて扱うことができる。どの要素を使って処理をするのかを表すためには、配列名と順番に割り振った番号である[①　　　　　　]を指定する。［①　　　　　］は、0から始まるか1から始まるかはプログラム言語により異なる。

２．1次元配列

データ要素が1列に並んだ配列。

　　　　　　　配列名 添字

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| T | 30 | 40 | 10 | 50 | 20 |

T(2)を指定すれば40を表し、T(5)を指定すれば［②　　　　］を表すことになる。

３．2次元配列

データ要素を行と列で構成された配列。表記方法は、配列名（行番号,列番号）。

列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T | 1 | 2 | 3 | 4 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  | Ｂ |
| 4 |  | Ａ |  |  |

行

T(2,3)

データＡを利用するためには［③　　　　］、データＢを利用するためには［④　　　　］と表現する。

≪範例1≫

2次元の整数型配列aの各要素a( i , j )の値は、i+2jである。a(2,1)の値は幾つか。

ア　1 イ　2 ウ　3 エ　4

≪解答≫　エ

変数iと変数jに与えられた数字を代入して考えます。与えられた数字はa(2,1)ですから、iに2を、

jに1を代入し、i+2j＝2+2×1となり、答えが4になります。

≪範例2≫

２次元の整数型配列aの各要素a( i , j )の値は、2i＋jである。

　このとき、a(a(1,1)×2,a(2,2)＋1)の値は幾つか。

　ア　12 イ　13 　　ウ　18 エ　19

≪解答≫　エ

解答ポイントは、内側の2次元配列から値を求めていきます。まず、a(a(1,1)×2,a(2,2)＋1)

　のアンダーラインを求めるとa(1,1)は2×1+1＝3、a(2,2)は2×2+2＝6となります。求めた数値を

代入するとa(a(1,1)×2,a(2,2)＋1)＝a(3×2,6＋1)＝a(6,7)になります。2×6+7＝19となり、解答

はエです。

【２】リスト　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

　［①　　　　］部と［②　　　　］部で構成されているデータ構造。データを数珠つなぎに並べて［②　　　　］部が次の要素の場所（アドレス）を示している。

１．単方向リスト

次のデータへのポインタを１つだけ持つリスト構造。

|  |  |  |
| --- | --- | --- |
| アドレス | データ部 | ポインタ部 |
| 100 | ウィーン | 160 |
| 120 | トウキョウ | 180 |
| 140 | パリ | 999 |
| 160 | ミラノ | 0 |
| 180 | ロンドン | 100 |

　　　先頭データへのポインタ

120

連結イメージを表すと下記のようになる。下記［　］を埋めてみましょう。

連結イメージを作成してみると、データとして存在している「パリ」は、リストには連結されていないことがわかる。このように、必ずしも存在しているデータがすべてリストに連結されているとは限らない。

２．双方向リスト

次のデータへのポインタの他に、前のデータへのポインタも持っている構造。

下の表と連結イメージの［　］を埋めてみましょう。

|  |  |  |  |
| --- | --- | --- | --- |
| アドレス | データ | 次ポインタ | 前ポインタ |
| 100 | ウィーン | 160 | ［①　　　　］ |
| 120 | トウキョウ | 180 | 0 |
| 140 | パリ | 999 | 999 |
| 160 | ミラノ | 0 | ［②　　　　］ |
| 180 | ロンドン | 100 | ［③　　　　］ |

　　　先頭データへのポインタ

120

　連結イメージを表すと下記のようになる。下記［　］を埋めてみましょう。



３．環状（循環）リスト

ポインタによってデータが環状に連結されている構造。

120

　トウキョウ

　　　180

　　ロンドン

120

　　　180

　　　100

ウィーン

　　100

160

　　 160

　　ミラノ

［①　　　　］

　　　　※リスト最後尾のポインタ部が［②　　　　］のデータへのアドレスを持っている

【重要】

|  |  |  |
| --- | --- | --- |
| データ構造 | データアクセス方法 | データの挿入・削除 |
| 配　列 | 配列名と添字を指定することでデータへ直接アクセスすることができる。データを読み取るための作業は一定。 | 挿入・削除をする場所より後ろのデータは１つずつずらさなければならない。**挿入・削除をする場所によってずらす作業量が異なる。** |
| リスト | リストの先頭から順番にたどっていく必要がある。目的のデータを探索するため、探索する作業量が一定ではない。 | 挿入・削除する場所のポインタを書き換えるだけで挿入・削除ができる。**挿入・削除の場所がどこでも作業量が一定となる。** |

≪範例≫

　　　リストを二つの１次元配列で実現する。配列要素box［i］とnext［i］の対がリストの一つの要素

　　に対応し、box［i］に要素の値が入り、next［i］に次の要素の番号が入る。配列が図の状態の場合、

　　リストの３番目と４番目との間に値がＨである要素を挿入したときのnext［8］の値はどれか。こ

　　こで、next［0］がリストの先頭（1番目）の要素を指し、next［i］の値が0である要素はリストの

　　最後を示し、next［i］の値が空白である要素はリストに連結されていない。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ０ | １ | ２ | ３ | ４ | ５ | ６ | ７ | ８ | ９ |
| box |  | Ａ | Ｂ | Ｃ | Ｄ | Ｅ | Ｆ | Ｇ | Ｈ | Ｉ |
|  | ０ | １ | ２ | ３ | ４ | ５ | ６ | ７ | ８ | ９ |
| next | １ | ５ | ０ | ７ |  | ３ |  | ２ |  |  |

ア　３　　　　　　　イ　５　　　　　　　ウ　７　　　　　　　エ　８

　≪解答≫　ウ

　　連結イメージは下記の通り。

next[0]

1

next[1]

box[1]

5

A

next[5]

box[5]

3

E

next[3]

box[3]

next[7]

box[7]

7

C

2

G

next[2]

box[2]

0

B

ここにbox[8]のHが入るように

前のポインタnext[3]の値を変更する

next[3]

box[3]

next[8]

box[8]

8

C

7

Ｈ

next[7]

box[7]

2

G

…

…

　　以上のように挿入する位置の前のポインタ部を変更することでリストへの挿入ができる。この問題

　の問いはnext[8]の値であるため7となります。よって、解答はウです。