

課題 6

アルゴリズムとデータ構造B

第14回

課題 5 解答例

問題 1 Remove関数, InsertbyIndex関数

2

- 前回スライド 4, 5 枚目の青枠の位置に下記を挿入
 - ✓ これ自体を関数化しても良い

```
// ポインタ r を使ってリスト全体を表示 (p, q が指しているノードを明示)
printf("i = %d\n", i);
for (j=0, r=head; r!=NULL; j++, r=r->next) {
    printf("%6d: %p %6d %p", j, r, r->data, r->next);
    if (r==p && r==q)
        printf(" <- p, q\n");
    else if (r==p)
        printf(" <- p\n");
    else if (r==q)
        printf(" <- q\n");
    else
        printf("\n");
}
```

```
void Display (struct Element *q) {  
    int i;  
    struct Element *p;  
    for (i=0,p=head; p!=NULL; i++,p=p->next) {  
        printf("%6d: %p %6d %p", i, p, p->data, p->next); // ここでは改行しない  
        if (p == q)  
            printf(" <- edited node¥n");  
        else  
            printf("¥n");  
    }  
}
```

```
// InsertFront 関数の適切な位置で Display 関数を呼び出す  
Display(p); // 挿入したノードのアドレスを渡す  
// Remove 関数の適切な位置で Display 関数を呼び出す  
Display(q); // 削除したノードの一つ前のノードのアドレスを渡す  
// InsertbyIndex 関数の適切な位置で Display 関数を呼び出す  
Display(r); // 挿入したノードのアドレスを渡す
```

課題 6

```
void InsertRear(int x);
```

- リストの末尾にノードを挿入する
- リストの末尾ノードを探索し、そのノードの後ろに新たな末尾ノードを挿入
 - ✓ ノードを動的に生成し、データ部（メンバ `data`）に引数 `x` を代入
 - ✓ ポインタ部（メンバ `next`）は末尾のため `NULL`
 - ✓ **現在の末尾ノードのポインタ部（メンバ `next`）に
新たな末尾ノード（生成したノード）のアドレスを代入**
- 末尾の探索：`struct Element` 型のポインタを用意し、`next` が `NULL` となるまでリストの `head` から順に探索
- 挿入したノードを明示するために課題5で作成した**Display**関数を呼び出すこと
- リストが空の場合は、`InsertFront` を呼び出す

```
void RemoveFront();
```

- 先頭ノードを削除する
- 先頭ノードを現在の先頭ノードの次のノードに変更
- 元の先頭ノードの領域を解放
- リストが空の場合は何もしない
- Remove 関数の中身にヒントがある

`void RemoveRear();`

- 末尾ノードを削除する
- 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
- それ以外の場合は、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、元の末尾ノードの領域を解放
 - ✓ 末尾の探索に現在のノードをポインタ `p`、一つ前のノードをポインタ `q` が指すループを使用


```
void Clear();
```

- 線形リスト中の全ノードを削除する
- リストが空になるまで（**head** が **NULL** になるまで）
順に **RemoveFront** により先頭ノードを削除する
 - ✓ 先頭ノードの値，アドレスを表示して正しいノードを削除（**free**）しているか確認する

```

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 = 26
  0: 0x6000013e8000      26 0x0 <- edited node

挿入するノードのデータ = 38
  0: 0x6000013e8000      26 0x6000013e8010
  1: 0x6000013e8010      38 0x0 <- edited node

挿入するノードのデータ = 70
  0: 0x6000013e8000      26 0x6000013e8010
  1: 0x6000013e8010      38 0x6000013e8020
  2: 0x6000013e8020      70 0x0 <- edited node

挿入するノードのデータ = 74
  0: 0x6000013e8030      74 0x6000013e8000 <- edited node
  1: 0x6000013e8000      26 0x6000013e8010
  2: 0x6000013e8010      38 0x6000013e8020
  3: 0x6000013e8020      70 0x0

挿入するノードのデータ = 74
i = 0
  0: 0x6000013e8030      74 0x6000013e8000 <- p, q
  1: 0x6000013e8000      26 0x6000013e8010
  2: 0x6000013e8010      38 0x6000013e8020
  3: 0x6000013e8020      70 0x0

i = 1
  0: 0x6000013e8030      74 0x6000013e8000 <- q
  1: 0x6000013e8000      26 0x6000013e8010 <- p
  2: 0x6000013e8010      38 0x6000013e8020
  3: 0x6000013e8020      70 0x0

i = 2
  0: 0x6000013e8030      74 0x6000013e8000
  1: 0x6000013e8000      26 0x6000013e8010 <- q
  2: 0x6000013e8010      38 0x6000013e8020 <- p
  3: 0x6000013e8020      70 0x0

i = 3
  0: 0x6000013e8030      74 0x6000013e8000
  1: 0x6000013e8000      26 0x6000013e8010
  2: 0x6000013e8010      38 0x6000013e8020 <- q
  3: 0x6000013e8020      70 0x0 <- p

  0: 0x6000013e8030      74 0x6000013e8000
  1: 0x6000013e8000      26 0x6000013e8010
  2: 0x6000013e8010      38 0x6000013e8040
  3: 0x6000013e8040      74 0x6000013e8020 <- edited node
  4: 0x6000013e8020      70 0x0

```

RemoveFront() で先頭のノードを削除

```

  0: 0x6000013e8000      26 0x6000013e8010
  1: 0x6000013e8010      38 0x6000013e8040
  2: 0x6000013e8040      74 0x6000013e8020
  3: 0x6000013e8020      70 0x0

```

RemoveRear() で末尾のノードを削除

```

  0: 0x6000013e8000      26 0x6000013e8010
  1: 0x6000013e8010      38 0x6000013e8040
  2: 0x6000013e8040      74 0x0

```

Clear() で全ノードを削除

```

free data:26, address:0x6000013e8000
free data:38, address:0x6000013e8010
free data:74, address:0x6000013e8040

```

head = 0x0

指示に従って問題を解き、
期限までにレポートを Teams 上で提出すること

- レポート提出期限：2025年~~11月26日（水）9:15~~
変更）12月1日（月）23:59 ※課題5も同じ期日とします

中間試験：12月8日 (月) 2コマ目

- 出席番号順に着席
- 試験時間：60分...といいつつチャイムなるまで延長
- 出題範囲：第1回～第16回
 - ✓ ポインタ，構造体，領域の動的確保，
テーブル（2次元配列），スタック，キュー，線形リスト

11/24 14回 課題6

11/26 15回 線形リストによる
スタック・キュー

12/01 16回 課題7

(12/02-12/04 中間試験の日)

12/08 17回 中間試験実施