

課題 5：Queue クラスのメンバ関数 push(), front(), pop()を実装し, prac05_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 Node {
    string value; Node* nextp;
public:
    Node(string a, Node* p = nullptr) :value{ a }, nextp{ p } {}
    string getData() const { return value; }
    Node* getNext() const { return nextp; }
    void setNext(Node* pNext) { nextp = pNext; }
};

class Queue {
    Node* head;           // キューの先頭
    Node* tail;           // キューの末尾
public:
    Queue() { head = nullptr; tail = nullptr; } // コンストラクタ
    ~Queue() { while (!empty()) pop(); } // デストラクタ
    void push(string s); // 値sを持つノードをキューの末尾に追加
    void pop(); // キューの先頭のノードを削除する
    string front() const; // キューの先頭のノードの値を返す
    bool empty() const { return head == nullptr; } // キューが空かの判定
    size_t size() const; // キューの要素数を返す
};

// キューの要素数を返す
size_t Queue::size() const {
    size_t count{ 0 }; // 要素数をカウント
    auto node = head; // 最初はキュー先頭のノードを指すようにする
    while (node != nullptr) { // nodeがnullptrでないかぎり
        count++; // ノードをカウントして
        node = node->getNext(); // 次のノードをnodeが指すようにする
    }
    return count;
}

// 値xを持つノードをキューの末尾に追加する
void Queue::push(string s) {

}

// キューの先頭のノードの値を返す
string Queue::front() const {

}

// キューの先頭のノードを削除し, nextを新しい先頭にする
void Queue::pop() {

}
```

```

void print_queue(Queue& q) {
    size_t n{ q.size() };
    cout << "num: " << n << ", data: ";
    for (size_t i{ 0 }; i < n; i++) {
        string s = q.front();
        cout << " " << s;
        q.pop();
        q.push(s);
    }
    cout << "\n";
}

```

```

int main() {
    Queue q1;

    q1.push("Inoue");
    q1.push("Abe");
    q1.push("Kanda");
    q1.push("Satou");
    print_queue(q1);

    q1.pop();
    q1.pop();
    print_queue(q1);

    q1.pop();
    q1.pop();
    print_queue(q1);

    q1.push("Satou");
    q1.push("Abe");
    q1.push("Suzuki");
    q1.push("Kobayashi");
    print_queue(q1);

    q1.push("Okita");
    print_queue(q1);

    return 0;
}

```

実行例

```

$ g++ prac05.cpp -o prac05
num: 4, data: Inoue Abe Kanda Satou
num: 2, data: Kanda Satou
num: 0, data:
num: 4, data: Satou Abe Suzuki Kobayashi
num: 5, data: Satou Abe Suzuki Kobayashi Okita

```