Міністерство освіти і науки України
Прикарпатський національний університет
імені В.Стефаника

Факультет математики та інформатики
Кафедра інформаційних технологій

Людинно-машинна взаємодія

Лабораторна робота № 6

Тема: Контейнерний клас QVector. Робота з двовимірними масивами

Варіант 2

Виконав: ***Гук Д.П.***
Група ІПЗ-31
Дата:20 жовтня 2023 р.
Викладач: Пікуляк М.В.

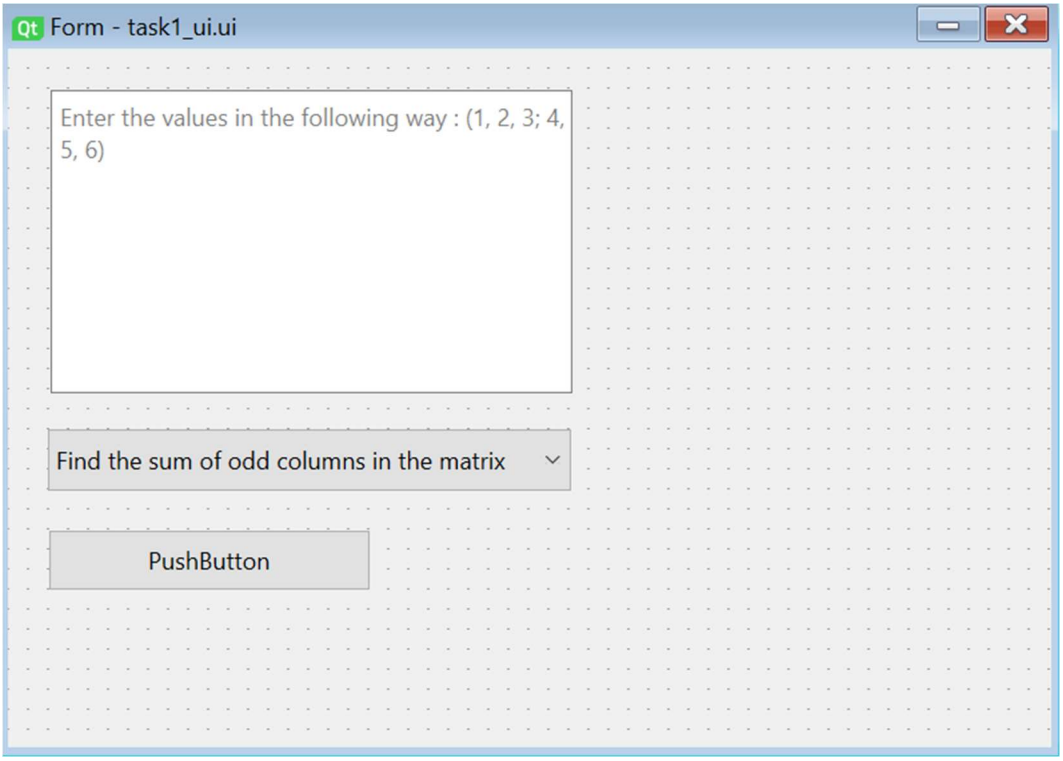
Івано-Франківськ – 2023

Мета роботи:
Отримати навички роботи з двовимірними масивами в ICP “Qt-Creator”.

Завдання для виконання :
1. Дана матриця розміру m x n. Знайти суми елементів всіх її парних і непарних стовпців (знаходження суми парних / непарних стовпців визначається користувачем у віджеті comboBox).
2. Дана матриця розміру m x n. Перетворити матрицю, помінявши місцями мінімальний і максимальний елемент в кожному рядку.

Тексти скриптів і зображення діалогових вікон QtCreator з виконаними завданнями :

Завдання №1:



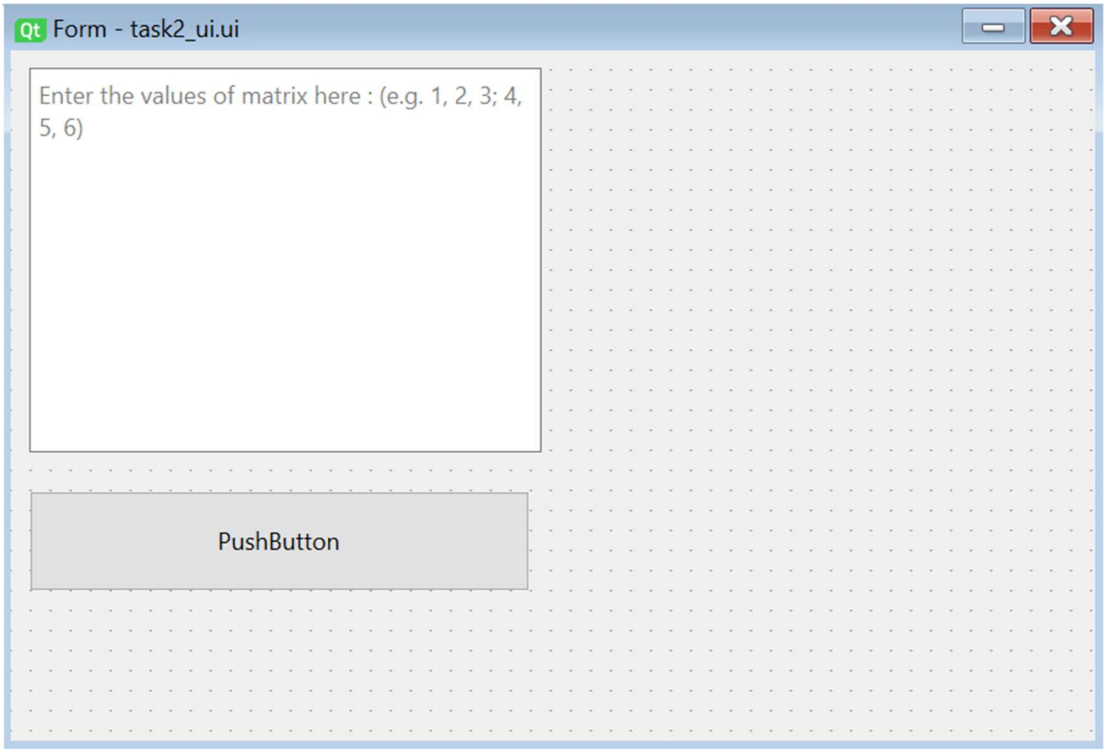

```
from PyQt6 import QtCore, QtGui, QtWidgets
import sys

class Ui_Form(object):
```

```
    def setupUi(self, Form):
        Form.setObjectName("Task 1")
        Form.resize(521, 349)
        self.matrix_input =
QtWidgets.QPlainTextEdit(parent=Form)
        self.matrix_input.setGeometry(QtCore.QRect(21,
21, 261, 151))

self.matrix_input.setObjectName("plainTextEdit")
        self.action_selection =
QtWidgets.QComboBox(parent=Form)

self.action_selection.setGeometry(QtCore.QRect(20, 190,
261, 31))
        self.action_selection.setObjectName("comboBox")
        self.action_selection.addItem("")
        self.action_selection.addItem("")
        self.pushButton =
QtWidgets.QPushButton(parent=Form)
        self.pushButton.setGeometry(QtCore.QRect(20,
240, 161, 31))
        self.pushButton.setObjectName("pushButton")
        self.result_label =
QtWidgets.QLabel(parent=Form)
        self.result_label.setGeometry(QtCore.QRect(20,
300, 251, 31))
        self.result_label.setText("")
        self.result_label.setObjectName("label")

        self.retranslateUi(Form)
        QtCore.QMetaObject.connectSlotsByName(Form)

        self.pushButton.clicked.connect(self.calculate)

    def retranslateUi(self, Form):
```

```
        _translate = QtCore.QCoreApplication.translate
        Form.setWindowTitle(_translate("Form", "Task
1"))

self.matrix_input.setPlaceholderText(_translate("Form",
"Enter the values in the following way : \n(\ne.g.\n1,
2, 3; \n4, 5, 6\n)"))
        self.action_selection.setItemText(0,
_translate("Form", "Find the sum of even columns in the
matrix"))
        self.action_selection.setItemText(1,
_translate("Form", "Find the sum of odd columns in the
matrix"))
        self.pushButton.setText(_translate("Form",
"Calculate the result"))

    def calculate(self):
        try:
            matrix_str =
self.matrix_input.toPlainText()
            matrix = [[int(num) for num in
row.split(',')] for row in matrix_str.split(';')]

            if len(matrix) < 1 or any(len(row) !=
len(matrix[0]) for row in matrix):
                self.result_label.setText("Invalid
matrix format.")
                return

            if self.action_selection.currentIndex() ==
0:
                even_sum =
self.calculate_even_columns(matrix)
                self.result_label.setText(f"Sum of even
columns: {even_sum}")
```

```
            elif self.action_selection.currentIndex()
== 1:
                odd_sum =
self.calculate_odd_columns(matrix)
                self.result_label.setText(f"Sum of odd
columns: {odd_sum}")

        except Exception as e:
            self.result_label.setText(f"Error: {e}")

    def calculate_even_columns(self, matrix):
        even_column_sum = 0
        for j in range(len(matrix[0])):
            if j % 2 == 0:
                for i in range(len(matrix)):
                    even_column_sum += matrix[i][j]
        return even_column_sum

    def calculate_odd_columns(self, matrix):
        odd_column_sum = 0
        for j in range(len(matrix[0])):
            if j % 2 != 0:
                for i in range(len(matrix)):
                    odd_column_sum += matrix[i][j]
        return odd_column_sum


if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    task1_dialog = QtWidgets.QDialog()
    ui = Ui_Form()
    ui.setupUi(task1_dialog)
    task1_dialog.show()
    sys.exit(app.exec())
```

Завдання №2:




```
from PyQt6 import QtCore, QtGui, QtWidgets
import sys

class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Task 2")
        Form.resize(542, 345)
        self.matrix_input =
QtWidgets.QTextEdit(parent=Form)
        self.matrix_input.setGeometry(QtCore.QRect(9,
9, 256, 192))
        self.matrix_input.setObjectName("textEdit")
        self.transform_button =
QtWidgets.QPushButton(parent=Form)

self.transform_button.setGeometry(QtCore.QRect(9, 220,
250, 50))
```

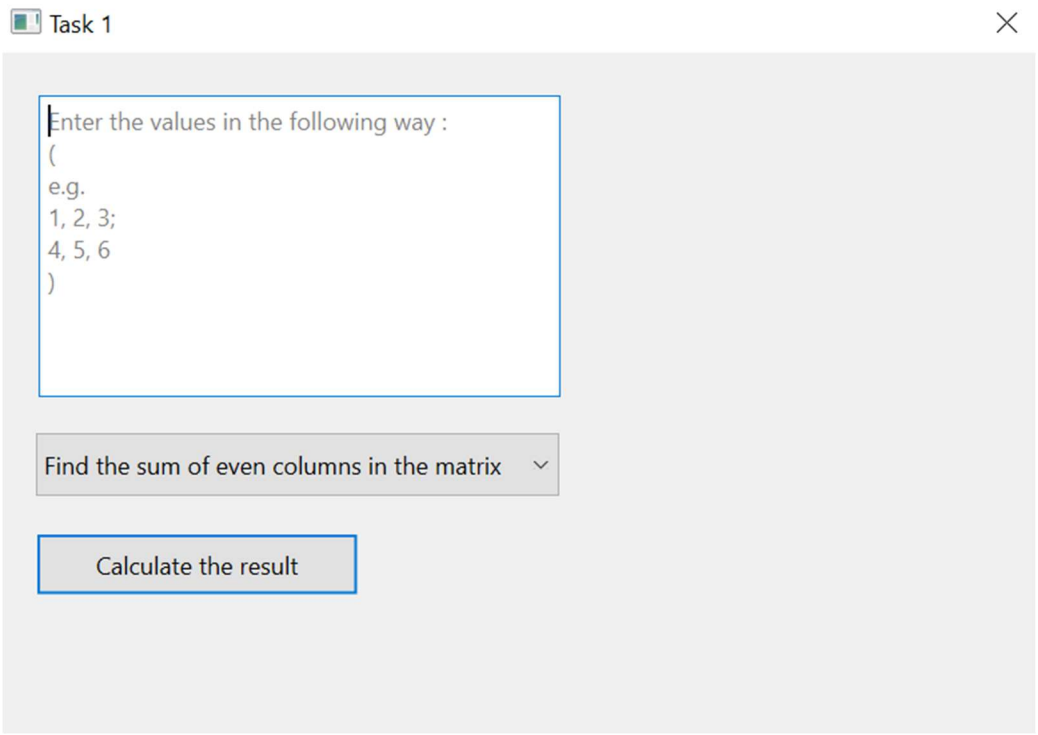
```
self.transform_button.setObjectName("pushButton")
        self.result_label =
QtWidgets.QLabel(parent=Form)
        self.result_label.setGeometry(QtCore.QRect(9,
280, 250, 60))
        self.result_label.setText("")
        self.result_label.setObjectName("label")


self.transform_button.clicked.connect(self.transform)
# Connect button click to transform function

        self.retranslateUi(Form)
        QtCore.QMetaObject.connectSlotsByName(Form)

    def retranslateUi(self, Form):
        _translate = QtCore.QCoreApplication.translate
        Form.setWindowTitle(_translate("Form", "Task
2"))

self.matrix_input.setPlaceholderText(_translate("Form",
"Enter the values in the following way : \n(\ne.g.\n1,
2, 3; \n4, 5, 6\n)"))

self.transform_button.setText(_translate("Form",
"Transform the matrix"))

    def transform(self):
        try:
            matrix_str =
self.matrix_input.toPlainText()
            matrix = [[int(num) for num in
row.split(',')] for row in matrix_str.split(';')]
```

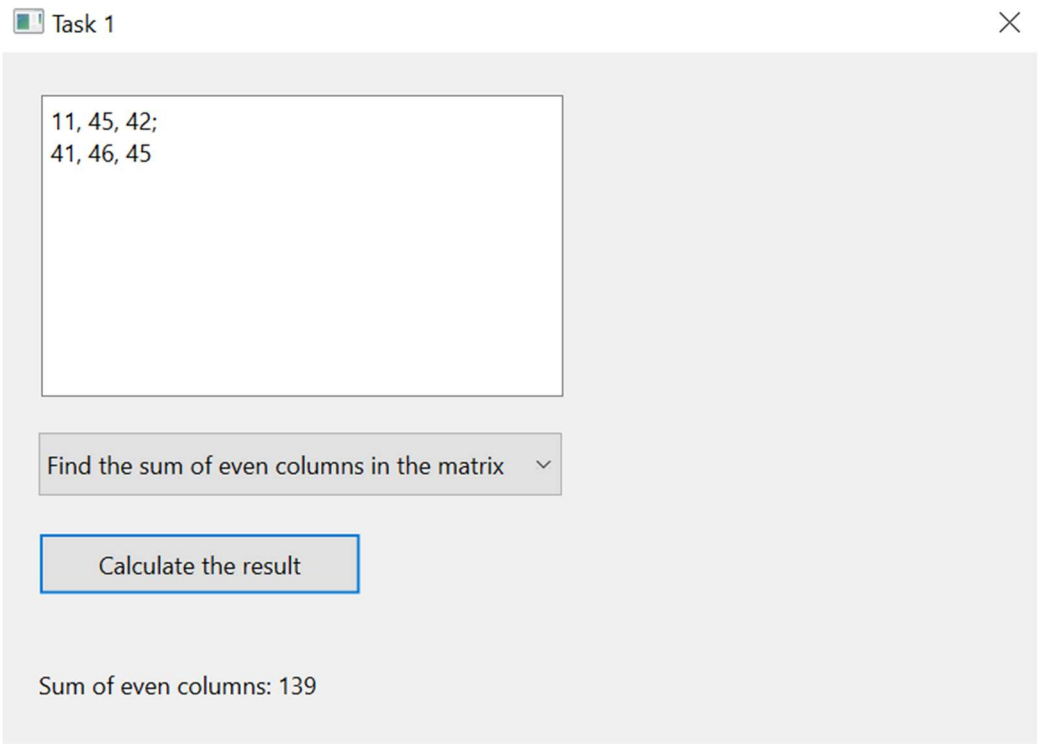
```
            if len(matrix) < 1 or any(len(row) !=
len(matrix[0]) for row in matrix):
                self.result_label.setText("Invalid
matrix format.")
                return

            transformed_matrix =
self.transform_matrix(matrix)
            self.result_label.setText("Transformed
Matrix:")
            for row in transformed_matrix:

self.result_label.setText(self.result_label.text() +
f"\n{', '.join(map(str, row))}")

        except Exception as e:
            self.result_label.setText(f"Error: {e}")

    def swap_min_max(self, row):
        min_val = min(row)
        max_val = max(row)
        min_idx = row.index(min_val)
        max_idx = row.index(max_val)
        row[min_idx], row[max_idx] = max_val, min_val
        return row

    def transform_matrix(self, matrix):
        transformed_matrix = []
        for row in matrix:
            transformed_row = self.swap_min_max(row)
            transformed_matrix.append(transformed_row)
        return transformed_matrix


if __name__ == "__main__":
```

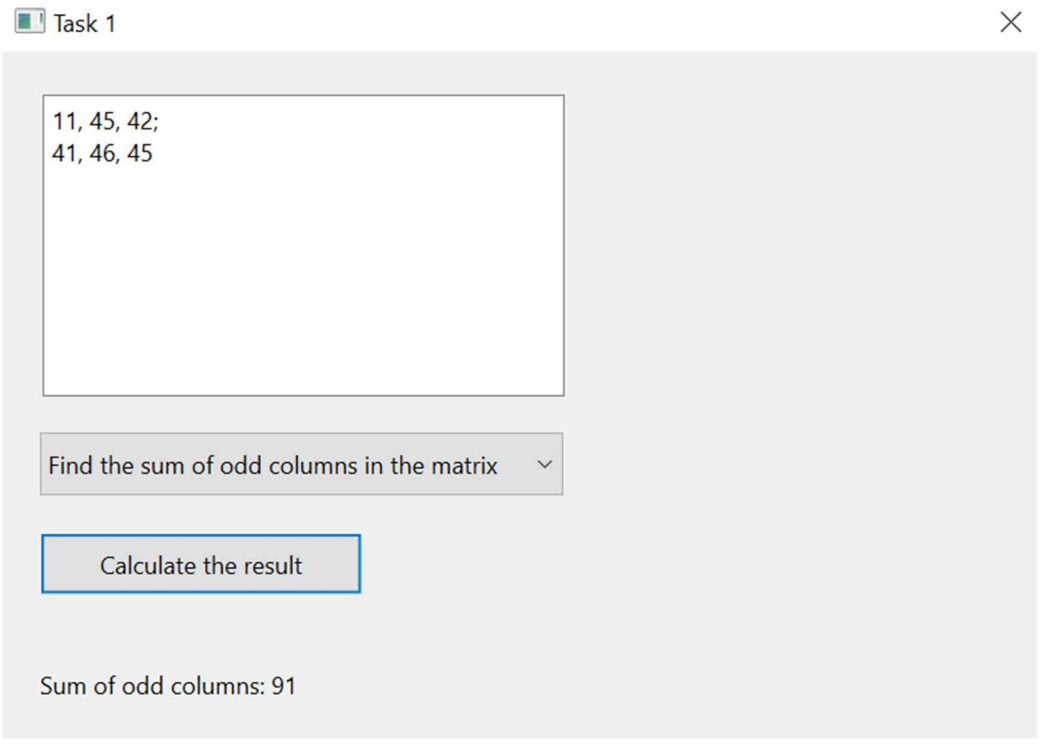
```
    app = QtWidgets.QApplication(sys.argv)
    task2_dialog = QtWidgets.QDialog()
    ui = Ui_Form()
    ui.setupUi(task2_dialog)
    task2_dialog.show()
    sys.exit(app.exec())
```

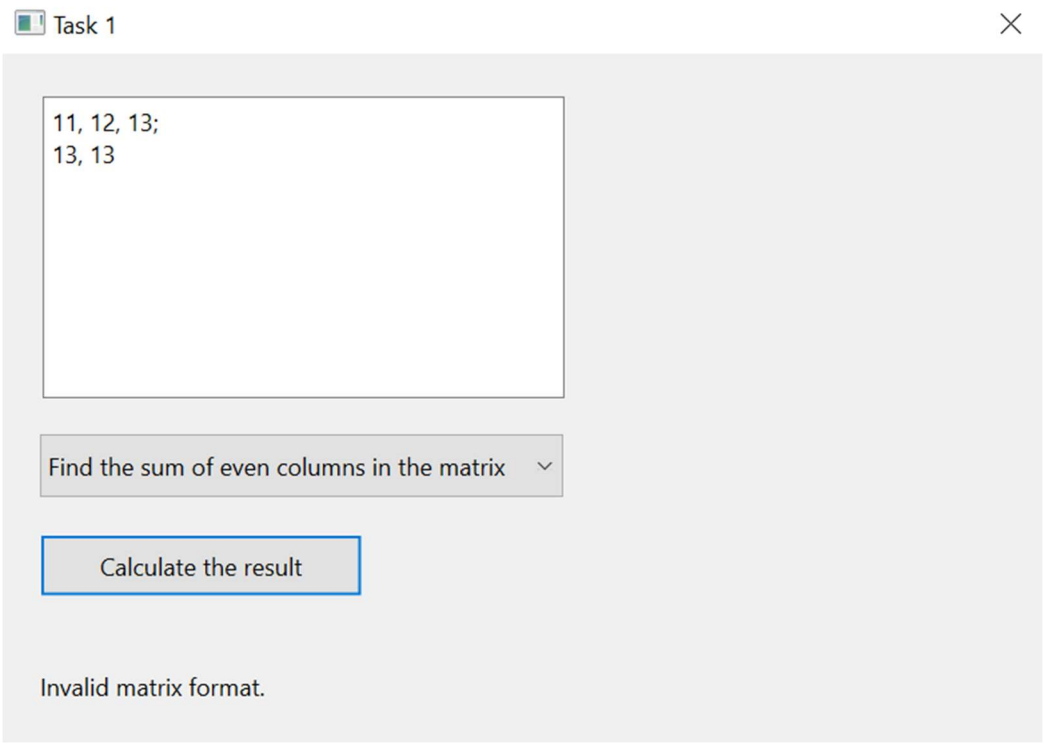
Скрін-шоти виконання завдань лабораторної роботи :

Завдання №1 :

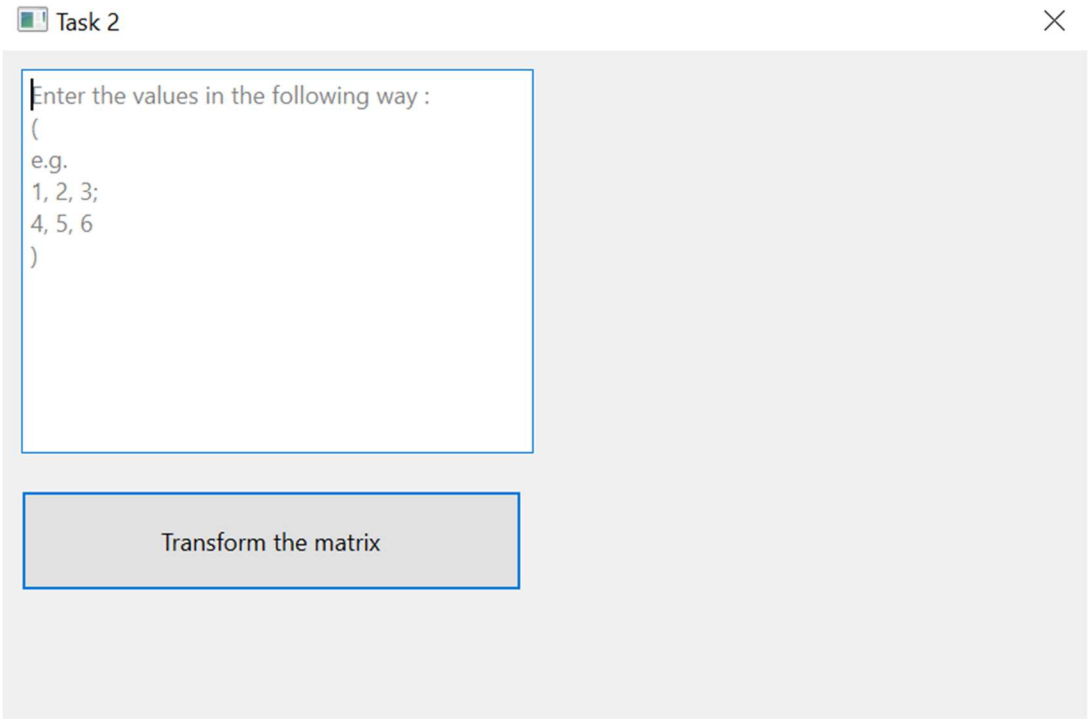


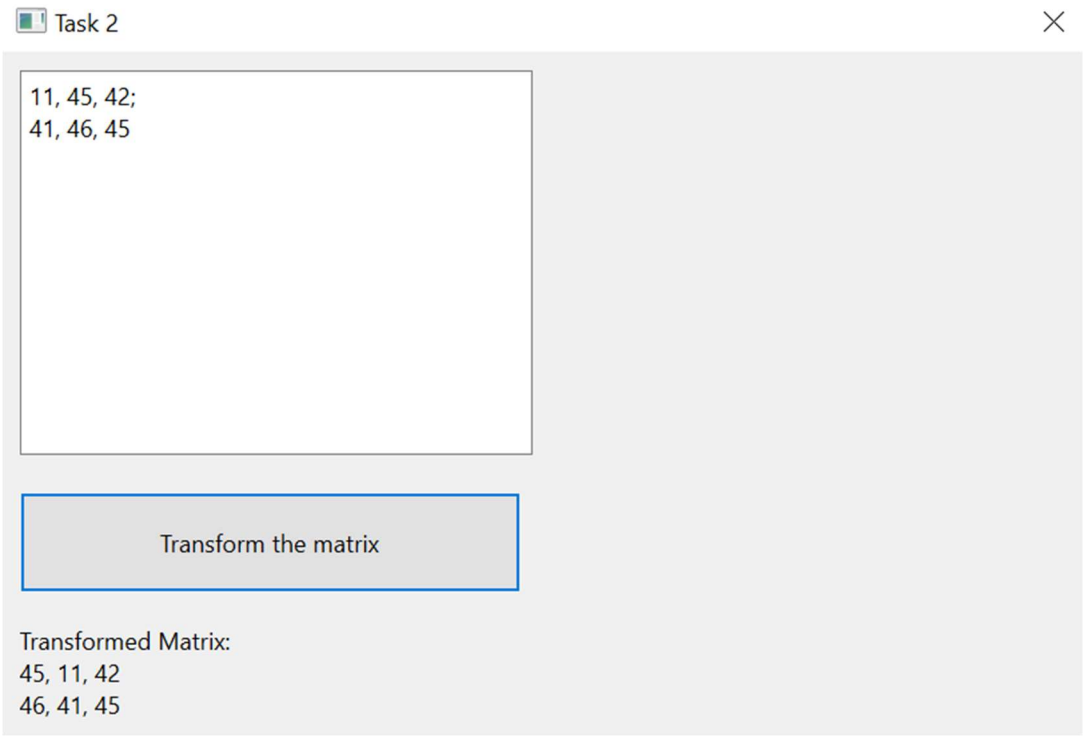

Task 1
11, 45, 42;
41, 46, 45
Find the sum of even columns in the matrix
Calculate the result
Sum of even columns: 139

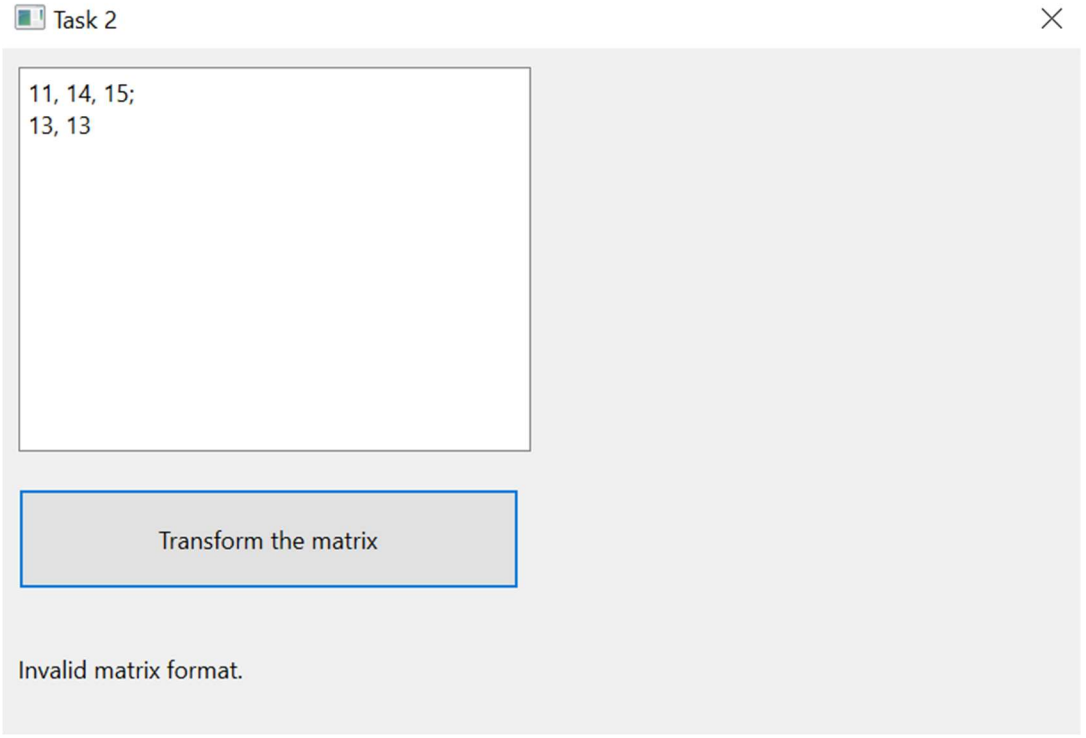

Task 1
11, 45, 42;
41, 46, 45
Find the sum of odd columns in the matrix
Calculate the result
Sum of odd columns: 91


Task 1
11, 12, 13;
13, 13
Find the sum of even columns in the matrix
Calculate the result
Invalid matrix format.

Завдання №2 :


Task 2
Enter the values in the following way :
(
e.g.
1, 2, 3;
4, 5, 6
)
Transform the matrix


Task 2
11, 45, 42;
41, 46, 45
Transform the matrix
Transformed Matrix:
45, 11, 42
46, 41, 45


Task 2
11, 14, 15;
13, 13
Transform the matrix
Invalid matrix format.