Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Людинно-машинна взаємодія*

Лабораторна робота № 7

Тема: Робота з контейнерами в середовищі Qt Creator: QList, QLinkedList

*Варіант 2*

Виконав: ***Гук Д.П.***

Група ІПЗ-31

Дата:12 листопада 2023 р.

Викладач: Пікуляк М.В.

Івано-Франківськ – 2023

**Мета роботи:**

Отримати навички роботи з контейнерами QList, QLinkedList в ICP “Qt-Creator”.

**Завдання для виконання :**

1. Заповнити список випадковими елементами і реалізувати

видалення елементів з позицій з N по K.

2. Заповнити зв’язний список випадковими елементами і

впорядкувати їх за спаданням.

3. Створити консольний проект для роботи з двохзв’язним

списком (оголосіть екземпляр класу QLinkedList <int> list). Заповніть його

випадковими значеннями і продемонструйте в ньому роботу таких алгоритмів STL:

а) count (first, last, value) – повертає значення, яке показує скільки разів

елемент зі значенням value входить в послідовність, задану ітераторами;

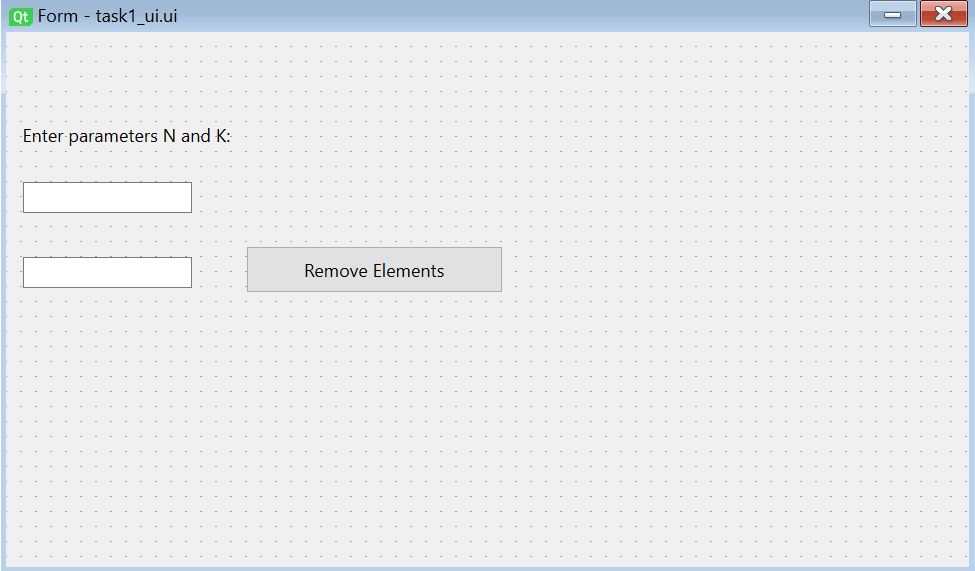
б) reverse (first, last) – переставляє елементи в зворотному порядку;

в) iter\_swap (first, last) – міняє місцями значення елементів, на які вказують

ітератори.

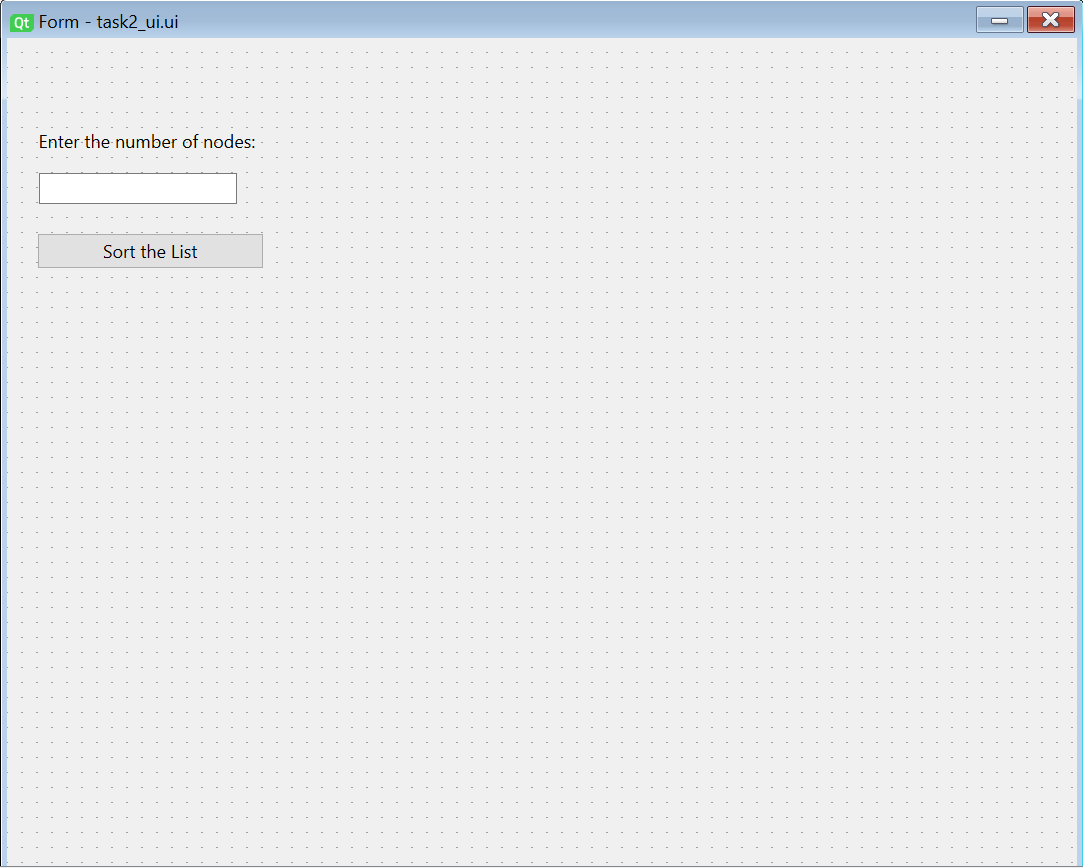
**Тексти скриптів і зображення діалогових вікон QtCreator з виконаними завданнями :**

**Завдання №1:**



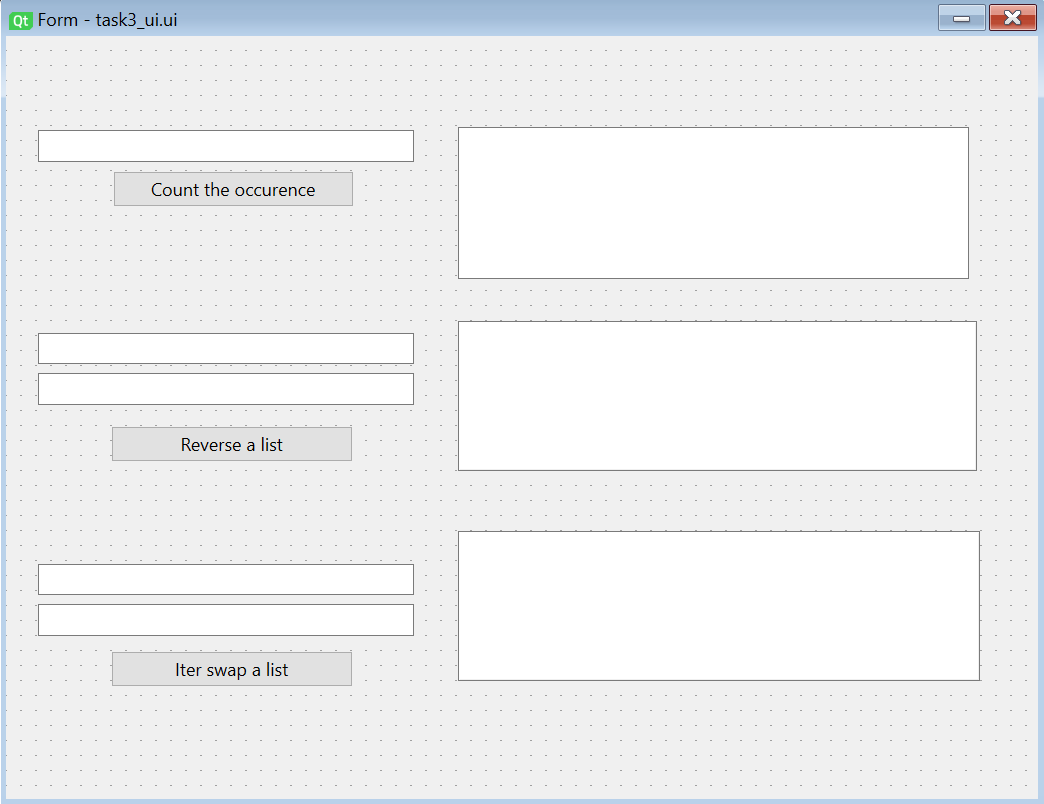
import random  
import sys  
  
from PyQt6 import QtCore, QtWidgets  
  
  
class Ui\_Form(object):  
 def setupUi(self, Form):  
 Form.setObjectName("Form")  
 Form.resize(642, 357)  
 self.label = QtWidgets.QLabel(parent=Form)  
 self.label.setGeometry(QtCore.QRect(11, 61, 151, 16))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Form)  
 self.label\_2.setGeometry(QtCore.QRect(10, 270, 551, 61))  
 self.label\_2.setText("")  
 self.label\_2.setObjectName("label\_2")  
 self.lineEdit = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit.setGeometry(QtCore.QRect(11, 100, 113, 21))  
 self.lineEdit.setObjectName("lineEdit")  
 self.lineEdit\_2 = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit\_2.setGeometry(QtCore.QRect(11, 150, 113, 21))  
 self.lineEdit\_2.setObjectName("lineEdit\_2")  
 self.pushButton = QtWidgets.QPushButton(parent=Form)  
 self.pushButton.setGeometry(QtCore.QRect(160, 143, 171, 31))  
 self.pushButton.setObjectName("pushButton")  
 self.pushButton.clicked.connect(self.remove\_elements)  
 self.label\_3 = QtWidgets.QLabel(parent=Form)  
 self.label\_3.setGeometry(QtCore.QRect(10, 200, 551, 61))  
 self.label\_3.setText("")  
 self.label\_3.setObjectName("label\_3")  
  
 self.retranslateUi(Form)  
 QtCore.QMetaObject.connectSlotsByName(Form)  
  
 def retranslateUi(self, Form):  
 \_translate = QtCore.QCoreApplication.translate  
 Form.setWindowTitle(\_translate("Form", "Task1"))  
 self.label.setText(\_translate("Form", "Enter parameters N and K: "))  
 self.pushButton.setText(\_translate("Form", "Remove Elements"))  
  
 def remove\_elements(self):  
 try:  
 N = int(self.lineEdit.text())  
 K = int(self.lineEdit\_2.text())  
  
 if N >= K:  
 self.label\_2.setText("N must be less than K")  
 return  
  
 random\_list = [random.randint(1, 100) for \_ in range(20)]  
 self.label\_3.setText("Random List: " + ', '.join(map(str, random\_list)))  
  
 if K > len(random\_list):  
 K = len(random\_list)  
  
 del random\_list[N - 1:K]  
  
 self.label\_2.setText("Updated List: " + ', '.join(map(str, random\_list)))  
  
 except ValueError:  
 self.label\_2.setText("Please enter valid integers for N and K")  
  
app = QtWidgets.QApplication(sys.argv)  
task3\_dialog = QtWidgets.QDialog()  
ui = Ui\_Form()  
ui.setupUi(task3\_dialog)  
task3\_dialog.show()  
sys.exit(app.exec())

**Завдання №2:**

****

import random  
import sys  
from collections.abc import Iterable  
  
from PyQt6 import QtCore, QtWidgets  
  
class Node:  
 def \_\_init\_\_(self, data=None):  
 self.data = data  
 self.next = None  
  
class LinkedList(Iterable):  
 def \_\_init\_\_(self):  
 self.head = None  
  
 def \_\_iter\_\_(self):  
 current = self.head  
 while current:  
 yield current.data  
 current = current.next  
  
 def append(self, data):  
 new\_node = Node(data)  
 if not self.head:  
 self.head = new\_node  
 return  
 current = self.head  
 while current.next:  
 current = current.next  
 current.next = new\_node  
  
 def sort\_descending(self):  
 if not self.head or not self.head.next:  
 return  
  
 def merge\_sort(node):  
 if not node or not node.next:  
 return node  
  
 middle = get\_middle(node)  
 next\_to\_middle = middle.next  
 middle.next = None  
  
 left = merge\_sort(node)  
 right = merge\_sort(next\_to\_middle)  
  
 return merge(left, right)  
  
 def merge(left, right):  
 result = None  
  
 if not left:  
 return right  
 if not right:  
 return left  
  
 if left.data >= right.data:  
 result = left  
 result.next = merge(left.next, right)  
 else:  
 result = right  
 result.next = merge(left, right.next)  
  
 return result  
  
 def get\_middle(node):  
 if not node:  
 return node  
  
 slow = node  
 fast = node  
  
 while fast.next and fast.next.next:  
 slow = slow.next  
 fast = fast.next.next  
  
 return slow  
  
 self.head = merge\_sort(self.head)  
  
  
class Ui\_Form(object):  
 def setupUi(self, Form):  
 Form.setObjectName("Form")  
 Form.resize(713, 571)  
 self.label = QtWidgets.QLabel(parent=Form)  
 self.label.setGeometry(QtCore.QRect(21, 61, 171, 16))  
 self.label.setObjectName("label")  
 self.lineEdit = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit.setGeometry(QtCore.QRect(21, 90, 132, 21))  
 self.lineEdit.setObjectName("lineEdit")  
 self.pushButton = QtWidgets.QPushButton(parent=Form)  
 self.pushButton.setGeometry(QtCore.QRect(20, 130, 151, 24))  
 self.pushButton.setObjectName("pushButton")  
 self.pushButton.clicked.connect(self.sort\_linked\_list)  
 self.label\_2 = QtWidgets.QLabel(parent=Form)  
 self.label\_2.setGeometry(QtCore.QRect(21, 200, 301, 321))  
 self.label\_2.setText("")  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Form)  
 self.label\_3.setGeometry(QtCore.QRect(350, 200, 331, 321))  
 self.label\_3.setText("")  
 self.label\_3.setObjectName("label\_3")  
  
 self.retranslateUi(Form)  
 QtCore.QMetaObject.connectSlotsByName(Form)  
  
 def retranslateUi(self, Form):  
 \_translate = QtCore.QCoreApplication.translate  
 Form.setWindowTitle(\_translate("Form", "Task 2"))  
 self.label.setText(\_translate("Form", "Enter the number of nodes: "))  
 self.pushButton.setText(\_translate("Form", "Sort the List"))  
  
  
 def sort\_linked\_list(self):  
 try:  
 num\_nodes = int(self.lineEdit.text())  
  
 linked\_list = LinkedList()  
 for \_ in range(num\_nodes):  
 linked\_list.append(random.randint(1, 100))  
  
 original\_list = "Original List:\n" + ', '.join(  
 str(node) + ('\n' if i % 10 == 9 else '') for i, node in enumerate(linked\_list))  
 linked\_list.sort\_descending()  
 sorted\_list = "Sorted List (descending):\n" + ', '.join(  
 str(node) + ('\n' if i % 10 == 9 else '') for i, node in enumerate(linked\_list))  
  
 self.label\_2.setText(original\_list)  
 self.label\_3.setText(sorted\_list)  
  
 except ValueError:  
 self.label\_2.setText("Please enter a valid integer for the number of nodes")  
 self.label\_3.setText("Please enter a valid integer for the number of nodes")  
  
  
app = QtWidgets.QApplication(sys.argv)  
task2 = QtWidgets.QDialog()  
ui = Ui\_Form()  
ui.setupUi(task2)  
task2.show()  
sys.exit(app.exec())

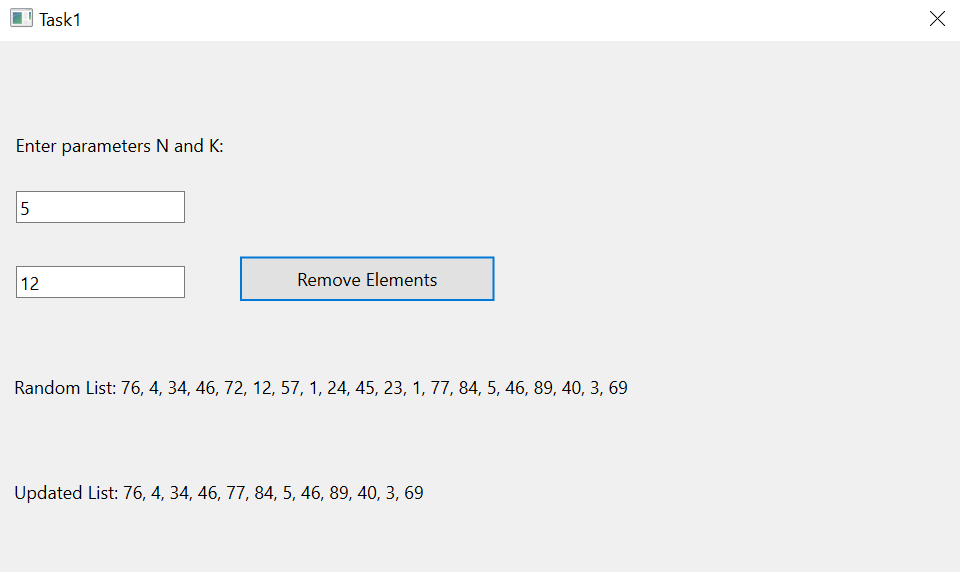
**Завдання №3:**

****

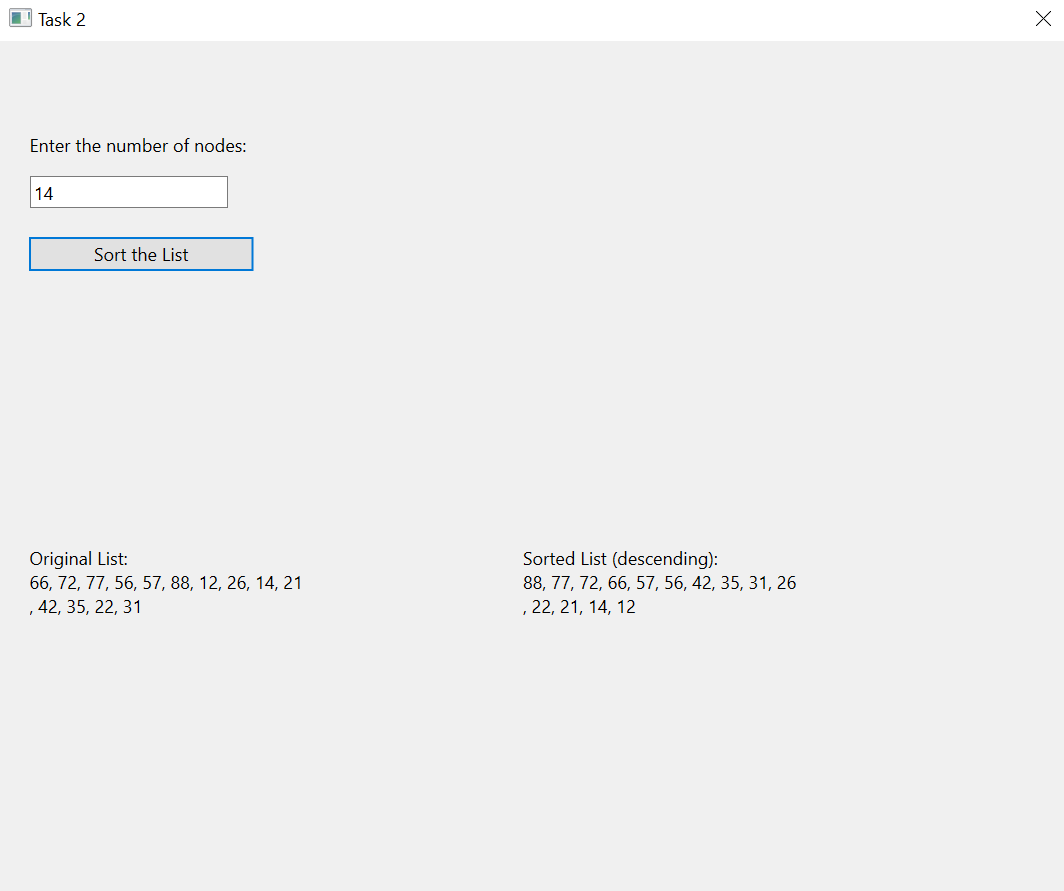
import random  
  
from PyQt6 import QtCore, QtWidgets  
  
class Node:  
 def \_\_init\_\_(self, data):  
 self.data = data  
 self.prev = None  
 self.next = None  
  
class DoublyLinkedList:  
 def \_\_init\_\_(self):  
 self.head = None  
 self.tail = None  
  
 def append(self, data):  
 new\_node = Node(data)  
 if not self.head:  
 self.head = new\_node  
 self.tail = new\_node  
 else:  
 new\_node.prev = self.tail  
 self.tail.next = new\_node  
 self.tail = new\_node  
  
 def count(self, value):  
 count = 0  
 current = self.head  
 while current:  
 if current.data == value:  
 count += 1  
 current = current.next  
 return count  
  
 def reverse(self, start\_idx, end\_idx, callback=None):  
 if start\_idx < 0 or end\_idx < 0 or start\_idx >= end\_idx:  
 return  
  
 current = self.head  
 count = 0  
  
 while count < start\_idx:  
 current = current.next  
 count += 1  
  
 start\_node = current  
  
 while count < end\_idx:  
 current = current.next  
 count += 1  
  
 end\_node = current  
  
 while start\_node != end\_node and end\_node.next != start\_node:  
 start\_node.data, end\_node.data = end\_node.data, start\_node.data  
 start\_node = start\_node.next  
 end\_node = end\_node.prev  
  
 if callback:  
 callback()  
 QtCore.QCoreApplication.processEvents()  
  
 def iter\_swap(self, first, last, callback=None):  
 first.data, last.data = last.data, first.data  
  
 if callback:  
 callback()  
 QtCore.QCoreApplication.processEvents()  
  
class Ui\_Form(object):  
 def setupUi(self, Form):  
 Form.setObjectName("Form")  
 Form.resize(688, 509)  
 self.lineEdit = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit.setGeometry(QtCore.QRect(21, 63, 251, 21))  
 self.lineEdit.setObjectName("lineEdit")  
 self.lineEdit\_2 = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit\_2.setGeometry(QtCore.QRect(21, 198, 251, 21))  
 self.lineEdit\_2.setObjectName("lineEdit\_2")  
 self.lineEdit\_3 = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit\_3.setGeometry(QtCore.QRect(21, 225, 251, 21))  
 self.lineEdit\_3.setObjectName("lineEdit\_3")  
 self.lineEdit\_4 = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit\_4.setGeometry(QtCore.QRect(21, 352, 251, 21))  
 self.lineEdit\_4.setObjectName("lineEdit\_4")  
 self.lineEdit\_5 = QtWidgets.QLineEdit(parent=Form)  
 self.lineEdit\_5.setGeometry(QtCore.QRect(21, 379, 251, 21))  
 self.lineEdit\_5.setObjectName("lineEdit\_5")  
 self.pushButton\_3 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_3.setGeometry(QtCore.QRect(71, 90, 161, 24))  
 self.pushButton\_3.setObjectName("pushButton\_3")  
 self.pushButton\_4 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_4.setGeometry(QtCore.QRect(70, 260, 161, 24))  
 self.pushButton\_4.setObjectName("pushButton\_4")  
 self.pushButton\_5 = QtWidgets.QPushButton(parent=Form)  
 self.pushButton\_5.setGeometry(QtCore.QRect(70, 410, 161, 24))  
 self.pushButton\_5.setObjectName("pushButton\_5")  
 self.textEdit = QtWidgets.QTextEdit(parent=Form)  
 self.textEdit.setGeometry(QtCore.QRect(301, 61, 341, 101))  
 self.textEdit.setObjectName("textEdit")  
 self.textEdit\_2 = QtWidgets.QTextEdit(parent=Form)  
 self.textEdit\_2.setGeometry(QtCore.QRect(301, 190, 346, 100))  
 self.textEdit\_2.setObjectName("textEdit\_2")  
 self.textEdit\_3 = QtWidgets.QTextEdit(parent=Form)  
 self.textEdit\_3.setGeometry(QtCore.QRect(301, 330, 348, 100))  
 self.textEdit\_3.setObjectName("textEdit\_3")  
  
 self.retranslateUi(Form)  
 QtCore.QMetaObject.connectSlotsByName(Form)  
  
  
 self.linked\_list = DoublyLinkedList()  
  
 values = [random.randint(1, 100) for \_ in range(10)]  
 for value in values:  
 self.linked\_list.append(value)  
  
 self.pushButton\_3.clicked.connect(self.count\_elements)  
 self.pushButton\_4.clicked.connect(self.reverse\_elements)  
 self.pushButton\_5.clicked.connect(self.iter\_swap\_elements)  
  
 def retranslateUi(self, Form):  
 \_translate = QtCore.QCoreApplication.translate  
 Form.setWindowTitle(\_translate("Form", "Form"))  
 self.pushButton\_3.setText(\_translate("Form", "Count the occurrence"))  
 self.pushButton\_4.setText(\_translate("Form", "Reverse a list"))  
 self.pushButton\_5.setText(\_translate("Form", "Iter swap a list"))  
  
 def count\_elements(self):  
 value\_to\_count = int(self.lineEdit.text())  
 count = self.linked\_list.count(value\_to\_count)  
 self.textEdit.clear()  
 self.textEdit.append(f"The value {value\_to\_count} appears {count} times in the linked list.")  
  
 def reverse\_elements(self):  
 start\_idx = int(self.lineEdit\_2.text())  
 end\_idx = int(self.lineEdit\_3.text())  
  
 def callback():  
 self.textEdit\_2.clear()  
 self.textEdit\_2.append(f"Elements reversed in the specified range (indices {start\_idx}-{end\_idx}) in the linked list.")  
 self.display\_linked\_list(self.textEdit\_2)  
  
 self.linked\_list.reverse(start\_idx, end\_idx, callback)  
  
 def iter\_swap\_elements(self):  
 index\_1 = int(self.lineEdit\_4.text())  
 index\_2 = int(self.lineEdit\_5.text())  
  
 def callback():  
 self.textEdit\_3.clear()  
 self.textEdit\_3.append(f"Elements at indices {index\_1} and {index\_2} swapped in the linked list.")  
 self.display\_linked\_list(self.textEdit\_3)  
  
 self.linked\_list.iter\_swap(self.linked\_list.head, self.linked\_list.tail, callback)  
  
 def display\_linked\_list(self, output\_field):  
 current = self.linked\_list.head  
 elements = []  
  
 while current:  
 elements.append(current.data)  
 current = current.next  
  
 output\_field.append("Resulting Linked List: " + ", ".join(map(str, elements)))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
 Form = QtWidgets.QWidget()  
 ui = Ui\_Form()  
 ui.setupUi(Form)  
 Form.show()  
 sys.exit(app.exec())

**Cкрін-шоти виконання завдань лабораторної роботи :**

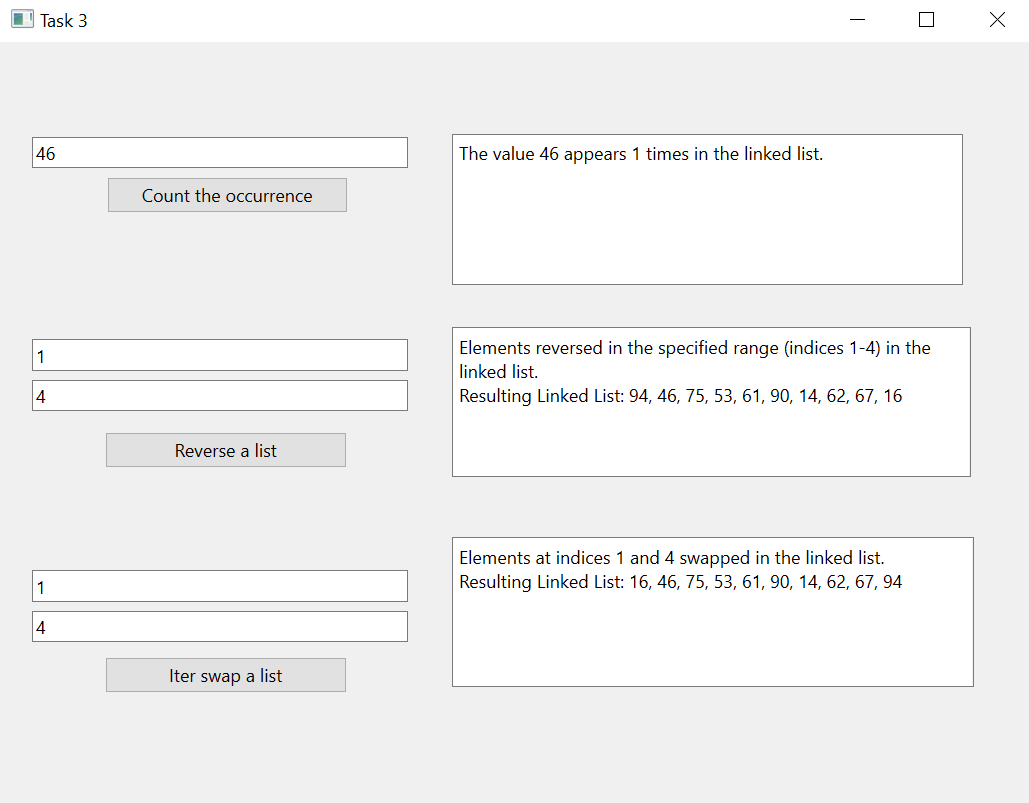
**Завдання №1 :**

****

**Завдання №2 :**

****

**Завдання №3 :**

****