Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

за 5 семестр

По дисциплине «ООТПиСП»

Тема: «Контейнеры в Qt5»

Выполнил:

Студент 3 курса

Группы ПО-4(2)

Кречко К.А.

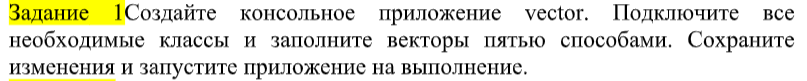
Проверила:

Хацкевич М.В.

Брест 2021

**Лабораторная работа №3**

**«Контейнеры в Qt5»**

****

**Код:**

#include <QTextStream>

#include <QVector>

int **main**()

{

QTextStream out(stdout);

out<<"First way==============================" << Qt::endl;

QVector <float> vec1(3);

vec1[0] = 1.0;

vec1[1] = 0.5;

vec1[2] = -0.4;

for(float val: vec1)

out<<val<<" ";

out << "\n\n";

out<<"Second way=============================="<< Qt::endl;

QVector<int> vec2(3,7);

for(float val: vec2)

out<<val<<" ";

out << "\n\n";

out<<"Third way=============================="<< Qt::endl;

QVector<int> vec3;

vec3.push\_back(1);

vec3.push\_back(2);

vec3.push\_back(3);

for(float val: vec3)

out<<val<<" ";

out << "\n\n";

out<<"Fourth way=============================="<< Qt::endl;

QVector<QString> vec4;

vec4.append("Monday");

vec4.append("Tuesday");

vec4.append("Wednesday");

for(const QString &val: vec4)

out<<val<<" ";

out << "\n\n";

out<<"Fiveth way=============================="<< Qt::endl;

QVector<bool> vec5;

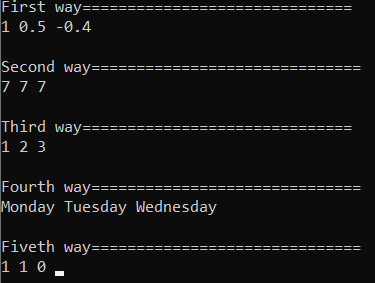
vec5<<0<<5<<1;

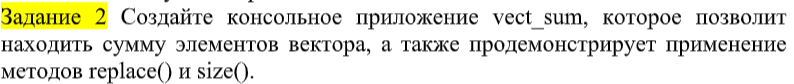
for(bool val: vec5)

out<<val<<" ";

}

**Результат:**

****

****

**Код:**

#include <QTextStream>

#include <QVector>

int **main**()

{

QTextStream out(stdout);

QVector <int> vec;

for (int i = 0; i < 5; i++)

vec.push\_back(i\*1.5);

out<<"Initial vector:\n";

for(int val: vec)

out<<val<<" ";

vec.replace(3,777);

out<<"\nVector, after replacements:\n";

for(int val: vec)

out<<val<<" ";

int sum = 0;

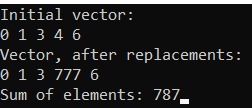
for(int i = 0; i < vec.size(); i++)

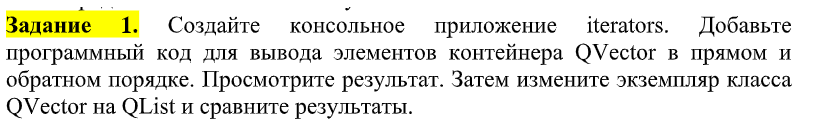
sum += vec[i];

out<<"\nSum of elements: " << sum;

}

**Результат:**

****

****

**Код:**

#include <QTextStream>

#include <QList>

#include <QVector>

int **main**()

{

QTextStream out(stdout);

QVector <int> vec;

for (int i = 0; i < 5; i++)

vec.push\_back(i\*1.5);

out<< "Vector==========================================\n";

QVector<int>::iterator it = vec.begin();

out<< "Direct order\n";

for(; it !=vec.end(); ++it)

out << "Element: "<<\*it << Qt::endl;

out<< "Reverse order\n";

it = vec.end();

for(; it !=vec.begin();){

--it;

out << "Element: "<<\*it << Qt::endl;

}

out<< "\nList============================================\n";

vec.toList();

it = vec.begin();

out<< "Direct order\n";

for(; it !=vec.end(); ++it)

out << "Element: "<<\*it << Qt::endl;

out<< "Reverse order\n";

it = vec.end();

for(; it !=vec.begin();){

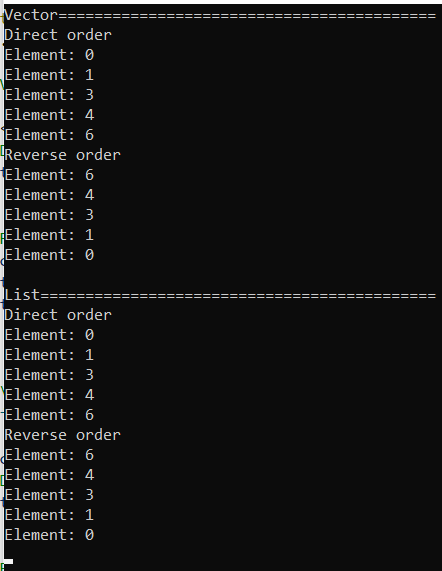
--it;

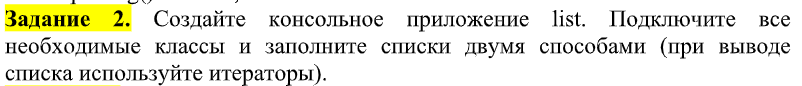
out << "Element: "<<\*it << Qt::endl;

}

}

**Результат:**

****

****

**Код:**

#include <QTextStream>

#include <QList>

#include <QVector>

int **main**()

{

QTextStream out(stdout);

out<< "First way============================\n";

QList<int> list;

list << 1<< 10 << 2021;

out << "Result: \n";

QVector<int>::iterator it = list.begin();

for(; it != list.end(); ++it)

out << \*it << "\n";

out<< "Second way============================\n";

QList<QString> list1;

list1.append("Hello");

list1.push\_back("World");

list1.push\_front(":)");

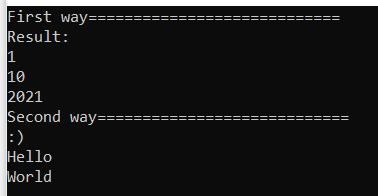
QVector<QString>::iterator it1 = list1.begin();

for(; it1 != list1.end(); ++it1)

out << \*it1 << "\n";

}

**Результат:**

****

****

**Код:**

**listgui.h**

#ifndef LISTGUI\_H

#define LISTGUI\_H

#include <QMainWindow>

#include <QWidget>

#include <QResizeEvent>

#include <QSize>

#include <QList>

#include <QStringList>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **listgui**; }

QT\_END\_NAMESPACE

class **listgui** : public QMainWindow

{

Q\_OBJECT

public:

**listgui**(QWidget \*parent = nullptr);

~***listgui***();

QList<int> list;

private slots:

void **on\_pushButton\_clicked**();

private:

Ui::listgui \*ui;

};

#endif // LISTGUI\_H

**listgui.cpp**

#include "listgui.h"

#include "ui\_listgui.h"

listgui::**listgui**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::listgui)

{

ui->setupUi(this);

for(int i = 0; i < 10; i++){

list.append(10+i\*2);

ui->textEdit->append(QString::number(list.at(i)) + " ");}

}

listgui::~***listgui***()

{

delete ui;

}

void listgui::**on\_pushButton\_clicked**()

{

if (ui->radioButton->isChecked())

list.removeLast();

else if (ui->radioButton\_2->isChecked()){

int n = ui->lineEdit->text().toInt();

list.push\_front(n);

}

else if (ui->radioButton\_3->isChecked()){

int a,b;

QStringList st = ui->lineEdit\_2->text().split(",");

a = st[0].toInt();

b = st[1].toInt();

list.swapItemsAt(a,b);

}

ui->textEdit->clear(); //перезаписываем список

QList<int>::iterator it = list.begin();

while (it != list.end()){

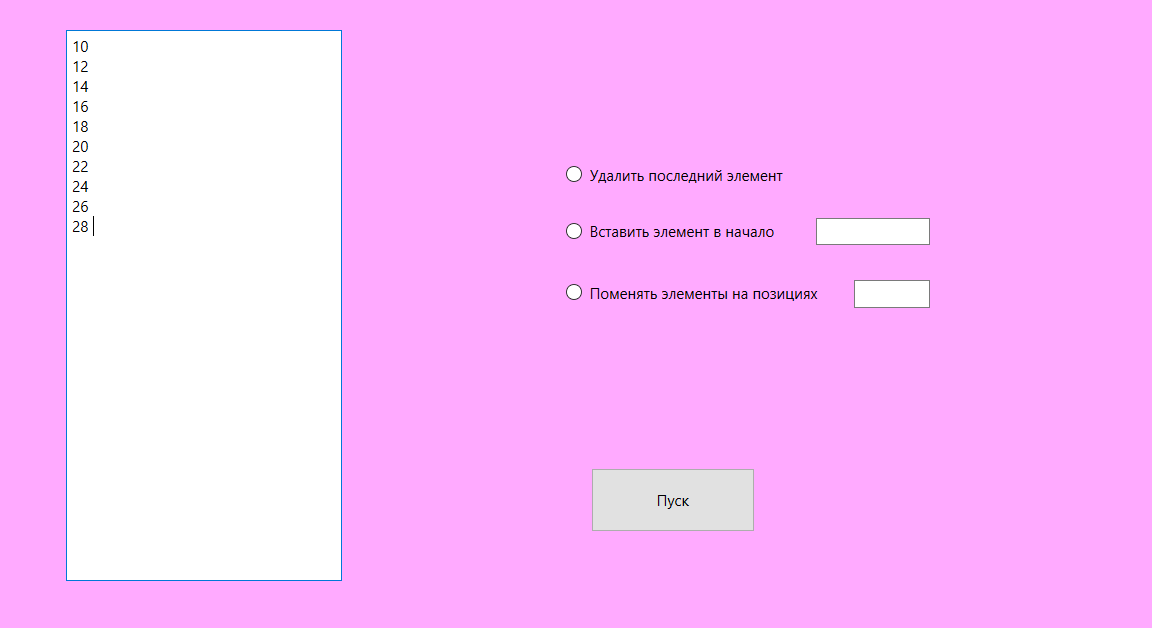
ui->textEdit->append(QString::number(\*it) + " ");

it++;

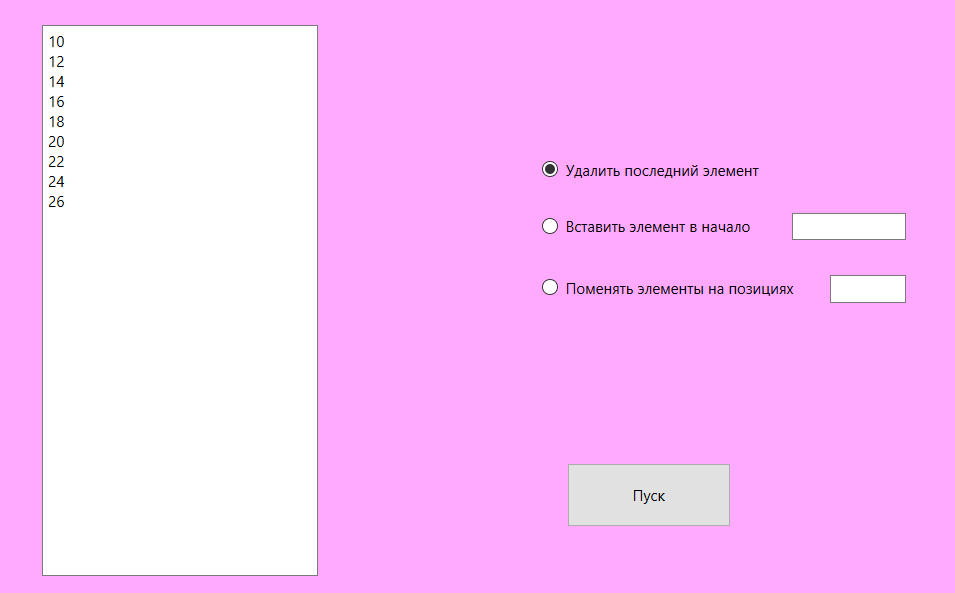
}

}

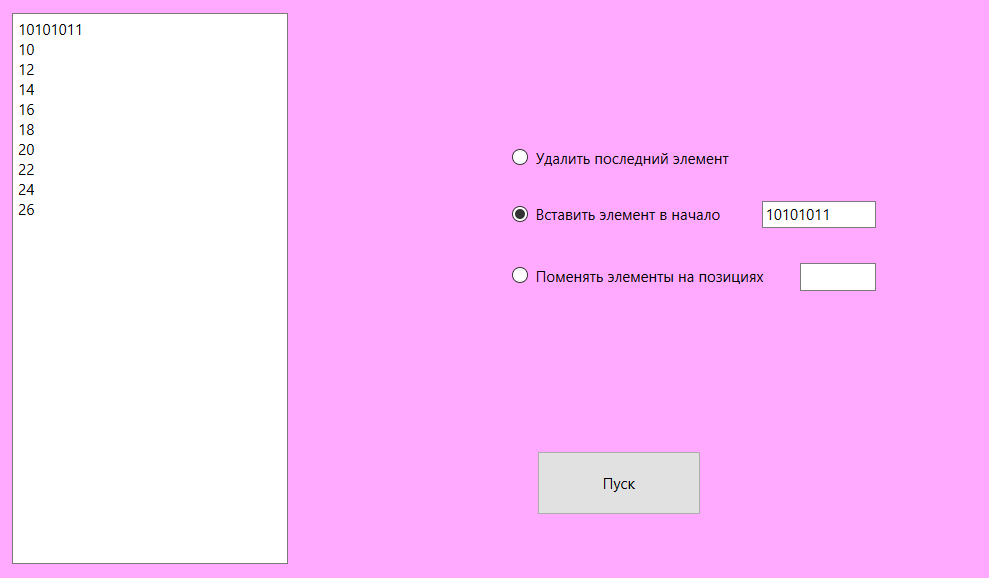
**Результат:**

****

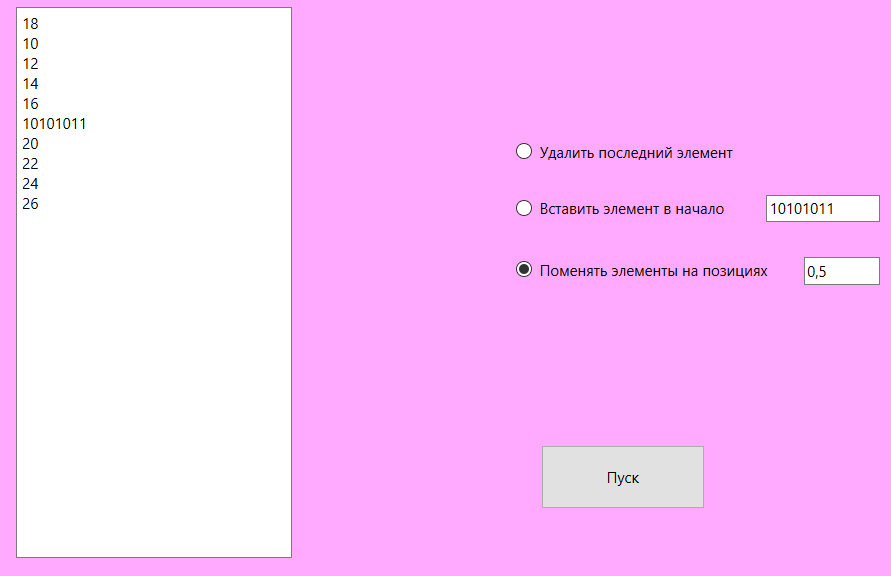
**удаление последнего элемента**

****

**вставка в начало**

****

**смена элементов на позициях**

****

****

**Код:**

#include <QCoreApplication>

#include <QStack>

#include <QTextStream>

#include <QDebug>

int **main**()

{

QStack<int> stack\_1, stack\_2;

QTextStream in(stdin);

qDebug() << "Number of elements";

int count = (in.readLine()).toInt();

qDebug() << "\nInput elements: \n";

for (int i = 0; i < count; i++){

int num = (in.readLine()).toInt();

stack\_1.push(num);

}

while (!stack\_1.empty()){

int num = stack\_1.pop();

if (num % 2 == 0)

num += 3;

stack\_2.push(num);

}

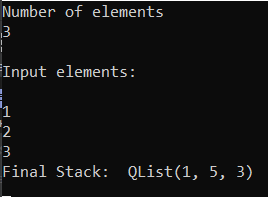
while(!stack\_2.empty())

stack\_1.push(stack\_2.pop());

qDebug() << "Final Stack: " << stack\_1;

}

**Результат:**

****

****

**Код:**

**tower.h**

#ifndef TOWER\_H

#define TOWER\_H

#include <QMainWindow>

#include <QStack>

#include <QMessageBox>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **tower**; }

QT\_END\_NAMESPACE

class **tower** : public QMainWindow

{

Q\_OBJECT

public:

**tower**(QWidget \*parent = nullptr);

~***tower***();

private slots:

void **on\_pushButton\_clicked**();

private:

QStack <int> stack1;

QStack <int> stack2;

QStack <int> stack3;

Ui::tower \*ui;

};

#endif // TOWER\_H

**tower.cpp**

#include "tower.h"

#include "ui\_tower.h"

tower::**tower**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::tower)

{

ui->setupUi(this);

for (int i = 1; i <= 5; i++){

stack1.push(i);

ui->textEdit->append(QString::number((stack1.at(i))));

}

}

tower::~***tower***()

{

delete ui;

}

void tower::**on\_pushButton\_clicked**()

{

if(ui->spinBox->value() == ui->spinBox\_2->value())

QMessageBox::warning(this, "Warning!", "it's forbidden");

else{

QStack <int> \*from\_st;

QStack <int> \*to\_st;

switch(ui->spinBox->value()){

case 1: {from\_st = &stack1; break;}

case 2: {from\_st = &stack2; break;}

case 3: {from\_st = &stack3; break;}

}

switch (ui->spinBox\_2->value()){

case 1: {to\_st = &stack1; break;}

case 2: {to\_st = &stack2; break;}

case 3: {to\_st = &stack3; break;}

}

if (from\_st->isEmpty())

QMessageBox::warning(this, "Warning!", "it's forbidden");

else

if((!(to\_st->isEmpty())) ||

(from\_st->at(from\_st->count()-1)>

to\_st->at(to\_st->count()-1)))

QMessageBox::warning(this, "Warning!", "it's forbidden");

else{

int val;

val = from\_st->pop();

to\_st->push(val);

}

}

ui->textEdit->clear();

ui->textEdit\_2->clear();

ui->textEdit\_3->clear();

for (int i = stack1.count()-1; i>=0; i--)

ui->textEdit->append(QString::number((stack1.at(i))));

for (int i = stack2.count()-1; i>=0; i--)

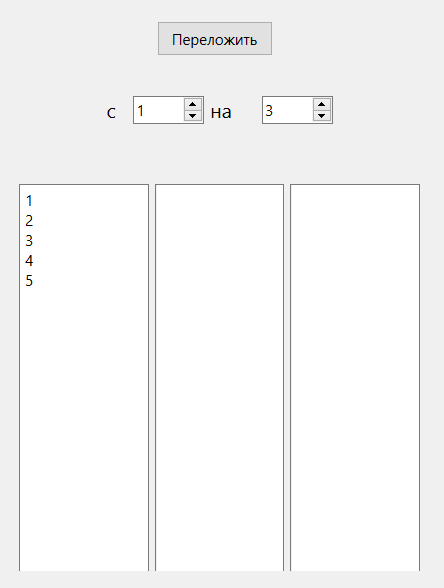
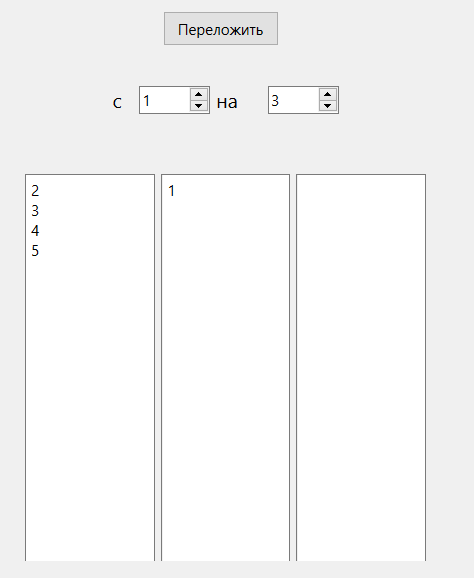
ui->textEdit\_2->append(QString::number((stack2.at(i))));

for (int i = stack3.count()-1; i>=0; i--)

ui->textEdit\_3->append(QString::number((stack3.at(i))));

}

**Результат:**

** **

****

**Код:**

#include <QCoreApplication>

#include <QQueue>

#include <QTextStream>

int **main**()

{

QQueue<int> que;

QTextStream out(stdout);

out<<"Start queue\n";

for (int i = 0; i < 10; i++)

{

que.push\_back(rand()%10);

out << que[i] << " ";

if (que[i] % 2 == 0) que[i] += 3;

}

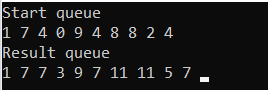
out<<"\nResult queue\n";

for (int i = 0; i < 10; i++)

out << que[i] << " ";

}

**Результат:**

****

****

**Код:**

#include <QCoreApplication>

#include <QMap>

#include <QDebug>

int **main**()

{

qDebug() << "First way:";

QMap<QString, int> map;

map["Aristarh"] = 20;

map["Kiril"] = 20;

map["Kolya"] = 19;

QMap<QString, int>::iterator id = map.begin();

for(; id != map.end(); ++id)

qDebug() << "key: "<<id.key() << " value: " << id.value();

qDebug() << "\nSecond way:";

QMap<QString, int> map2;

map2.insert("Hurma", 2);

map2.insert("Kivi", 3);

map2.insert("Apple", 5);

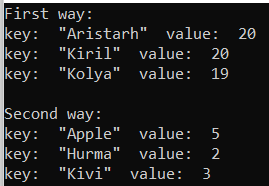
QMap<QString, int>::iterator id2 = map2.begin();

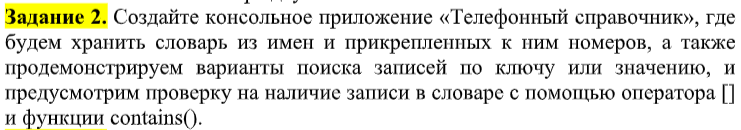
for(; id2 != map.end(); ++id2)

qDebug() << "key: "<<id2.key() << " value: " << id2.value();

}

**Результат:**

****

****

**Код:**

#include <QCoreApplication>

#include <QMap>

#include <QDebug>

#include <QTextStream>

int **main**()

{

qDebug() << "Phonebook";

QTextStream in(stdin);

QMap<QString, int> phonebook;

phonebook["Andrey"] = 1546248;

phonebook["Gena"] = 880035535;

phonebook["Serega"] = 7788;

qDebug() << "Key search:\n"

"Input name: ";

QString name = in.readLine();

if(phonebook.contains(name))

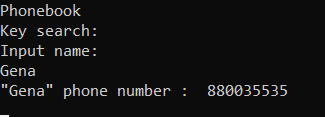
qDebug() << name << "phone number" <<": " << phonebook[name];

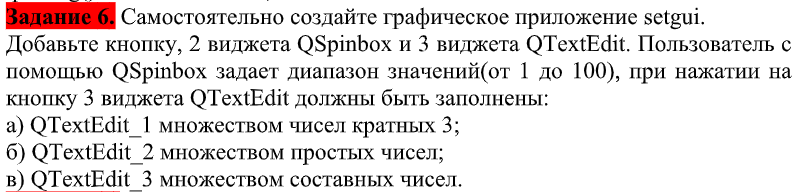
else

qDebug() << "Name is epsent";

}

**Результат:**

****

****

**Код:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QSet>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

private:

Ui::MainWindow \*ui;

QSet <int> set = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97};

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

int min = ui->spinBox->value();

int max = ui->spinBox\_2->value();

for (int i = min; i < max; i++){

if (i % 3 == 0)

ui->textEdit\_5->append(QString::number(i));

if (set.contains(i) && i != 1)

ui->textEdit\_6->append(QString::number(i));

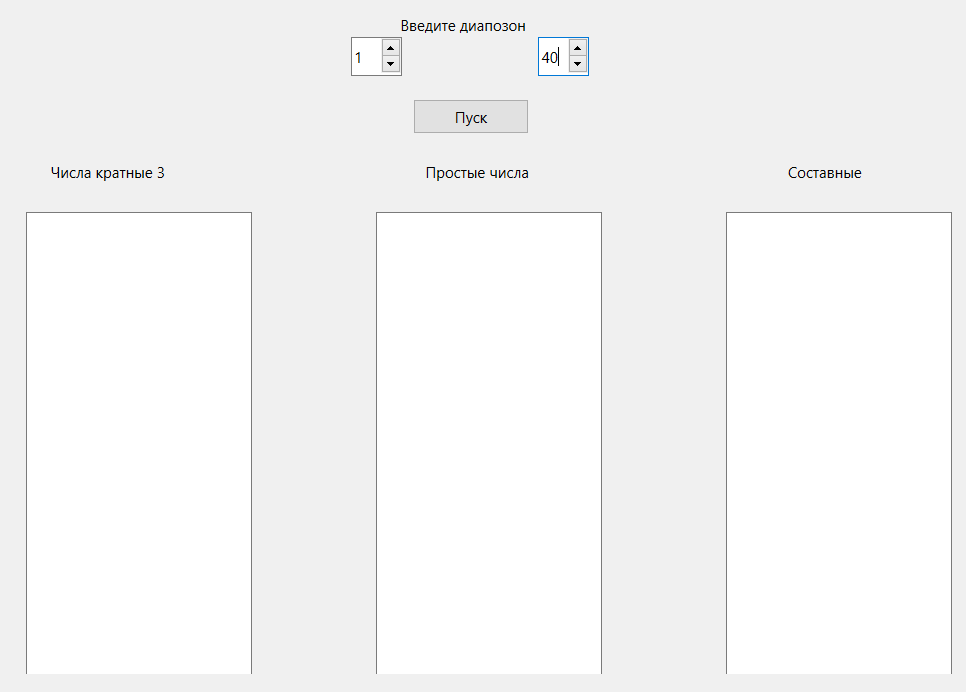
if (!set.contains(i) && i != 1)

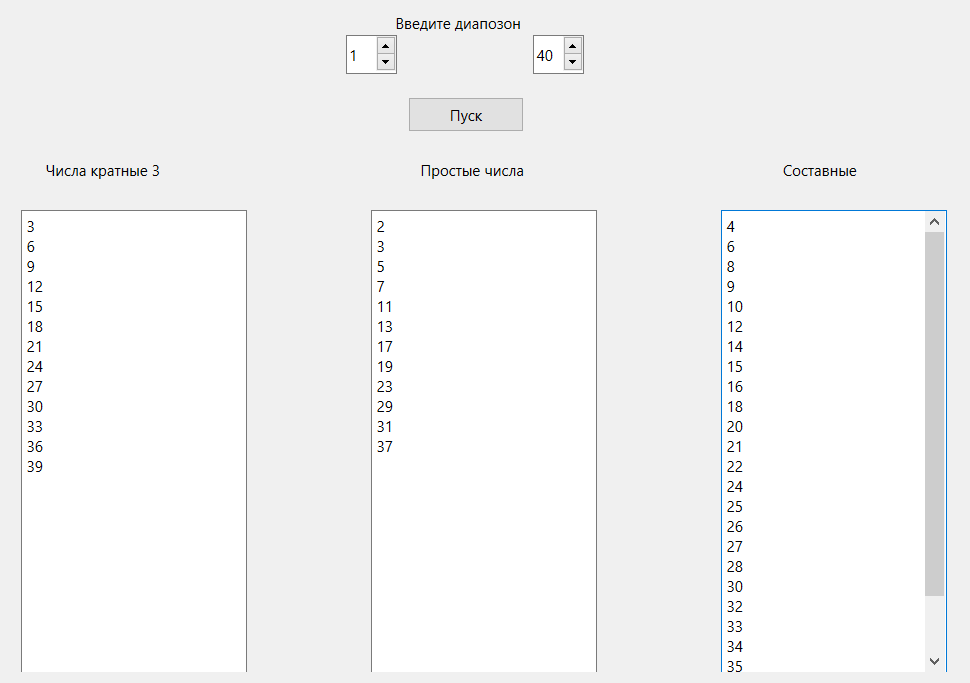
ui->textEdit\_7->append(QString::number(i));

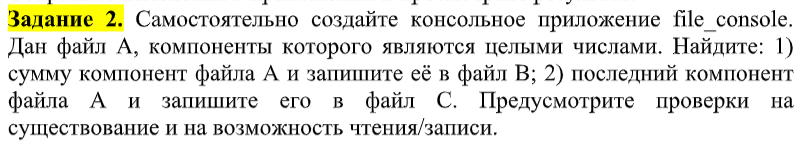
}

}

**Результат:**

****

****

****

**Код:**

**main.cpp**

#include <QCoreApplication>

#include <QDebug>

#include <QFile>

#include <QVector>

int **main**()

{

QFile A ("D:\\Qt projects\\OOTPiSP\_LAB3\\A.txt");

if (!A.exists()) qDebug() << "Can't find file A";

else

{

if(!A.*open*(QIODevice::ReadOnly)) qDebug() << "Can't open file A";

else

{

QTextStream stream(*&A*);

QString buffer = stream.readAll();

QStringList numbers = buffer.split(" ");

int sum = 0;

for (int i = 0; i < numbers.length(); i++)

sum += numbers[i].toInt();

QFile B ("D:\\Qt projects\\OOTPiSP\_LAB3\\B.txt");

if (!B.exists()) qDebug() << "Can't find file B";

if(!B.*open*(QIODevice::WriteOnly)) qDebug() << "Can't open file B";

else{

QTextStream streamB(*&B*);

streamB << sum;

B.*close*();

qDebug() << "File B is ready";

}

QFile C ("D:\\Qt projects\\OOTPiSP\_LAB3\\C.txt");

if (!C.exists()) qDebug() << "Can't find file C";

if(!C.*open*(QIODevice::WriteOnly)) qDebug() << "Can't open file C";

else{

QTextStream streamC(*&C*);

streamC << numbers[numbers.length() - 1];

C.*close*();

qDebug() << "File C is ready";

}

A.*close*();

}

}

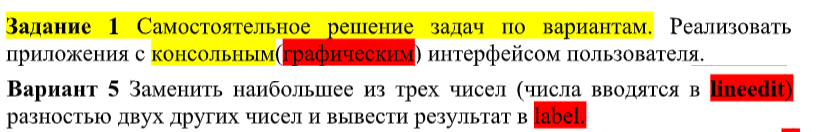
}

****

**Результат:**

** **

****

****

**Код:**

**mainwindow.cpp:**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

int **find\_biggest**(QVector<int> vec){

int biggest = 0;

int biggest\_index = 0;

for(int i = 0; i < 3; i++)

if (biggest < vec[i]){

biggest = vec[i];

biggest\_index = i;

}

return biggest\_index;

}

void MainWindow::**on\_pushButton\_clicked**()

{

QTextStream out(stdout);

QStringList st = ui->lineEdit->text().split(" ");

QVector<int> vec;

for(int i = 0; i < 3; i++){

if (st[i].toInt()>999 || st[i].toInt()<-999){

QMessageBox::warning(this, "Warning!", "it's forbidden");

st[i] = "0";

vec.push\_back(0);

}else

vec.push\_back(st[i].toInt());

}

switch (find\_biggest(vec)){

case 0: {int i = abs(st[2].toInt()-st[1].toInt());

ui->label->setText(QString::number(i)); break;}

case 1: {int i = abs(st[2].toInt()-st[0].toInt());

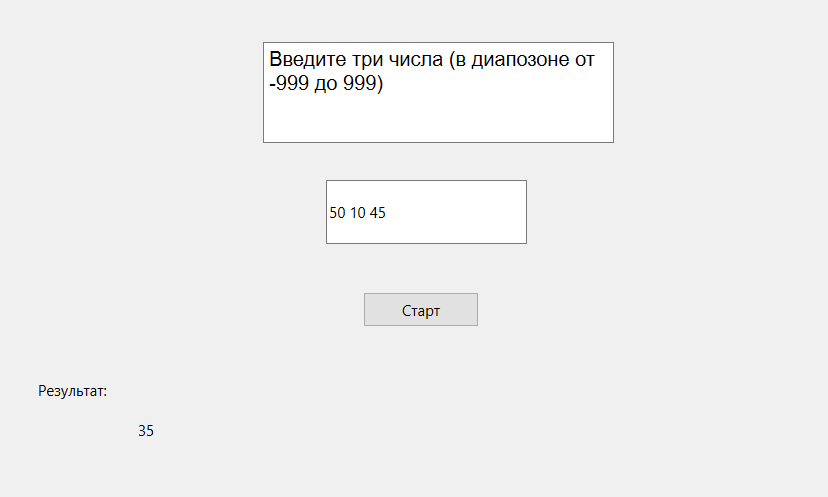
ui->label->setText(QString::number(i));break;}

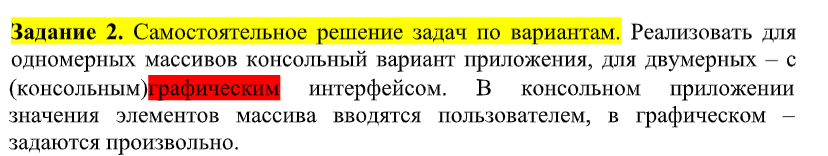
case 2: {int i = abs(st[1].toInt()-st[0].toInt());

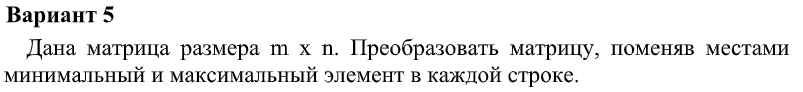
ui->label->setText(QString::number(i));break;}

}}

**Результат:**

****

****

****

**Код:**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QString str;

for(int i = 0; i < 3; i++){

for(int j = 0; j < 3; j++){

str.append(tr("%1 ").arg(arr[i][j]));

}

str.append('\n');

}

ui->textEdit->setText(str);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

int **find\_biggest**(int arr[]){

int biggest = 0;

int biggest\_index = 0;

for(int i = 0; i < 3; i++)

if (biggest < arr[i]){

biggest = arr[i];

biggest\_index = i;

}

return biggest\_index;

}

int **find\_smallest**(int arr[]){

int smallest = 0;

int smallest\_index = 0;

for(int i = 0; i < 3; i++)

if (smallest > arr[i]){

smallest = arr[i];

smallest\_index = i;

}

return smallest\_index;

}

void MainWindow::**on\_pushButton\_clicked**()

{

int arr1[3];

QTextStream out(stdout);

for(int i = 0; i < 3; i++){

for(int j = 0; j < 3; j++)

arr1[j] = arr[i][j];

int biggest = find\_biggest(arr1);

int smallest = find\_smallest(arr1);

int temp = arr1[biggest];

arr1[biggest] = arr1[smallest];

arr1[smallest] = temp;

for(int x = 0; x < 3; x++)

arr[i][x] = arr1[x];

}

QString str;

for(int i = 0; i < 3; i++){

for(int j = 0; j < 3; j++){

str.append(tr("%1 ").arg(arr[i][j]));

}

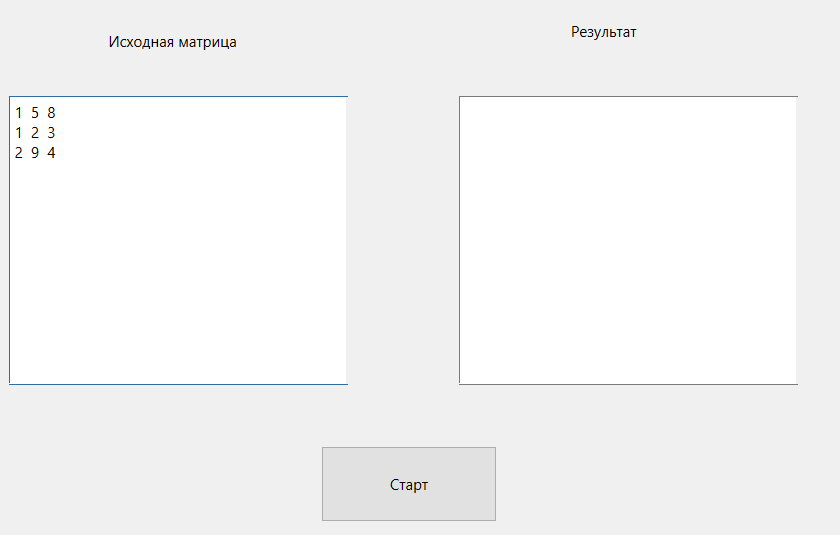
str.append('\n');

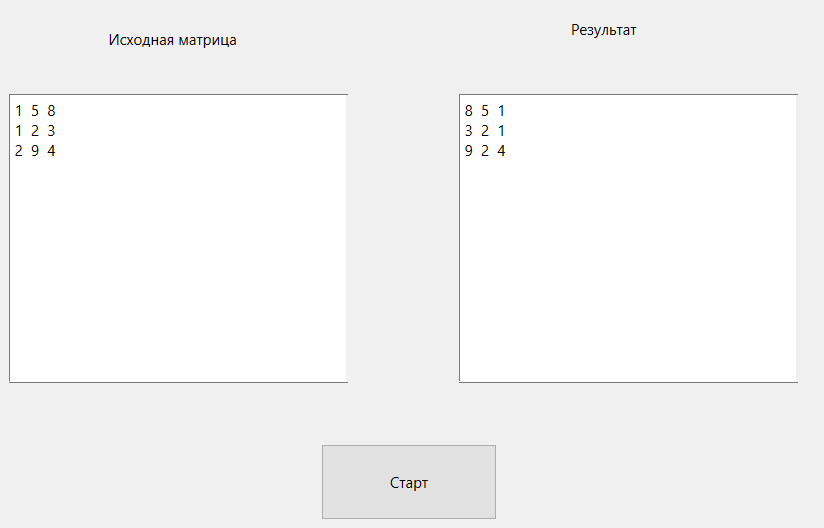
}

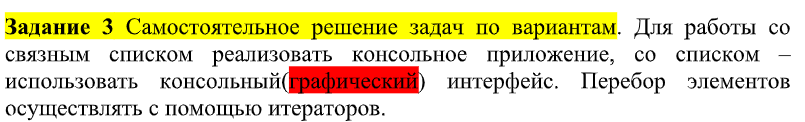
ui->textEdit\_2->setText(str);

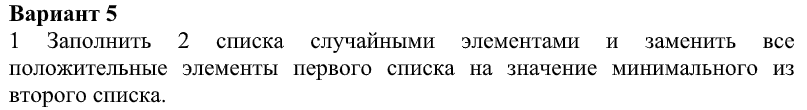
}

**Результат:**

****

****

****

****

**Код:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QList>

#include <QVector>

#include <QTextStream>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

private:

Ui::MainWindow \*ui;

QList<int> list1;

QList<int> list2;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

int **find\_smallest**(QList<int> list){

int smallest = list[0];

for(int i = 0; i < list.size(); i++){

if (smallest > list[i]){

smallest = list[i];

}

}

return smallest;

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

for(int i = 0; i < 10; i++){

list1.append(rand()%100-50);

list2.append(rand()%100-50);

ui->textEdit->append(QString::number(list1.at(i)) + " ");

ui->textEdit\_2->append(QString::number(list2.at(i)) + " ");

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

int smallest = find\_smallest(list2);

QVector<int>::iterator it = list1.begin();

for(; it !=list1.end(); ++it){

if (\*it > 0)

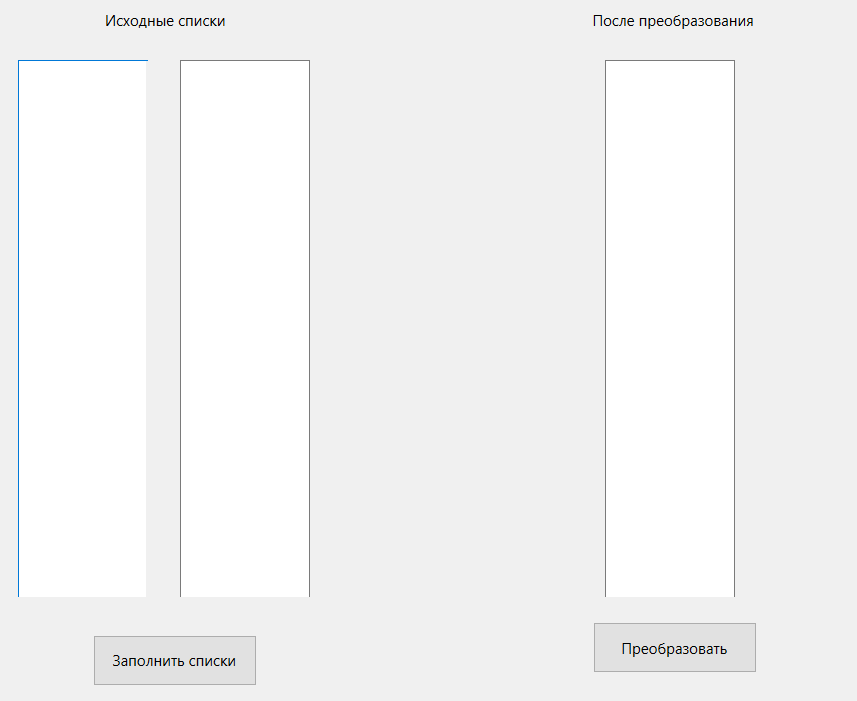
\*it = smallest;

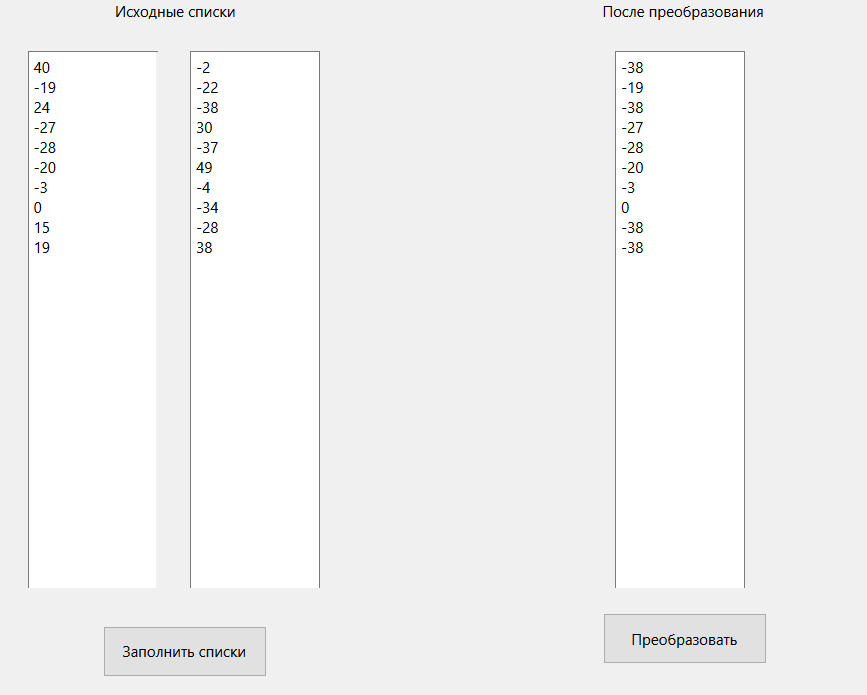
ui->textEdit\_3->append(QString("%1").arg(\*it));

}

}

**Результат:**

****

****

****

**Код:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QList>

#include <QVector>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

private:

Ui::MainWindow \*ui;

QList<int> list;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

for(int i = 0; i < 10; i++){

list.append(rand()%100);

ui->textEdit->append(QString::number(list.at(i)) + " ");

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

int k = ui->lineEdit->text().toInt();

QVector<int>::iterator it = list.begin();

for(; it !=list.end(); ++it){

if (\*it > k)

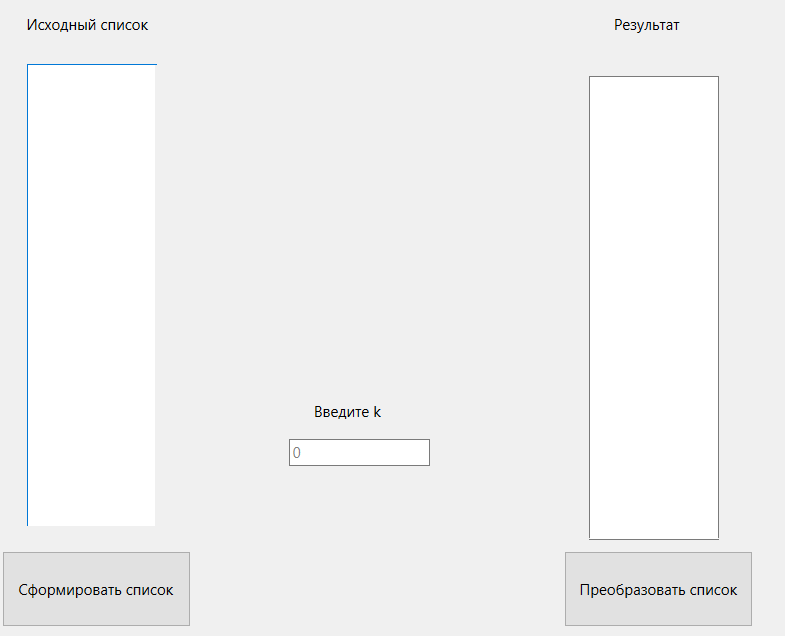
continue;

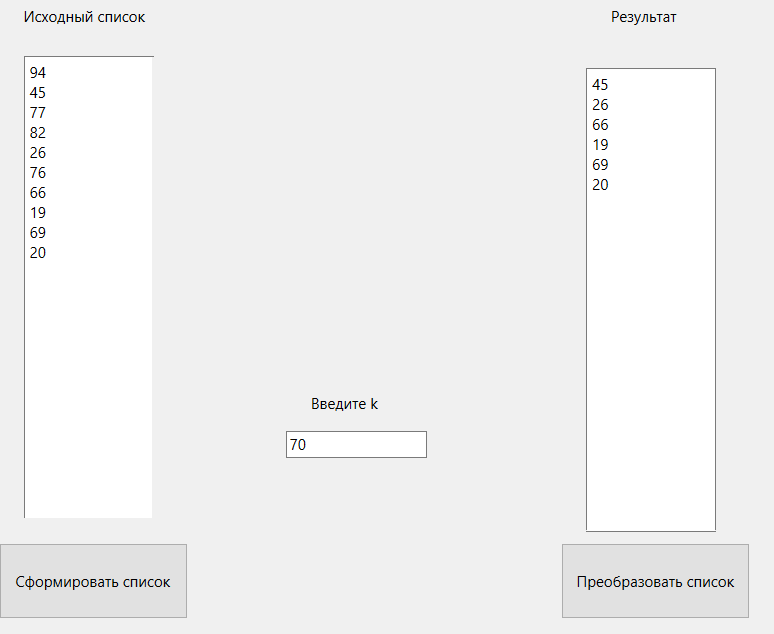
ui->textEdit\_2->append(QString::number(\*it));

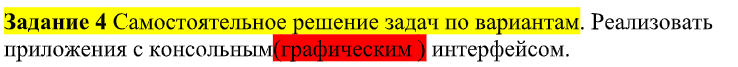
}

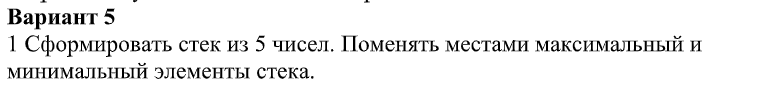
}

**Результат:**

****

****

****

****

**Код:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QStack>

#include <QTextStream>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

int **find\_biggest**(int arr[]);

int **find\_smallest**(int arr[]);

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

private:

Ui::MainWindow \*ui;

QStack<int> stack;

const int SIZE = 5;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

int MainWindow::**find\_biggest**(int arr[]){

int biggest = arr[0];

int biggest\_index = 0;

for(int i = 0; i < SIZE; i++)

if (biggest < arr[i]){

biggest = arr[i];

biggest\_index = i;

}

return biggest\_index;

}

int MainWindow::**find\_smallest**(int arr[]){

int smallest = arr[0];

int smallest\_index = 0;

for(int i = 0; i < SIZE; i++)

if (smallest > arr[i]){

smallest = arr[i];

smallest\_index = i;

}

return smallest\_index;

}

void MainWindow::**on\_pushButton\_clicked**()

{

for(int i = 0; i < SIZE; i++){

stack.push(rand()%100);

ui->textEdit->append(QString::number((stack.at(i))));

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

int arr[SIZE];

QTextStream out(stdout);

if (!stack.isEmpty())

for(int i = 0; i < SIZE; i++)

arr[i] = stack.pop();

int biggest = find\_biggest(arr);

int smallest = find\_smallest(arr);

int temp = arr[biggest];

arr[biggest] = arr[smallest];

arr[smallest] = temp;

for(int i = SIZE-1; i >= 0; i--){

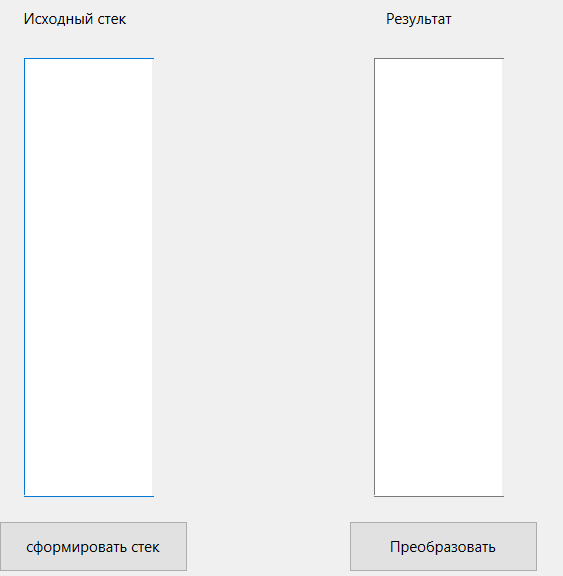
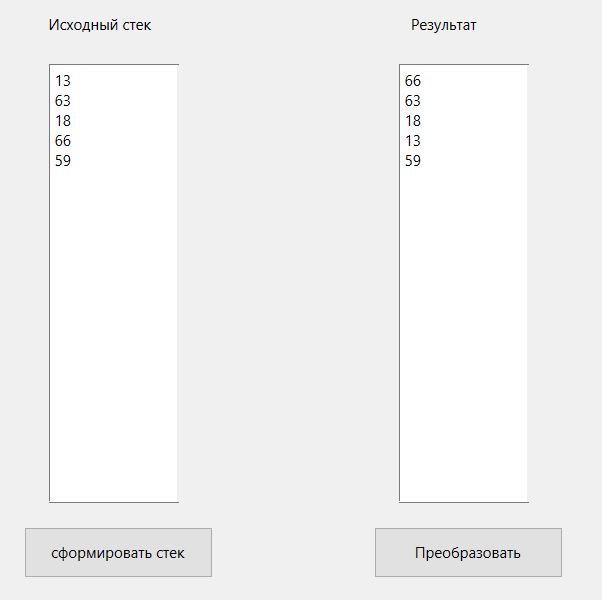
stack.push(arr[i]);

ui->textEdit\_2->append(QString::number((stack.top())));

}

}

**Результат:**

** **

****

**Код:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QQueue>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

private:

Ui::MainWindow \*ui;

QQueue<int> que;

int average = 0;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

for (int i = 0; i < 8; i++){

que.enqueue(rand()%50);

ui->textEdit->append(QString::number(que[i]));

average += que[i];

}

average /= 8;

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

for (int i = 0; i < 8; i++){

if(que[i]%2 == 0)

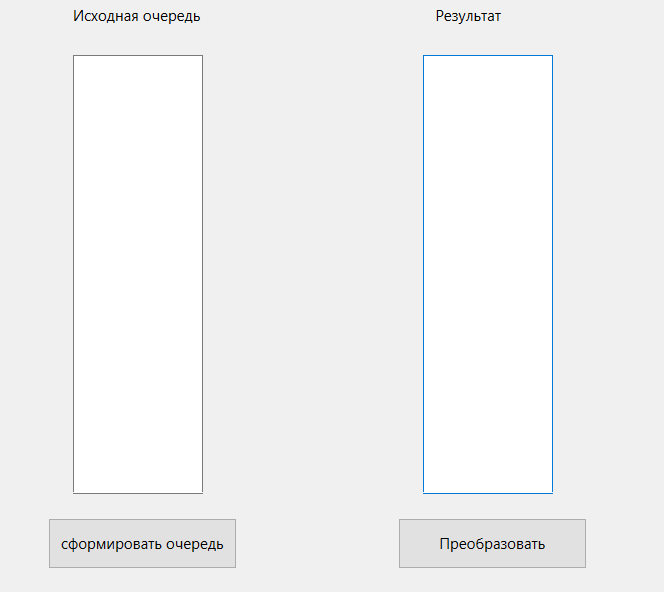
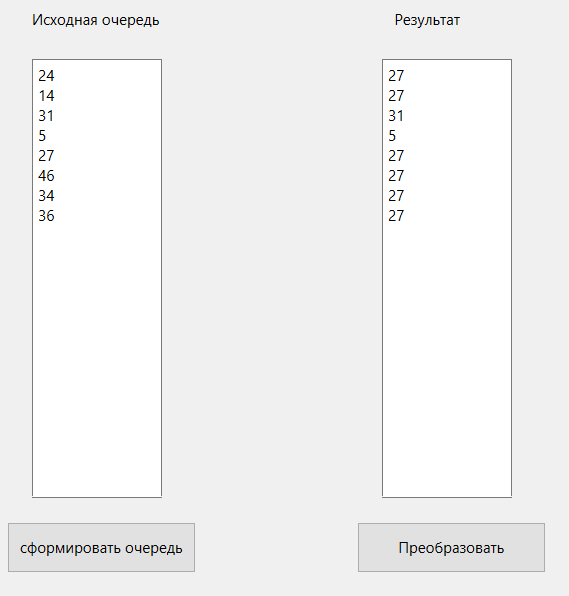
que[i] = average;

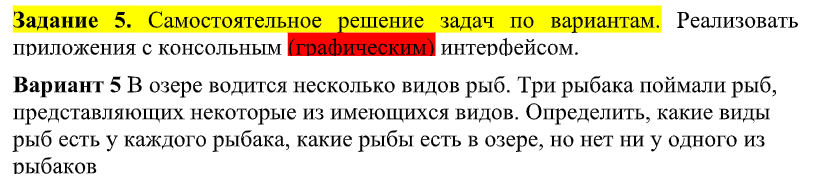
ui->textEdit\_2->append(QString::number(que[i]));

}

}

**Результат:**

** **

****

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QList>

#include <QTextEdit>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

void **on\_pushButton\_3\_clicked**();

void **on\_pushButton\_4\_clicked**();

private:

QList<QString> fish\_list = {"Plotva", "Karp", "Shchuka", "Okun", "Osetr"};

QList<QString> first;

QList<QString> second;

QList<QString> third;

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

int size = rand() % 6 + 1;

for (int i = 0; i < size; i++){

QString temp = fish\_list[rand()%5];

first.append(temp);

ui->textEdit->append(temp);

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

int size = rand() % 6 + 1;

for (int i = 0; i < size; i++){

QString temp = fish\_list[rand()%5];

second.append(temp);

ui->textEdit\_2->append(temp);

}

}

void MainWindow::**on\_pushButton\_3\_clicked**()

{

int size = rand() % 6 + 1;

for (int i = 0; i < size; i++){

QString temp = fish\_list[rand()%5];

third.append(temp);

ui->textEdit\_3->append(temp);

}

}

void MainWindow::**on\_pushButton\_4\_clicked**()

{

ui->textEdit\_4->clear();

ui->textEdit\_5->clear();

for (int i = 0; i < fish\_list.count(); i++) {

if(first.contains(fish\_list[i]) && second.contains(fish\_list[i]) && third.contains(fish\_list[i]))

ui->textEdit\_4->append(fish\_list[i]);

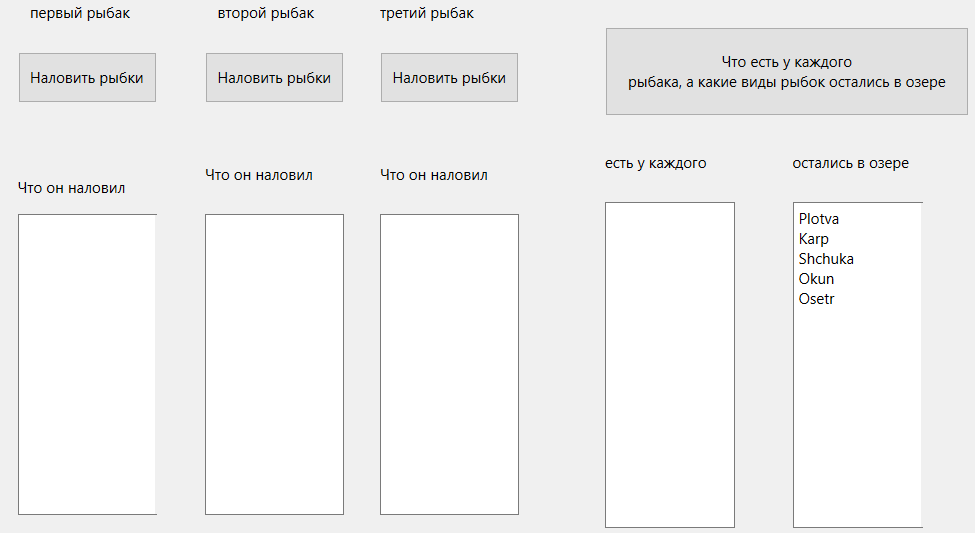
if(!first.contains(fish\_list[i]) && !second.contains(fish\_list[i]) && !third.contains(fish\_list[i]))

ui->textEdit\_5->append(fish\_list[i]);

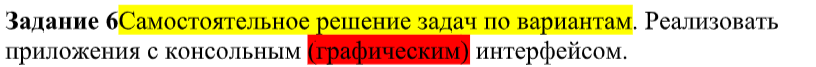
}

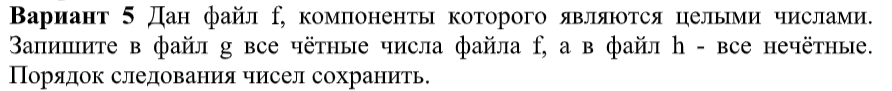
}

**Результат:**

****

****

****

****

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QFile>

#include <QVector>

#include <QTextStream>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

QFile f ("D:\\Qt projects\\OOTPiSP\_LAB3\_DIY\\Task\_6\\f.txt");

if (!f.exists()) {

qDebug() << "Can't find file f";

return;

}

else {

if(!f.*open*(QIODevice::ReadOnly)) {

qDebug() << "Can't open file f";

return;

}

else {

QTextStream stream(*&f*);

QString buffer = stream.readAll();

QStringList stringNumbers = buffer.split(" ");

for (int i = 0; i < stringNumbers.length(); i++){

intNumbers.append(stringNumbers[i].toInt());

ui->textEdit->append(stringNumbers[i]);

}

f.*close*();

}

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

QFile g ("D:\\Qt projects\\OOTPiSP\_LAB3\_DIY\\Task\_6\\g.txt");

if (!g.exists()) {

qDebug() << "Can't find file g";

return;

}

else {

if(!g.*open*(QIODevice::ReadOnly)) {

qDebug() << "Can't open file g";

return;

} else {

QTextStream streamG(*&g*);

QFile h ("D:\\Qt projects\\OOTPiSP\_LAB3\_DIY\\Task\_6\\h.txt");

if (!h.exists()) {

qDebug() << "Can't find file h";

return;

}

else {

if(!h.*open*(QIODevice::ReadOnly)) {

qDebug() << "Can't open file h";

return;

}

else {

QTextStream streamH(*&h*);

for (int i = 0; i < intNumbers.count(); i++){

if (intNumbers[i] % 2 == 0) {

streamG << intNumbers[i] << " ";

ui->textEdit\_2->append(QString::number(intNumbers[i]));

} else {

streamH << intNumbers[i] << " ";

ui->textEdit\_3->append(QString::number(intNumbers[i]));

}

}

g.*close*();

h.*close*();

}

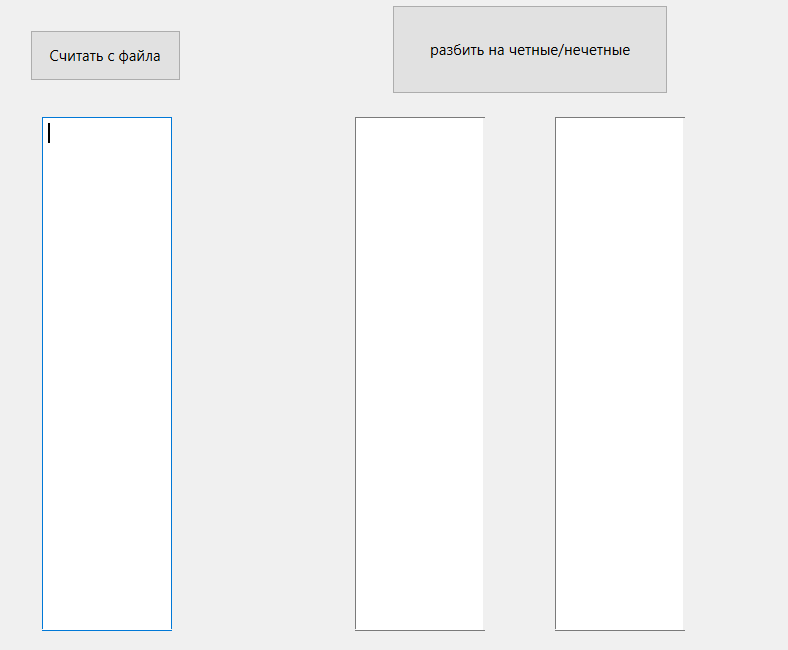
}

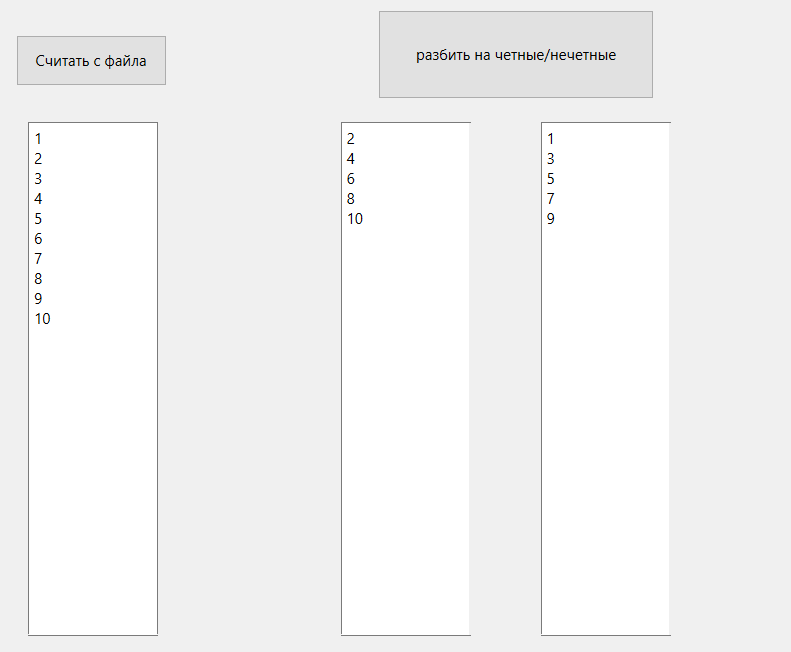
}

}

}

**Результат:**

****

****