

## 必須問題

問題 1 次のスタックとキューの説明を読み、各設問に答えよ。

[スタックについて]

スタックとは後入先出法 (LIFO) によりデータを管理するメモリ領域のことである。プログラム中で副プログラムや関数を呼び出すときのアドレス情報などを一時的に蓄える場所として使われる。ここでは、スタックにデータを格納する場合は push, スタックからデータを取り出す場合には pop を使う。

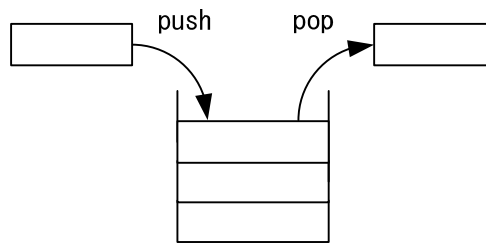


図 1 スタック

- push 書式: `push(データ)`  
例: `push(100)` ※スタックに 100 を格納
- pop 書式: `pop()`  
例: `x ← pop()` ※スタックからデータを取り出して変数 x へ代入

[キューについて]

キューとは先入先出法 (FIFO) によりデータを管理するメモリ領域のことである。OS の待ち行列管理などに用いられる。ここでは、キューにデータを格納する場合は enqueue, キューからデータを取り出す場合は dequeue を使う。

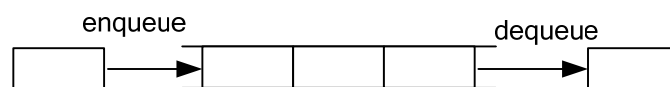


図 2 キュー

- enqueue 書式: `enqueue(データ)`  
例: `enqueue(100)` ※キューに 100 を格納
- dequeue 書式: `dequeue()`  
例: `x ← dequeue()` ※キューから取り出したデータを変数 x へ代入

＜設問 1＞ 次のスタックとキューを操作する処理を実行後，変数 x に格納される値を解答群から選べ。ここではスタックとキューは異なるメモリ領域を使っており，スタックおよびキューの領域は空の状態から各操作を始めるものとする。

(1) `push(100)`  
`push(200)`  
`push(300)`  
`push(400)`  
`x ← pop()`  
`x ← pop()`

(2) `enqueue(100)`  
`enqueue(200)`  
`enqueue(300)`  
`enqueue(400)`  
`x ← dequeue()`  
`x ← dequeue()`

(3) `enqueue(100)`  
`enqueue(200)`  
`push(300)`  
`push(400)`  
`push(dequeue())`  
`x ← pop()`

(1) ～ (3) の解答群

ア. 100                  イ. 200                  ウ. 300                  エ. 400

＜設問 2＞ 次の処理の説明を読み，処理の   に入れるべき適切な字句を解答群から選べ。なお，処理の「print」は，後に続く値を標準出力装置へ出力するための命令である。

[処理の説明]

- ・スタックおよびキューは空の状態から始める
- ・キューへ順番に 30，20，10 を格納する
- ・スタックを利用してキューの値を入れ替え，10，20，30 の順に出力する

[処理]

```
enqueue(30)
enqueue(20)
enqueue(10)
(4)
push(dequeue())
(5)
enqueue(pop())
print dequeue() ※ 10 が表示される
print dequeue() ※ 20 が表示される
print dequeue() ※ 30 が表示される
```

} ※ スタックを利用してデータを入れ替える

(4)，(5) の解答群

ア. `enqueue(dequeue())`                  イ. `enqueue(pop())`  
 ウ. `push(dequeue())`                  エ. `push(pop())`