

## アルゴリズムとデータ構造 B 課題 6

線形リストの操作を行う関数として、リストの末尾にノードを挿入する **InsertRear** 関数、リストの先頭ノードを削除する **RemoveFront** 関数および末尾ノードを削除する **RemoveRear** 関数、リスト中の全ノードを削除する（リストを空にする）**Clear** 関数を実装する。

### 問題 1

関数 **InsertRear()**, **RemoveFront()**, **RemoveRear()**をそれぞれ実装する。

**InsertRear** はリストの末尾ノードを探索し、動的に生成したノードを現在の末尾ノードの後ろに新たな末尾ノードとして挿入する。リストが空の場合は

**InsertFront** を呼び出す（リストの末尾ノードの探索は、**InsertbyIndex()**関数における **for** 文が参考になる）。挿入したノードを明示するために

**InsertRear** 内で課題 5 で作成した **Display** 関数を呼び出すこと。

**RemoveFront** はリストの先頭を現在の先頭ノードの次のノードに変更したうえで、元の先頭ノードの領域を解放する。リストが空の場合は何もしない。

**RemoveRear** は、リストが空の場合は何もしない。リストに先頭ノードのみがある場合は、**RemoveFront** を呼び出す。そうでない場合、リストの末尾ノードを探索し、末尾ノードの一つ前のノードを新たな末尾ノードに変更したうえで、元の末尾ノードの領域を解放する。

### 問題 2

線形リスト中の全ノードを削除する関数 **Clear()**を実装する。リストが空になるまで **RemoveFront** を呼び出す。その際、ノードの領域の解放がきちんと行えているかの確認の意味を込めて、**RemoveFront** を呼び出す前に先頭ノードの値やアドレスを表示すること。

### 問題 3

以下のプログラムによってノード数 5 の線形リストを作成する.

```
srand((unsigned)time(NULL)); //time.h が必要
int i, x;
for (int i = 0; i < 3; i++) {
    // シードに従って 100 より小さい乱数を生成して int x に代入する
    x = rand() % 100;
    printf("挿入するノードのデータ = %d\n", x);
    InsertRear(x);
}
x = rand() % 100;
printf("挿入するノードのデータ = %d\n", x);
InsertbyIndex(0, x);
x = rand() % 100;
printf("挿入するノードのデータ = %d\n", x);
InsertbyIndex(3, x);
```

線形リストを作成した後, **RemoveFront()**による先頭ノードの削除,  
**RemoveRear()**による末尾ノードの削除を一回ずつ順に実行する. 最後に,  
**Clear()**による全ノードの削除を行い, **head** ポインタの値 (アドレス) を出力  
してリストが空になっていることを確認するプログラムを記述せよ.  
全体の出力例は以下の通りである.

```
ノード数 5 の線形リストの作成
挿入するノードのデータ = 8
    0: 0x600001994040      8 0x0 <- edited node

挿入するノードのデータ = 17
    0: 0x600001994040      8 0x600001994050
    1: 0x600001994050     17 0x0 <- edited node

挿入するノードのデータ = 88
    0: 0x600001994040      8 0x600001994050
    1: 0x600001994050     17 0x600001994060
    2: 0x600001994060     88 0x0 <- edited node

挿入するノードのデータ = 29
    0: 0x600001994070     29 0x600001994040 <- edited node
```

1: 0x600001994040	8 0x600001994050
2: 0x600001994050	17 0x600001994060
3: 0x600001994060	88 0x0

挿入するノードのデータ = 29

i = 0

0: 0x600001994070	29 0x600001994040 <- p, q
1: 0x600001994040	8 0x600001994050
2: 0x600001994050	17 0x600001994060
3: 0x600001994060	88 0x0

i = 1

0: 0x600001994070	29 0x600001994040 <- q
1: 0x600001994040	8 0x600001994050 <- p
2: 0x600001994050	17 0x600001994060
3: 0x600001994060	88 0x0

i = 2

0: 0x600001994070	29 0x600001994040
1: 0x600001994040	8 0x600001994050 <- q
2: 0x600001994050	17 0x600001994060 <- p
3: 0x600001994060	88 0x0

i = 3

0: 0x600001994070	29 0x600001994040
1: 0x600001994040	8 0x600001994050
2: 0x600001994050	17 0x600001994060 <- q
3: 0x600001994060	88 0x0 <- p

0: 0x600001994070	29 0x600001994040
1: 0x600001994040	8 0x600001994050
2: 0x600001994050	17 0x600001994080
3: 0x600001994080	29 0x600001994060 <- edited node
4: 0x600001994060	88 0x0

-----

RemoveFront() で先頭のノードを削除

0: 0x600001994040	8 0x600001994050
1: 0x600001994050	17 0x600001994080
2: 0x600001994080	29 0x600001994060
3: 0x600001994060	88 0x0

-----

RemoveRear() で末尾のノードを削除

0: 0x600001994040	8 0x600001994050
-------------------	------------------

```
1: 0x600001994050    17 0x600001994080
2: 0x600001994080    29 0x0
```

-----

Clear() で全ノードを削除

```
free data:8, address:0x600001994040
free data:17, address:0x600001994050
free data:29, address:0x600001994080
```

-----

head = 0x0

## 課題の提出方法

問題を一連のプログラムとして作成し、完成したプログラム全体（適切にコメントを付ける，スクリーンショット不可），プログラムの実行結果（スクリーンショット）をレポートとしてまとめ，提出期限までに Teams 上で提出せよ．レポートのテンプレートファイル (word) は Teams に掲載してある．提出ファイル形式は PDF とする．

提出期限：2025 年 11 月 26 日 (水) 9:15

※12 月 1 日 (月) 23:59 に変更（課題 5 も同様）