

問題

入力された英単語(すべて小文字と考えてよい)を、アルファベット昇順にリスト構造で保持するプログラムを作成したい。但し、リストにある単語と同じ単語が入力された場合には、各ノードのデータメンバである単語の出現回数を増やすことで対応する。また、指定した出現回数の単語がすべて削除できるようにしたい。このため、以下の SortedLinkedList クラスを用いてプログラムを完成させなさい。

```
class SortedLinkedList{
private:
    class Node{
    public:
        string data;           // ノードの値
        int hindo;             // 頻度
        Node *next;            // 次のノードへのポインタ
        Node(string word="", int num=0, Node *n=NULL){ data=word; hindo=num; next=n; } // コンストラクタ
        ~Node(){ cout << data << " is released.¥n"; } // デストラクタ
    };
    Node *head;
    void print(Node *pos) const;
    Node* insertNode(Node *pos, string newword);
    Node* removeNode(Node *pos, int n);
    void clearNode(Node *pos);
public:
    SortedLinkedList(){ head=NULL; } // リストの先頭
    SortedLinkedList(string*, string*); // リストのpos以降のデータを出力
    ~SortedLinkedList(){ clear(); } // リストのpos以降にnewwordを昇順に追加
    void printAll() const { print(head); } // リストのpos以降の頻度nの単語を削除
    void insert(string newword){ head=insertNode(head, newword); } // リストのpos以降を削除
    void remove(int n){ head=removeNode(head, n); }
    void clear(){ clearNode(head); head=NULL; }
};
```

問題(続き)

mainは以下のプログラムを使用してください。

```
int main(int argc, char *argv[]){

    string a[]={ "apple", "apple", "banana", "peach", "banana", "peach", "banana", "peach", "melon", "melon", "lemon", "orange",
        "watermelon"};

    string newword;
    int n;
    char select;
    SortedLinkedList ichain(a, a+13);

    // メニューを表示して対応する処理を行う
    cout << "¥nMenu[I:Insert, R:Remove, P:Print, Q:Quit]";
    while( (cout << "¥n  Select I/R/S/P/C/Q-->" ) && (cin >> select) ){
        switch(select){
            case 'I':
                // リストへ新規ノードの追加
                case 'i': cout << "Input a data-->"; cin >> newword; ichain.insert(newword); break;
                // リストから指定ノードを削除
                case 'R':
                    // リストの全データを表示
                    case 'r': cout << "Remove a data-->"; cin >> n; ichain.remove(n); break;
                    // プログラムを終了
                case 'P': ichain.printAll(); break;
                case 'Q':
                case 'q': break;
                default: continue;
            }
            if( (select=='Q') || (select=='q') ){ break;}
        }

    return 0;
}
```

問題(続き)

<実行例>

comsv001% ./a.out

Menu[I:Insert, R:Remove, P:Print, Q:Quit]

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

apple(2)->banana(3)->lemon(1)->melon(2)->orange(1)->peach(3)->watermelon(1)->END_OF_DATA

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

Input a data-->apple

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

apple(3)->banana(3)->lemon(1)->melon(2)->orange(1)->peach(3)->watermelon(1)->END_OF_DATA

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

Remove a data-->3

apple is released.

banana is released.

peach is released.

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

lemon(1)->melon(2)->orange(1)->watermelon(1)->END_OF_DATA

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

watermelon is released.

orange is released.

melon is released.

lemon is released.

comsv001%