

(1) 入力例

まず変数 `op` を入力する. `op` は線形リストの操作を選択するための変数であり, 以下にコマンド一覧と動作内容を図 1 に提示する.

コマンド	入力例	動作内容	補足
1	<code>v</code>	1 5 先頭に値 <code>x</code> を挿入	<code>insertFront</code>
2	<code>v</code>	2 8 末尾に値 <code>x</code> を挿入	<code>insertRear</code>
3		3 先頭の値を削除	<code>removeFront</code>
4		4 末尾の値を削除	<code>removeRear</code>
5	<code>v</code>	5 3 値 <code>v</code> を持つノードを削除	<code>removeSearch</code>
6	<code>v</code>	6 10 値 <code>v</code> の位置を検索して返す	<code>searchIndex</code> (先頭は 1)
7		7 現在のリストサイズを出力	<code>size</code>
8		8 リストのすべての値を出力	<code>print</code>
9		9 リストを全消去	<code>clear + size=0</code>
10		10 終了処理をしてプログラム終了	<code>terminate + return</code>
11		11 明示的終了 (条件終了)	無処理でループ終了

図 1 線形リストのコマンド一覧と動作内容

図 1 よりコマンドによって引数を必要とする場合がある. その時の変数は `v` である.

## (2) 出力例

op や v の入力を受け取り、関数により出力する。出力仕様を図 2 に示す。

コマン ド	概要	出力内容
1 v	先頭に v を挿入	v を出力
2 v	末尾に v を挿入	v を出力
3	先頭の要素を削除	出力なし（削除に成功しても出力しない）
4	末尾の要素を削除	出力なし
5 v	値 v の要素を削除	成功すれば 1、存在しなければ -1 を出力
6 v	値 v のインデックスを検索	見つければ 1 から始まるインデックス、なければ -1
7	リストのサイズ（要素数）を出力	int 型のサイズを出力
8	全要素を表示	すべてのノードの data を改行付きで出力
9	リストの全要素を削除	削除後のサイズ (= 0) を出力
10	終了	main() を終了し、プログラムを終了（出力なし）

図 2 コマンドとそれによる出力の仕様

## (3) 入力に対する出力結果の妥当性の説明

このプログラムは、単方向連結リストを操作するもので、ユーザーの入力に応じて要素の追加・削除・検索・表示・クリアなどを行い、その結果を標準出力に表示します。出力は、先頭または末尾への要素の挿入時には挿入した値を、指定値の削除（`removeSearch`）時には成功なら `1`、失敗なら `-1` を、値の検索（`searchIndex`）時には見つかった位置（1 始まり）または `-1` を、リストのサイズ取得時には現在の要素数を、リストの内容表示時には各要素を 1 行ずつ順に、全ノード削除（`clear`）後には `0` を出力します。すべての出力は整数で、1 行につき 1 つの数値を表示する簡素な形式であり、メッセージやラベルは付加されず、改行区切りで表示されます。以下は出力例です：`1 10` → `10`、`2 20` → `20`、`1 5` → `5`、`8` → `5 10 20`（各行）、`5 20` → `1`、`8` → `5 10`、`6 5`

アルゴリズムとデータ構造 問題○のプログラム実装の考え方  
学籍番号 22059 氏名 来間 空 提出日 2025 年 5 月 23 日

→ `1`、`7` → `2`、`9` → `0`、`10` → 終了。