```cpp
struct Node
{
  int data;
  Node *pNext;
};

Node* initNode(int value)
{
  Node *p = new Node;

  if (p == NULL)
  {
    cout << "Khong du bo nho de cap phat" << endl;
    return NULL;
  }

  p->data = value;
  p->pNext = NULL;

  return p;
}

struct Queue
{
  Node* pFront;
  Node* pRear;
};

void initQueue(Queue &q)
{
  q.pFront = NULL;
  q.pRear = NULL;
};

bool isEmpty(Queue q)
{
  return q.pFront == NULL;
};

/* 3. Viết hàm thêm phần tử vào Queue
*/
void enQueue(Queue &q, Node *p)
{
  if (q.pFront == NULL)
  {
    q.pFront = p;
    q.pRear = p;
  }
  else
  {
    q.pRear->pNext = p;
    q.pRear = p;
  }
};

```

```cpp
/* 4. Viết hàm lấy phần tử ra khỏi Queue
*/
bool deQueue(Queue &q)
{
  if (q.pFront == NULL)
  {
    return false;
  }

  Node *p = q.pFront;
  q.pFront = q.pFront->pNext;
  delete p;

  if (q.pFront == NULL)
  {
    q.pRear = NULL;
  }

  return true;
};

void printfQueue(Queue q)
{
  if (q.pFront == NULL)
  {
    cout << "Hang doi rong" << endl;
    return;
  }

  cout << "<<Front<< ";
  for (Node *p = q.pFront; p != NULL; p = p->pNext)
  {
    cout << p->data << " ";
  }
  cout << "<<Rear<<" << endl;
}
```