

課題 6：SortedLinkedList クラスのメンバ関数 insert(), remove()を実装し， prac06\_skel.cpp に途中まで作成されているプログラムの動作を確認せよ。

```
#include <iostream>
using namespace std;
class SortedLinkedList {
    class Node { // privateなので, SortedLinkedListのメンバ関数以外からはアクセスできない
    public:
        int data; // ノードの値
        Node* next; // 次のノードへのポインタ
        Node(int num = 0, Node* p = NULL) { data = num; next = p; }
        ~Node() { cout << data << " is released" << endl; }
    };
    Node* head;
    void push(int x) { head = new Node(x, head); }
    void pop() { Node* node = head; head = head->next; delete node; }
public:
    SortedLinkedList() { head = NULL; } // 空リストの生成
    SortedLinkedList(int*, int*); // int配列の内容でリストを初期化
    ~SortedLinkedList() { clear(); }
    bool empty() const { return head == NULL; } // リストが空ならTrueを返す
    void clear() { while (head) pop(); }; // リストから全データを削除
    void printALL() const; // リストの全データを表示
    void insert(int n); // リストにデータnを追加
    void remove(int n); // リストからデータnを削除
    int size() const; // リストのデータ数を表示
};

// int配列の内容でリストを初期化
SortedLinkedList::SortedLinkedList(int* begin, int* end) {
    head = NULL;
    for (int* p = begin; p != end; p++) insert(*p);
}

// リストの全データを表示
void SortedLinkedList::printALL() const {
    for (Node* p = head; p; p = p->next)
        cout << p->data << "-->";
    cout << "END_OF_DATA" << endl;
}

// リストの要素数を返す
int SortedLinkedList::size() const {
    int n = 0;
    for (Node* p = head; p; p = p->next) n++;
    return n;
}

// リストに要素nを昇順になるように挿入する
void SortedLinkedList::insert(int n) {
}

// リストから値がnの要素の中で最も先頭にある要素を取り除く
void SortedLinkedList::remove(int n) {
}
}
```

```

int main() {
    int x[] = { 0, 1, 2, 5, 3, 3, 5, 7, 8, 8 };
    int n; char select;
    SortedLinkedList ichain(x, x + 10);
    cout << endl << "Menu[I:Insert, R:Remove, S:Size, P:Print, C:Clear, Q:Quit]";
    while ((cout << endl << "Select I/R/S/P/C/Q-->") && (cin >> select)) {
        switch (select) {
            case 'I': // 新しいノードを追加
            case 'i': cout << "Input a data-->"; cin >> n; ichain.insert(n); break;
            case 'R': // 指定したノードを削除
            case 'r': cout << "Input a data-->"; cin >> n; ichain.remove(n); break;
            case 'S': // リストの要素数を表示
            case 's': cout << "Length=" << ichain.size() << endl; break;
            case 'P': // リストの全データを表示. iteratorという抽象化されたポインタでアクセスする
            case 'p': ichain.printALL(); break;
                        cout << "END_OF_DATA" << endl; break;
            case 'C': // リストの全データを削除
            case 'c': ichain.clear(); break;
            case 'Q': // プログラムを終了
            case 'q': break;
            default: continue;
        }
        if ((select == 'Q') || (select == 'q')) break;
    }
    return 0;
}

```

## 実行例

```
$ g++ prac05.cpp -o prac05
$ ./prac05
Menu[I:Insert, R:Remove, S:Size, P:Print, C:Clear, Q:Quit]
Select I/R/S/P/C/Q-->P
0-->1-->2-->3-->3-->5-->5-->7-->8-->8-->END_OF_DATA
```

```
Select I/R/S/P/C/Q-->S
Length=10
```

```
Select I/R/S/P/C/Q-->R
Input a data-->3
3 is released
```

```
Select I/R/S/P/C/Q-->R
Input a data-->0
0 is released
```

```
Select I/R/S/P/C/Q-->r
Input a data-->6
```

```
Select I/R/S/P/C/Q-->s
Length=8
```

```
Select I/R/S/P/C/Q-->p
1-->2-->3-->5-->5-->7-->8-->8-->END_OF_DATA
```

```
Select I/R/S/P/C/Q-->i
Input a data-->6
```

```
Select I/R/S/P/C/Q-->P
1-->2-->3-->5-->5-->6-->7-->8-->8-->END_OF_DATA
```

```
Select I/R/S/P/C/Q--> r
Input a data-->8
8 is released
```

```
Select I/R/S/P/C/Q--> r
Input a data-->8
8 is released
```

```
Select I/R/S/P/C/Q--> P
1-->2-->3-->5-->5-->6-->7-->END_OF_DATA
```

```
Select I/R/S/P/C/Q-->c
1 is released
2 is released
3 is released
5 is released
5 is released
6 is released
7 is released
8 is released
8 is released
```

```
Select I/R/S/P/C/Q-->s
Length=0
```

```
Select I/R/S/P/C/Q-->q
```