SOURCE CODE PUSH QUEUE

```
#include <iostream>
#include <queue>

using namespace std;

int main() {

  // create a queue of string
  queue<string> animals;

  // push elements into the queue
  animals.push("Kucing");
  animals.push("Anjing");
  animals.push("Kuda");

  cout << "Queue: ";

  // print elements of queue
  // loop until queue is empty
  while(!animals.empty()) {

    // print the element
    cout << animals.front() << ", ";

    // pop element from the queue
    animals.pop();
  }

  cout << endl;

  return 0;
}
```

SOURCE CODE POP QUEUE

```
#include <iostream>
#include <queue>
using namespace std;

// function prototype for display_queue utility
void display_queue(queue<string> q);

int main() {

  // create a queue of string
  queue<string> animals;
  // push element into the queue
  animals.push("Kucing");
  animals.push("Anjing");
  animals.push("Kuda");
  animals.push("Gajah");

  cout << "Initial Queue: ";
  display_queue(animals);

  // remove element from queue
  animals.pop();

  cout << "Final Queue: ";
  display_queue(animals);

  return 0;
}

// utility function to display queue
void display_queue(queue<string> q) {
  while(!q.empty()) {
    cout << q.front() << ", ";
    q.pop();
  }

  cout << endl;
}
```

SOURCE CODE ACCESS QUEUE (FRONT & BACK)

```
#include <iostream>
#include <queue>
using namespace std;

int main() {

  // create a queue of int
  queue<int> nums;

  // push element into the queue
  nums.push(1);
  nums.push(2);
  nums.push(3);
  nums.push(4);
  nums.push(5);
  // get the element at the front
  int front = nums.front();
  cout << "First element: " << front << endl;

  // get the element at the back
  int back = nums.back();
  cout << "Last element: " << back << endl;

  return 0;
}
```

SOURCE CODE GET SIZE PADA QUEUE

```
#include <iostream>
#include <queue>
using namespace std;

int main() {

  // create a queue of string
  queue<string> makanan;

  // push element into the queue
  makanan.push("Nasi Kebuli");
  makanan.push("Nasi Kuning");
  makanan.push("Nasi Campur");
  makanan.push("Nasi Padang");

  // get the size of the queue
  int size = makanan.size();
  cout << "Size of the queue: " << size;

  return 0;
}
```

SOURCE CODE CEK EMPTY QUEUE

```
#include <iostream>
#include <queue>
using namespace std;

int main() {

  // create a queue of string
  queue<string> minuman;

  cout << "Apakah queue ini kosong?";
  // check if the queue is empty
  if (minuman.empty()) {
    cout << "Yes" << endl;
  }
  else {
    cout << "No" << endl;
  }

  cout << "Pushing elements..." << endl;

  // push element into the queue
  minuman.push("Es Jeruk");
  minuman.push("Es Teh");

  cout << "Apakah queue ini kosong?";

  // check if the queue is empty
  if (minuman.empty()) {
    cout << "Yes";
  }
  else {
    cout << "No";
  }

  return 0;
}
```

SOURCE CODE INSERT PRIORITY QUEUE

```
#include<iostream>
#include <queue>
using namespace std;

int main() {

  // create a queue of int
  priority_queue<int> numbers;

  // add items to priority_queue
  numbers.push(1);
  numbers.push(20);
  numbers.push(7);
  numbers.push(100);

  cout << "Priority Queue: ";
  // display all elements of numbers
  while(!numbers.empty()) {
    cout << numbers.top() << ", ";
    numbers.pop();
  }

  cout << endl;

  return 0;
}
```

SOURCE CODE DELETE PRIORITY QUEUE

```
#include<iostream>
#include <queue>
using namespace std;

// function prototype for display_priority_queue()
void display_priority_queue(priority_queue<int> pq);

int main() {

  // create a queue of int
  priority_queue<int> numbers;

  // add items to priority_queue
  numbers.push(1);
  numbers.push(20);
  numbers.push(7);
  numbers.push(100);

  cout << "Initial Priority Queue: ";
  display_priority_queue(numbers);

  // remove element from queue
  numbers.pop();

  cout << "Final Priority Queue: ";
  display_priority_queue(numbers);

  return 0;
}

// utility function to dislay priority queue
void display_priority_queue(priority_queue<int> pq) {
  while(!pq.empty()) {
    cout << pq.top() << ", ";
    pq.pop();
  }

  cout << endl;
}
```

SOURCE CODE ACCESS ELEMENT PADA PRIORITY QUEUE

```
#include<iostream>
#include <queue>
using namespace std;

int main() {

  // create a priority queue of int
  priority_queue<int> numbers;

  // add items to priority_queue
  numbers.push(1);
  numbers.push(20);
  numbers.push(7);

  // get the element at the top
  int top = numbers.top();
  cout << "Top element: " << top;

  return 0;
}
```

SOURCE CODE DEQUE SEDERHANA

```
#include <iostream>
#include <deque>

int main() {
    // Deklarasi deque
    std::deque<int> dq;

    // Menambahkan elemen ke belakang deque
    dq.push_back(10);
    dq.push_back(20);
    dq.push_back(30);

    // Menambahkan elemen ke depan deque
    dq.push_front(5);
    dq.push_front(2);
    dq.push_front(100);
    // Menampilkan elemen deque
    std::cout << "Isi deque: ";
    for(int i : dq) {
        std::cout << i << " ";
    }
    std::cout << std::endl;

    // Mengakses elemen depan dan belakang
    std::cout << "Elemen depan: " << dq.front() << std::endl;
    std::cout << "Elemen belakang: " << dq.back() << std::endl;

    // Menghapus elemen depan dan belakang
    dq.pop_front();
    dq.pop_back();

    // Menampilkan elemen deque setelah penghapusan
    std::cout << "Isi deque setelah pop: ";
    for(int i : dq) {
        std::cout << i << " ";
    }
    std::cout << std::endl;

    return 0;
}
```