【１】スタックとキュー　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

１．スタック

②

③

|  |
| --- |
|  |
|  |
|  |
|  |

格納した順序と逆の順序でデータを取り出すことができるデータ構造。最後に格納したデータを最初に取り出すことができる。このような特徴を［①　　　　］（Last-In First-Out：後入先出）という。

スタックにデータを格納する操作を［②　　　　］と呼ぶ。

スタックからデータを取り出す操作を［③　　　　］と呼ぶ。

２．キュー

格納した順でデータを取り出すことができるデータ構造。最初に格納したデータを最初に取り出すことができる。このような特徴を［④　　　　］（First-In First-Out：先入先出）という。

キューにデータを格納する操作を［⑤　　　　］と呼ぶ。

キューからデータを取り出す操作を［⑥　　　　］と呼ぶ。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

⑤

⑥

　≪範例≫

　　空の状態のスタックとキューの二つのデータ構造がある。次の手続きを順に実行した場合，変数ｘに代入されるデータはどれか。ここで，手続で引用している関数は，次のとおりとする。

　[関数の定義]

　　push(y)：データｙをスタックに積む。

　　pop() ：データをスタックから取り出して，その値を返す。

　　enq(y) ：データｙをキューに挿入する。

　　deq()　：データをキューから取り出して，その値を返す。

　[手続]

　　push(a)

　　push(b)

　　enq(pop())

　　enq(c)

　　push(d)

　　push(deq())

　　x ← pop()

　　　ア　ａ　　　　　　イ　ｂ　　　　　　ウ　ｃ　　　　　　エ　ｄ

　≪解答≫　イ

　　手続きごとのスタック及びキューのデータは下記のとおりです。

|  |
| --- |
|  |
|  |
| b |
| a |

　　push(a)、push(b)はaとbをスタックに入れます。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

　　enq(pop())はpopしたbをキューに入れます。

|  |
| --- |
|  |
|  |
|  |
| a |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | b |

　　enq(c)はキューにcを、push(d)はスタックにdを入れます。

|  |
| --- |
|  |
|  |
| d |
| a |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | c | b |

　　push(deq())はキューから出したbをスタックに入れます。

|  |
| --- |
|  |
| b |
| d |
| a |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | ｃ |

　　x ← pop()でスタックから取り出したbを変数xに代入します。よって、解答はイになります。

【２】木構造　　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

要素同士の階層関係を表現するデータ構造であり、データ間の分岐関係を表現するのに適している。

１

２

４

５

６

８

９

３

７

10

［①　　　　］

［②　　　　］

［④　　　　］

［③　　　　］

|  |  |
| --- | --- |
| ［①　　　　　　　　］ | 一番上に位置する節（上図の①） |
| ［②　　　　　　　　］ | データを格納する領域（上図の①～⑩） |
| ［③　　　　　　　　］ | 節を結ぶ線 |
| ［④　　　　　　　　］ | 一番下に位置し、枝の分岐していない節（上図の④，⑥，⑧～⑩） |

１．２分木

各節の子の数が［⑤　　　　］の木のことであり、２つの子をそれぞれ左の子、右の子と区別する。また、左の子を根とする部分を［⑥　　　　］、右の子を根とする部分を［⑦　　　　］という。

節①の左部分木

１

２

４

５

７

８

３

６

節①の右部分木

２分木（例）

２．完全２分木

２分木のうち、根から葉までの経路の長さが［⑧　　　　］２分木。根から葉までの各径路の深さの差が［⑨　　　　］であり、かつ一番下は左から順に葉が詰められている２分木を完全２分木と定義。

１

２

４

５

６

３

７

４

２

５

８

６

７

３

１

深さ［⑨　 　］

完全２分木（例）

３．２分探索木

どの親子関係をみても左部分木のすべての節のキー値は親のキー値より小さく、右部分木のすべての節のキー値は親と等しいか、親よりも大きい木のこと。つまり、「左部分木のキー値［⑩　　］親のキー値［⑪　　］右部分木のキー値」の関係。

１

２

４

３

９

７

６

８

５

２分探索木（例）

４．ヒープ木

完全２分木において、各節が「親＜子」または「親＞子」という関係をもつ木。「親＜子」のときは根の値が［⑫　　　　］となり、「親＞子」のときは根の値が［⑬　　　　］となる。

６

７

５

１

２

３

４

８

９

値が大きい

値が小さい

ヒープ木（「親＞子」の例）

　≪範例≫

　　データ構造の一つである木構造に関する記述として、適切なものはどれか。

　　ア　階層の上位から下位に節点をたどることによって，データを取り出すことができる構造である。

　　イ　格納した順序でデータを取り出すことができる構造である。

　　ウ　格納した順序とは逆の順序でデータを取り出すことができる構造である。

　　エ　データ部と一つのポインタ部で構成されるセルをたどることによって，データを取り出すこと

　　　　ができる構造である。

　≪解答≫　ア

　　イ　キューに関する記述です。

　　ウ　スタックに関する記述です。

　　エ　リストに関する記述です。