INDUSTRIAL QUANTUM ROBOT

WEEK 2

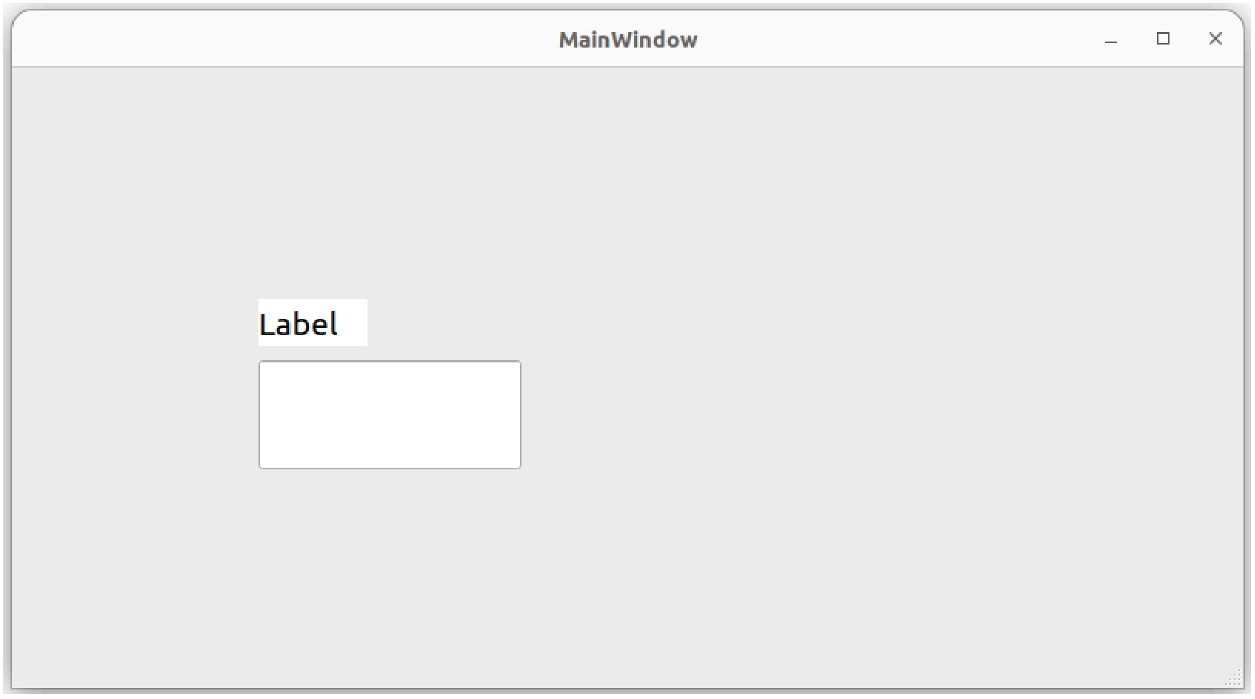
Name : Vivitha M
Roll No : 7376221CS352

1.Create a QT label and line edit.

Coding:

```
#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_lineEdit_textChanged(const QString &arg1)
{

}
```

Output:

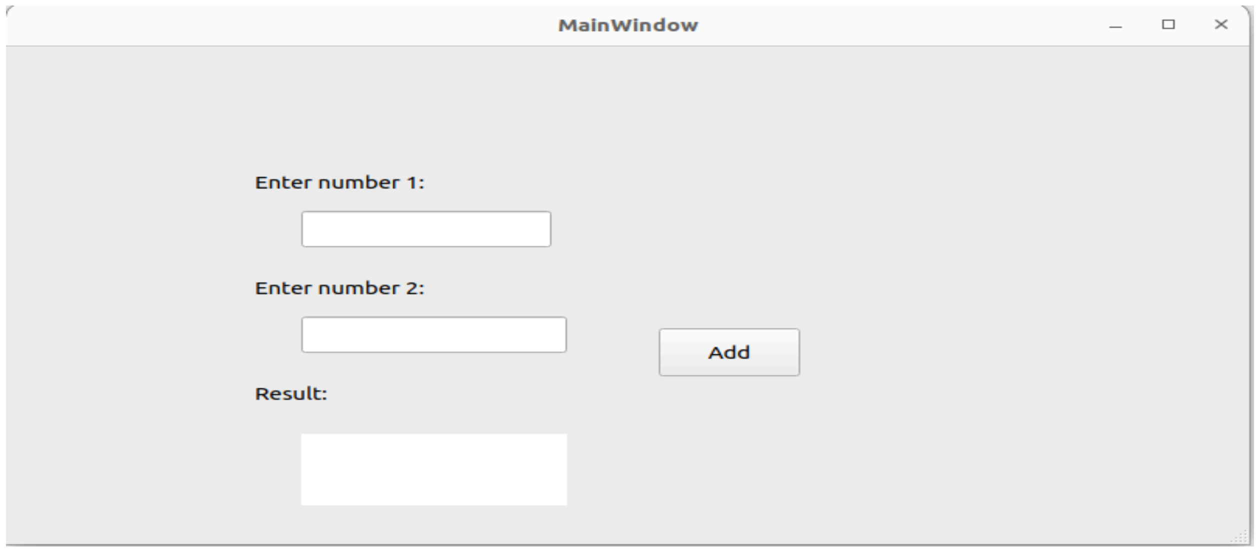


2. Create a QT Push button line edit label for addition.

Coding:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
float number1, number2, result = 0;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{       ui->setupUi(this);
}
MainWindow::~MainWindow()
{       delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{       number1 = arg1.toFloat();
}
void MainWindow::on_lineEdit_2_textChanged(const QString &arg1)
{
        number2 = arg1.toFloat();
}
void MainWindow::on_pushButton_clicked()
{
        result = number1 + number2;
        ui->label->setText(QString::number(result));
}
```

Output:



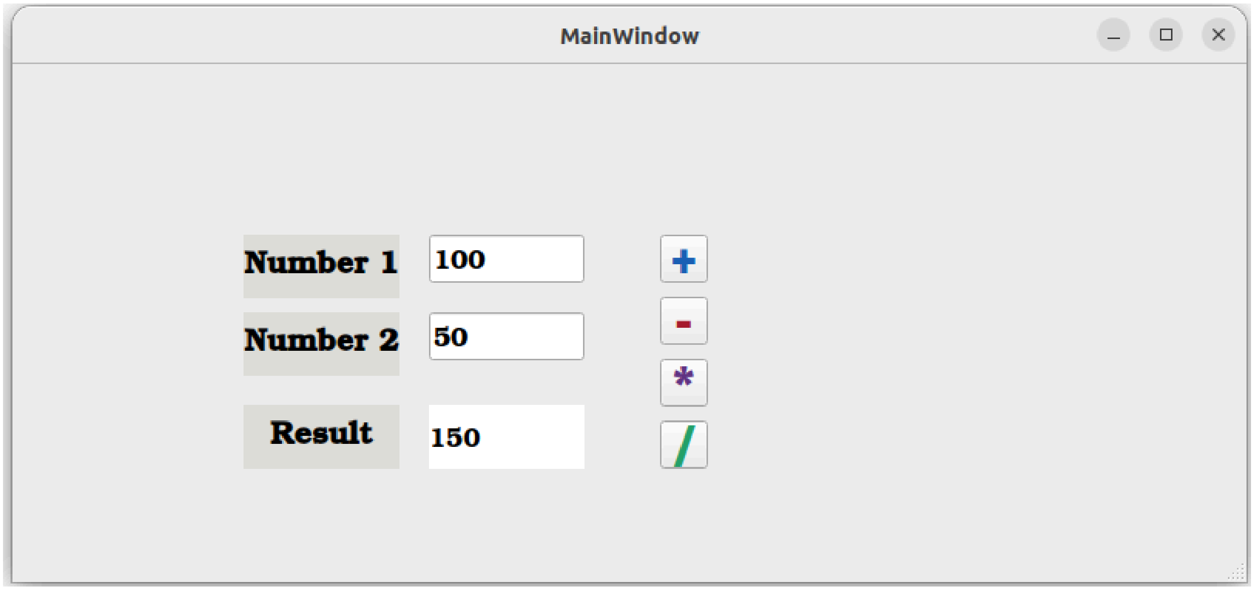
3.Create a simple calculator by collecting two numbers from users and perform (+,-,* and /) and show the results in a line edit.

Coding:

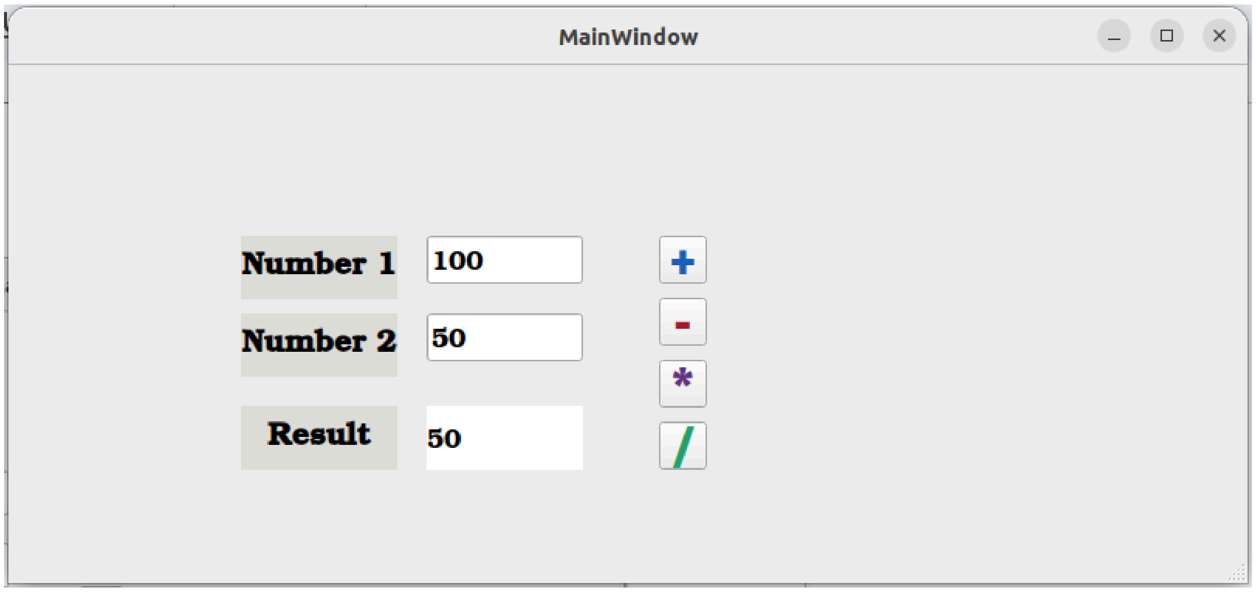
```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
float number1,number2,result;
MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}
MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        number1=arg1.toFloat();
}
void MainWindow::on_lineEdit_2_textChanged(const QString &arg1)
{
        number2=arg1.toFloat();
}
void MainWindow::on_pushButton_clicked()
{
        result=number1+number2;
        ui->label_4->setText(QString::number(result));
}
void MainWindow::on_pushButton_2_clicked()
{
        result=number1-number2;
        ui->label_4->setText(QString::number(result));
}
void MainWindow::on_pushButton_3_clicked()
{
        result=number1*number2;
        ui->label_4->setText(QString::number(result));
}
void MainWindow::on_pushButton_4_clicked()
{
        result=number1/number2;
        ui->label_4->setText(QString::number(result));
}
```

Output:

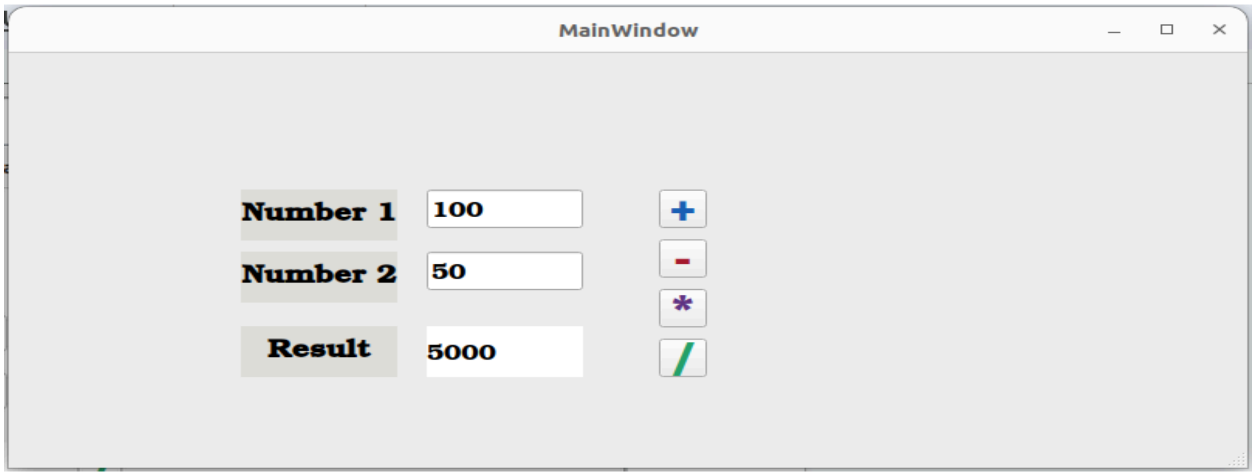
Addition:



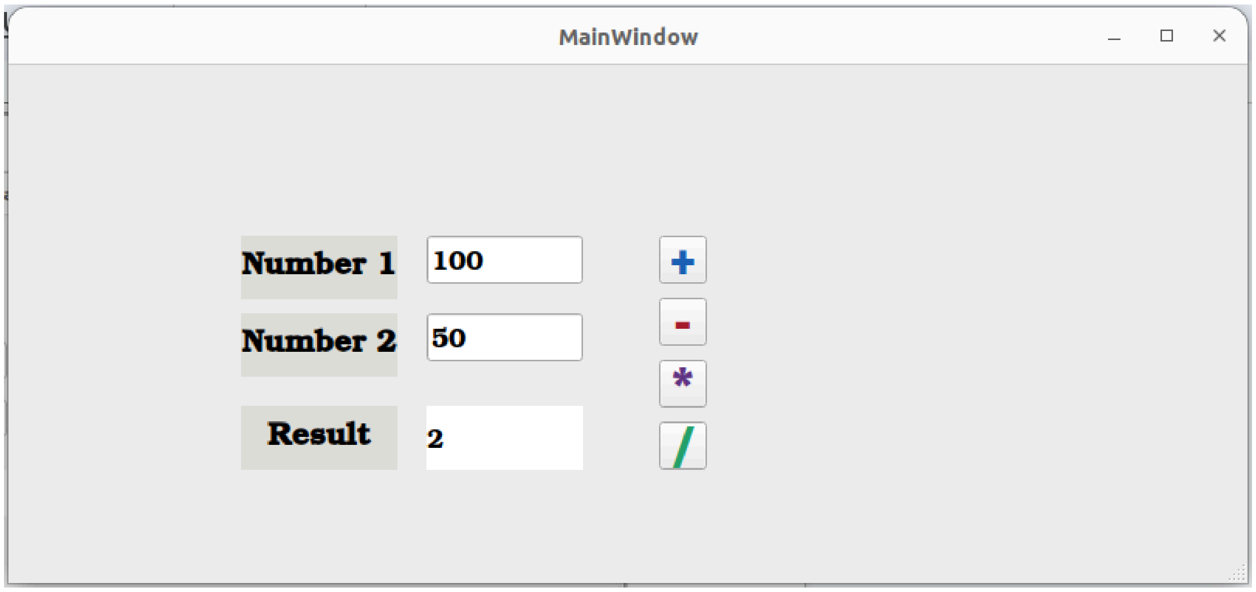
Subtraction:



Multiplication:



Division:



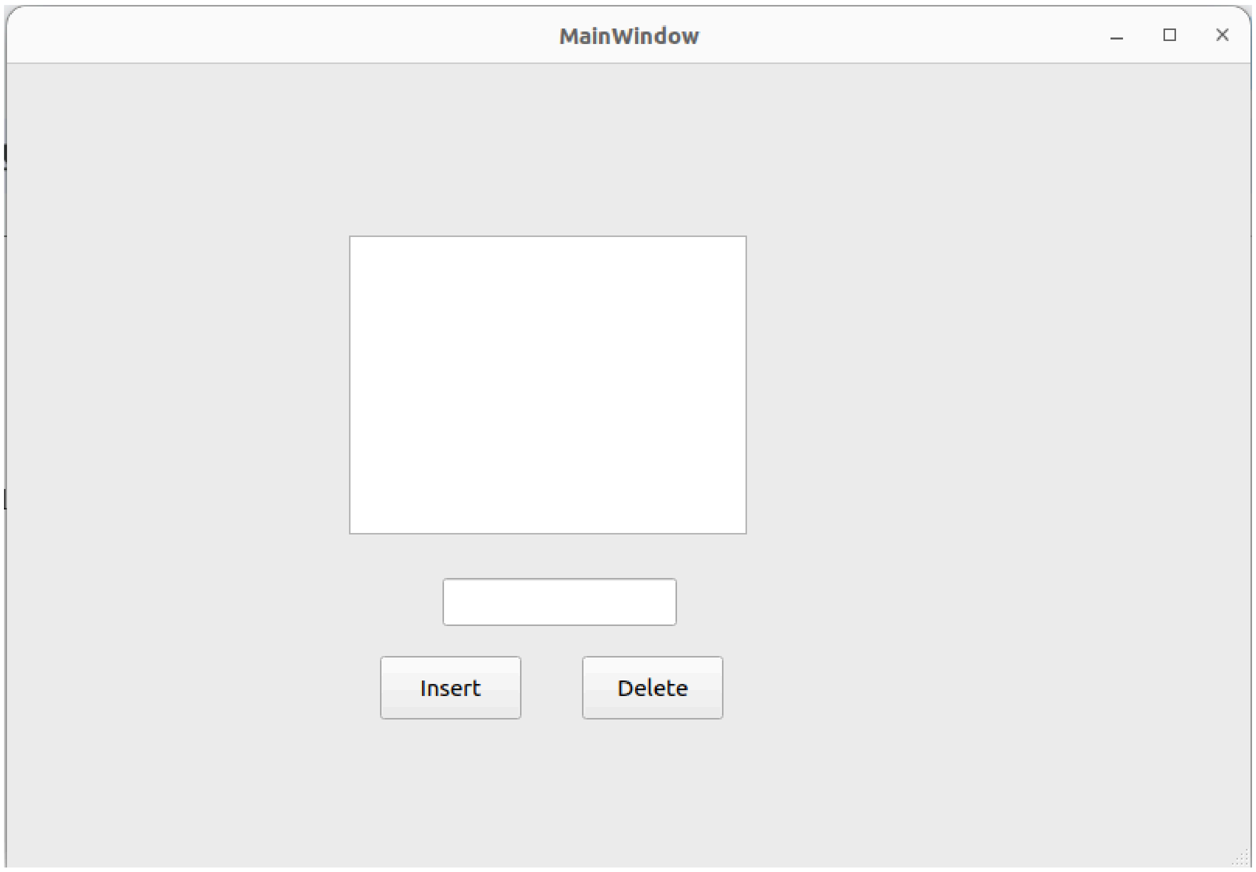
4. Create a List Widget for Insert, Delete Button.

Coding:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

QString number1;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}

MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        number1 = arg1;
}

void MainWindow::on_pushButton_clicked()
{
        ui->listWidget->addItem(number1);
}
```

```
void MainWindow::on_pushButton_2_clicked()
{
        qDeleteAll(ui->listWidget->selectedItems());
}
```

Output:



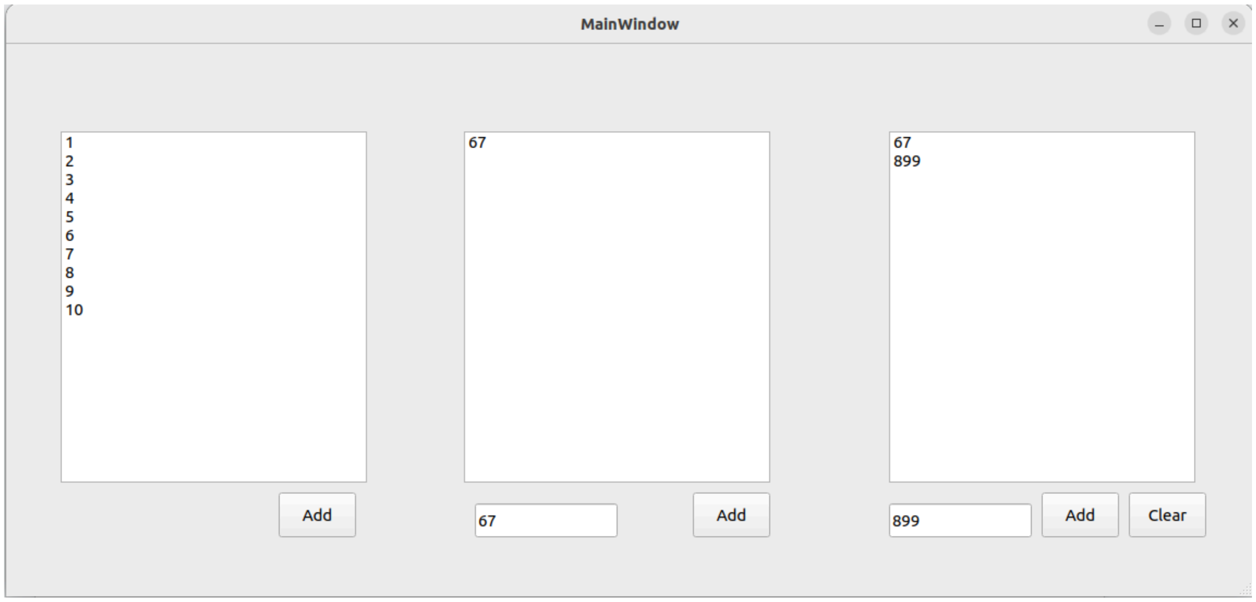
5.Create a three QT list widget .In the first list widget add the elements,second widget has a line edit and updates its value in the list widget.In the third one add or clear the values.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString speed,S;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}
MainWindow::~MainWindow()
{
        delete ui;
}
```

```
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        speed=arg1;
}
void MainWindow::on_pushButton_clicked()
{
        int str=ui->comboBox->currentIndex();
        QString str1=ui->comboBox->currentText();
        QString str2=ui->comboBox_2->currentText();
        QString str3=ui->comboBox_3->currentText();
        QString str4=ui->comboBox_4->currentText();
        if(str==0){
        QString S=str1+" "+str2+" "+str3+" "+speed;
        ui->listWidget->addItem(S);
        }
        else if(str==1){
        QString S=str1+" "+str2+" "+str3+" "+str4+" "+speed;
        ui->listWidget->addItem(S);
        }
        else if(str==2){
        QString S=str1+" "+str2+" "+str3+" "+str4+" "+speed;
        ui->listWidget->addItem(S);
        }
}
void MainWindow::on_pushButton_2_clicked()
{
        qDeleteAll(ui->listWidget->selectedItems());
}
void MainWindow::on_pushButton_3_clicked()
{
        ui->listWidget->clear();
}
```

Output:



```
1
2
3
4
5
6
7
8
9
10
```

```
67
```

```
67
899
```

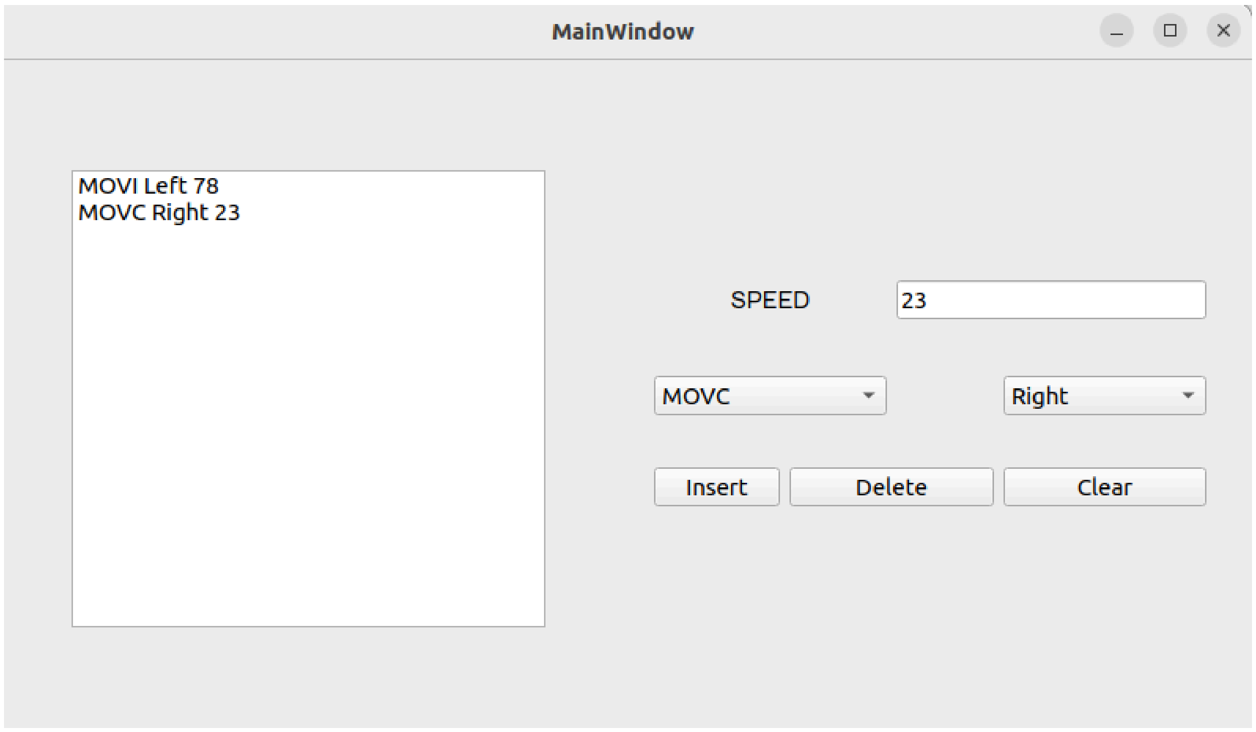
6.Create a combo box,list widget and three push buttons.While insert the values in the combo box it should be displayed in the list widget and also perform delete ,clear operations.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString speed,S;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}
MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        speed=arg1;
}
void MainWindow::on_pushButton_clicked()
{
        int str=ui->comboBox->currentIndex();
        QString str1=ui->comboBox->currentText();
        QString str2=ui->comboBox_2->currentText();
        QString str3=ui->comboBox_3->currentText();
        QString str4=ui->comboBox_4->currentText();
        if(str==0){
```

```
        QString S=str1+" "+str2+" "+str3+" "+speed;
        ui->listWidget->addItem(S);
        }
        else if(str==1){
        QString S=str1+" "+str2+" "+str3+" "+str4+" "+speed;
        ui->listWidget->addItem(S);
        }
        else if(str==2){
        QString S=str1+" "+str2+" "+str3+" "+str4+" "+speed;
        ui->listWidget->addItem(S);
        }
}
void MainWindow::on_pushButton_2_clicked()
{
        qDeleteAll(ui->listWidget->selectedItems());
}
void MainWindow::on_pushButton_3_clicked()
{
        ui->listWidget->clear();
}
```

Output:



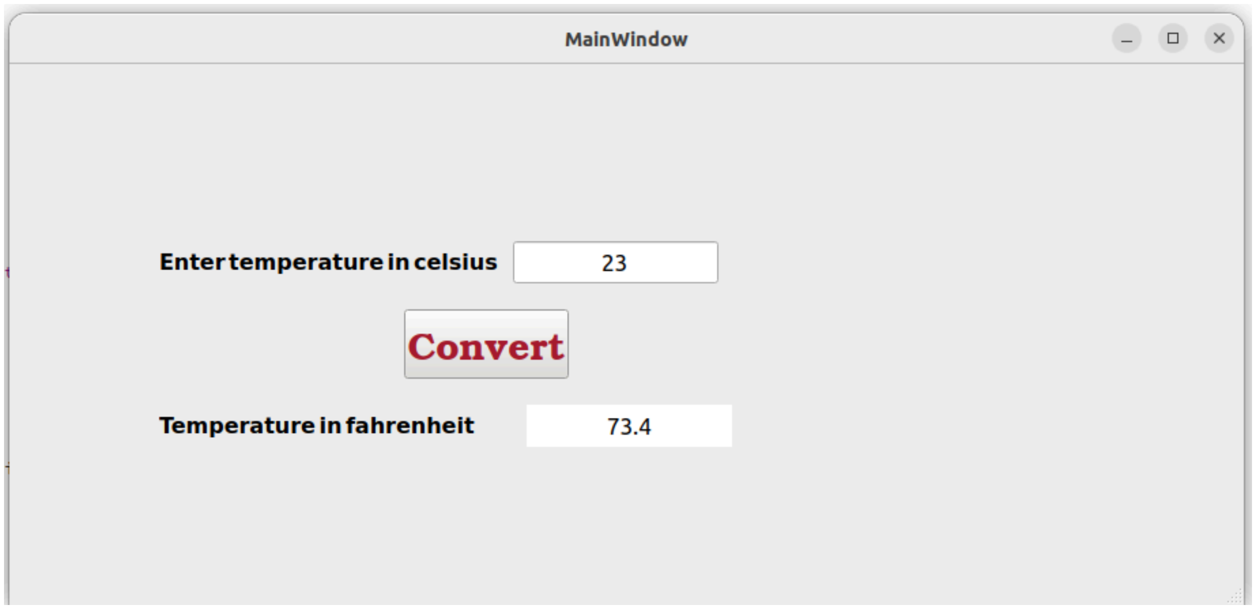
```
MOVI Left 78
MOVC Right 23
```

SPEED: 23

7.Create a Qt application that converts temperatures from Celsius to Fahrenheit.
Coding:
#include "mainwindow.h"

```
#include "./ui_mainwindow.h"
float celsius,fahrenheit;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}
MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        celsius=arg1.toFloat();
}
void MainWindow::on_pushButton_clicked()
{
        fahrenheit=(celsius*9/5)+32;
        ui->label_3->setText(QString::number(fahrenheit));
}
```

Output:

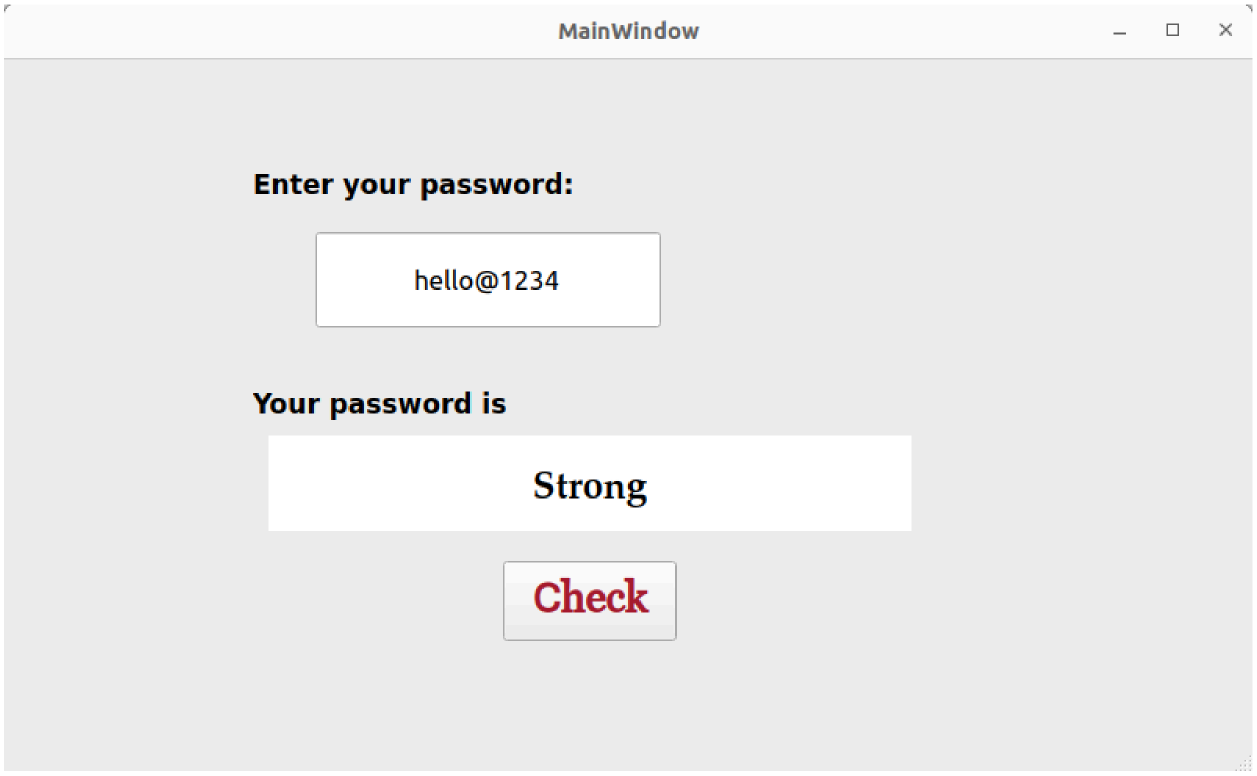


8. Design a password strength checker application. Include a QLabel to display a password input field using QLineEdit and a QPushButton to check the strength. Implement a function that analyzes the password's strength (e.g., length, complexity) and displays the result in the QLabel.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString password;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}

MainWindow::~MainWindow()
{
        delete ui;
}


void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        password = ui->lineEdit->text();
}


void MainWindow::on_pushButton_clicked()
{
        int length = password.length();
        QString resultMessage;

        if (length < 6) {
        resultMessage = "Weak (too short)";
        } else if (length < 10) {
        resultMessage = "Moderate";
        } else {
        resultMessage = "Strong";
        }


        ui->label_3->setText(resultMessage);
}
```

Output:



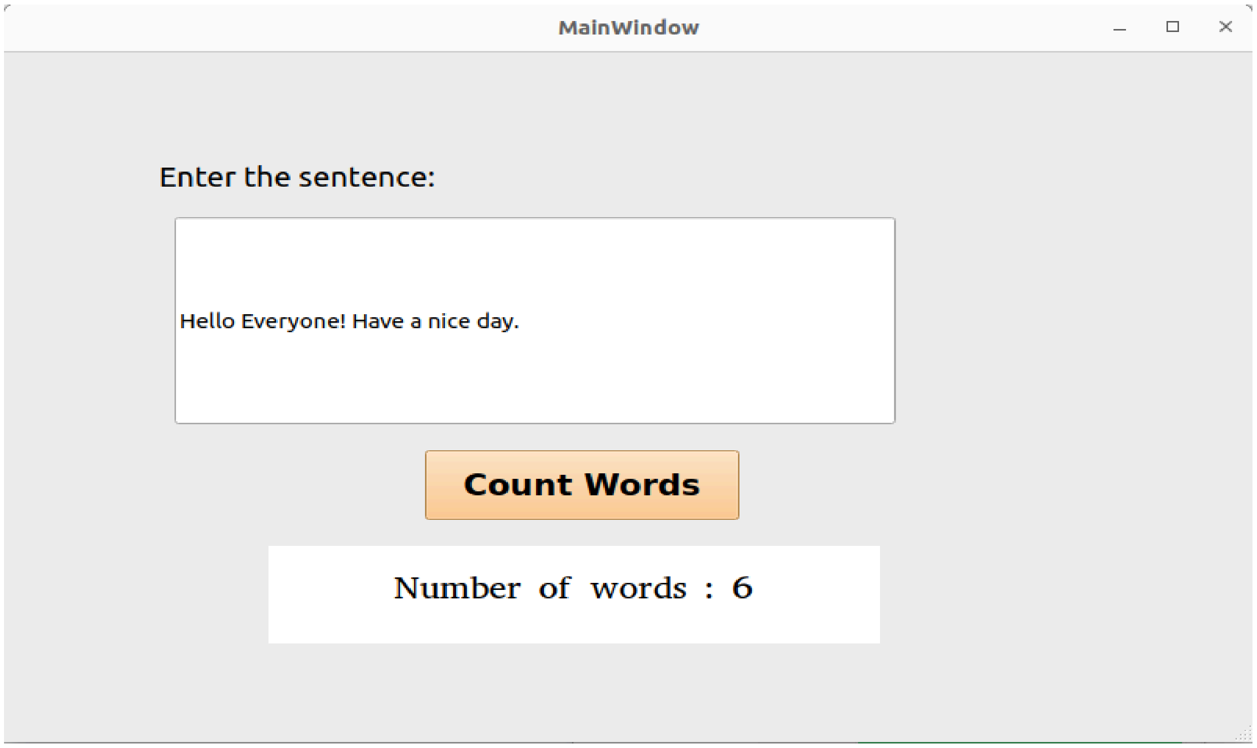
9. Create a word counter application with a QLabel to display a text input field using QLineEdit and a QPushButton to count words. When the user clicks the "Count Words" button, count and display the number of words in the input text in the QLabel.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString text;
QStringList words;
int count;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}

MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        text=ui->lineEdit->text();
}
```

```
void MainWindow::on_pushButton_clicked()
{
        words=text.split(" ",QString::SkipEmptyParts);
        count=words.size();
        ui->label_3->setText("Number of words : "+QString::number(count));
}
```

Output:



```
Hello Everyone! Have a nice day.
Number of words : 6
```

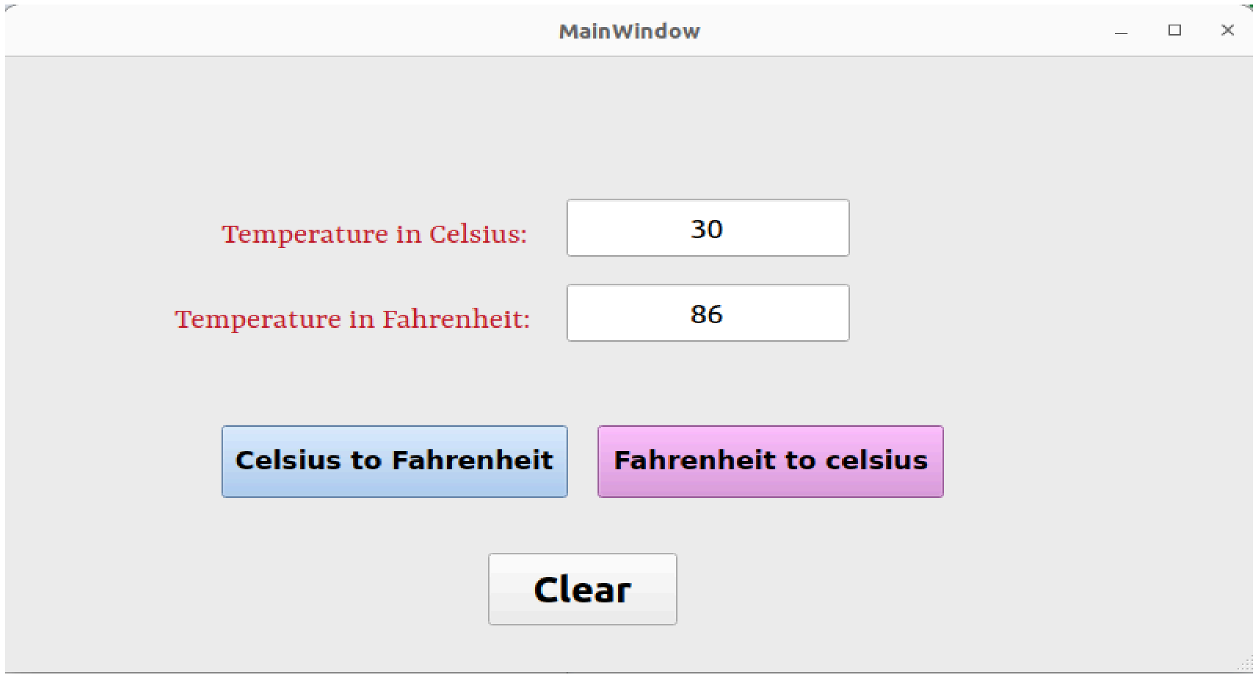
10.Develop a temperature converter application with QLabel components to display labels, QLineEdit for input, and QPushButton to perform the conversion. Implement the logic to convert temperatures between Celsius and Fahrenheit when the user clicks the appropriate button.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
float temp1,temp2,fahrenheit,celsius;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}
MainWindow::~MainWindow()
{
        delete ui;
}
```

```
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        temp1=arg1.toFloat();
}
void MainWindow::on_lineEdit_2_textChanged(const QString &arg1)
{
        temp2=arg1.toFloat();
}
void MainWindow::on_pushButton_clicked()
{
        fahrenheit=(temp1*9/5)+32;
        ui->lineEdit_2->setText(QString::number(fahrenheit));
}
void MainWindow::on_pushButton_2_clicked()
{
        celsius=(temp2-32)*5/9;
        ui->lineEdit->setText(QString::number(celsius));
}
void MainWindow::on_pushButton_3_clicked()
{
        ui->lineEdit->clear();
        ui->lineEdit_2->clear();
}
```

Output:



Temperature in Celsius: 30

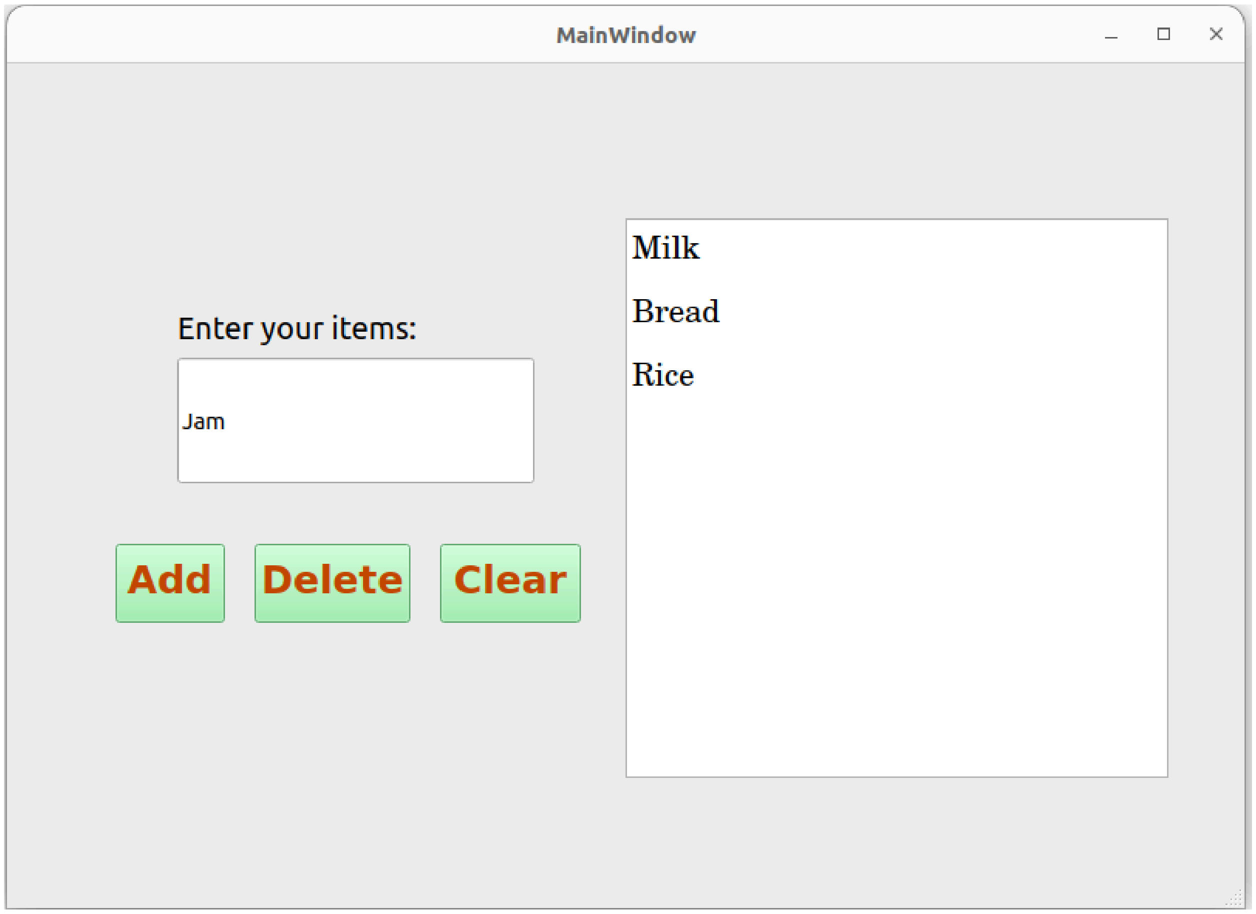
Temperature in Fahrenheit: 86

11.Design a grocery shopping list application using a QListWidget. Allow users to add grocery items (e.g., "Milk," "Bread") with a quantity field (QLineEdit) and an "Add" button. Include a "Delete" button to remove selected items from the list.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString items;

MainWindow::MainWindow(QWidget *parent)
	: QMainWindow(parent)
	, ui(new Ui::MainWindow)
{
	ui->setupUi(this);
}

MainWindow::~MainWindow()
{
	delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
	items=arg1;
}
void MainWindow::on_pushButton_clicked()
{
	ui->listWidget->addItem(items);
	ui->lineEdit->clear();
}
void MainWindow::on_pushButton_2_clicked()
{
	qDeleteAll(ui->listWidget->selectedItems());
}
void MainWindow::on_pushButton_3_clicked()
{
	ui->listWidget->clear();
}
```

Output:



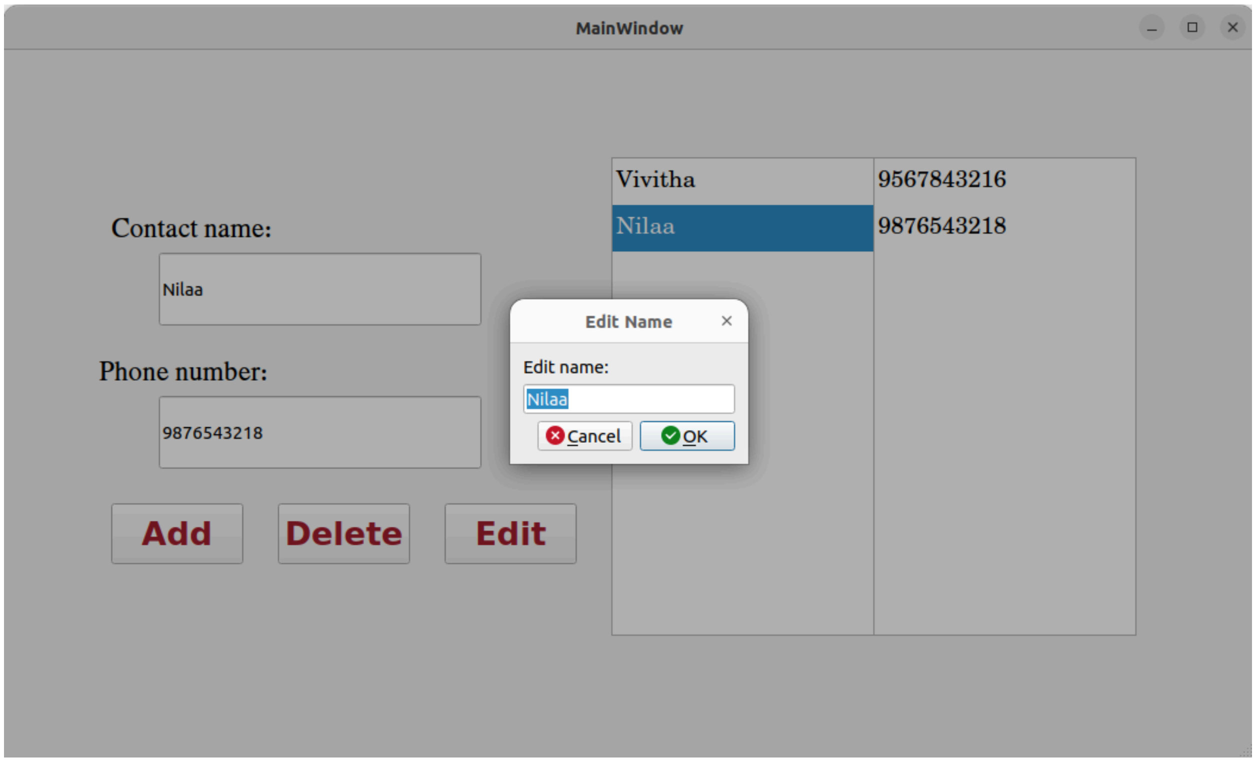
12..Build a contact list application using a QListWidget. Allow users to enter contact names and phone numbers. Use two QListWidgets side by side: one for displaying contact names and another for displaying phone numbers. Include buttons to add new contacts, delete selected contacts, and edit existing contacts.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString name,number;


MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}

MainWindow::~MainWindow()
{
        delete ui;
}
```

```
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        name=arg1;
}
void MainWindow::on_lineEdit_2_textChanged(const QString &arg1)
{
        number=arg1;
}
void MainWindow::on_pushButton_clicked()
{
        ui->listWidget->addItem(name);
        ui->listWidget_2->addItem(number);
}
void MainWindow::on_pushButton_2_clicked()
{
        qDeleteAll(ui->listWidget->selectedItems());
        qDeleteAll(ui->listWidget_2->selectedItems());
}
void MainWindow::on_pushButton_3_clicked()
{
        QListWidgetItem* selectedItem1 = ui->listWidget->currentItem();
        if (selectedItem1) {
        QString currentName = selectedItem1->text();
        QString newName = QInputDialog::getText(this, "Edit Name", "Edit name:",
QLineEdit::Normal, currentName);
        if (!newName.isEmpty() && newName != currentName) {
        selectedItem1->setText(newName);
        }
        }
        QListWidgetItem* selectedItem2 = ui->listWidget_2->currentItem();
        if (selectedItem2) {
        QString currentName = selectedItem2->text();
        QString newName = QInputDialog::getText(this, "Edit Name", "Edit name:",
QLineEdit::Normal, currentName);
        if (!newName.isEmpty() && newName != currentName) {
        selectedItem2->setText(newName);
        }
        }
}
```

Output:



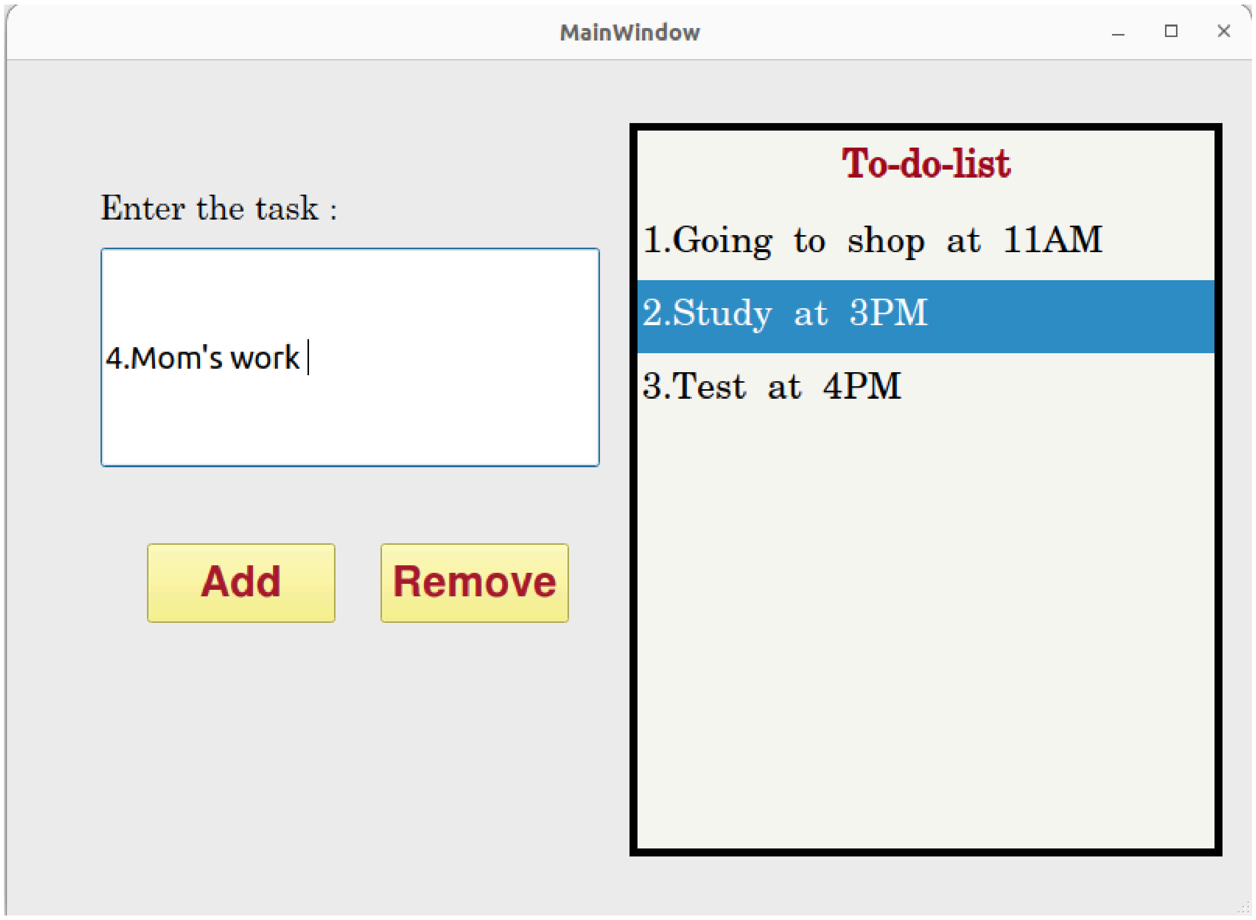
13.Create a basic to-do list application using a QListWidget. Design the UI with an input field (QLineEdit), an "Add" button (QPushButton), and a "Remove" button (QPushButton). When the user enters a task and clicks "Add," it should add the task to the QListWidget. When a task is selected and the "Remove" button is clicked, the selected task should be deleted.

Coding:

```
#include "mainwindow.h"
#include "./ui_mainwindow.h"
QString task;

MainWindow::MainWindow(QWidget *parent)
        : QMainWindow(parent)
        , ui(new Ui::MainWindow)
{
        ui->setupUi(this);
}

MainWindow::~MainWindow()
{
        delete ui;
}
void MainWindow::on_lineEdit_textChanged(const QString &arg1)
{
        task=arg1;
}
void MainWindow::on_pushButton_clicked()
```

```
{
        ui->listWidget->addItem(task);
        ui->lineEdit->clear();
}
void MainWindow::on_pushButton_2_clicked()
{
        qDeleteAll(ui->listWidget->selectedItems());
}
```

Output:



14.Design an interactive quiz application with Qt. Create a user-friendly interface that displays multiple-choice questions and options using QLabel and QRadioButton. Allow users to select their answers, and display the correct answer after submission using QMessageBox. Implement a scoring system to keep track of users' scores.

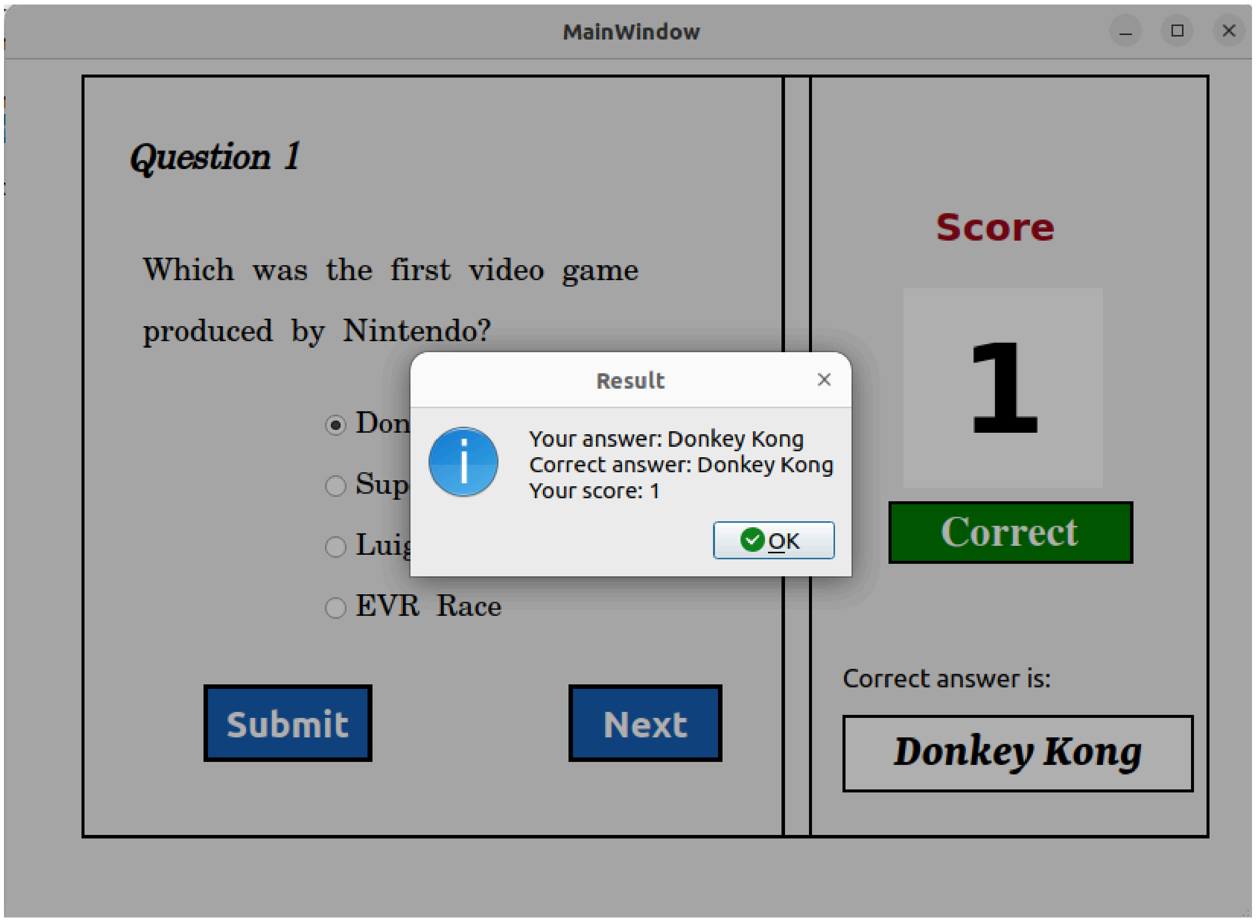
Coding:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>

MainWindow::MainWindow(QWidget *parent)
```

```
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    score=0;
    connect(ui->pushButton, &QPushButton::clicked, this,
&MainWindow::on_pushButton_2_clicked);
    QString question = "Which was the first video game produced by Nintendo?";
    QString answer1 = "Super Mario Bros";
    QString answer2 = "Luigi’s Mansion";
    QString answer3 = "EVR Race";
    QString answer4 = "Donkey Kong";
    QString correctAnswer = "Donkey Kong";

    ui->label_3->setText(question);
    ui->radioButton->setText(answer1);
    ui->radioButton_2->setText(answer2);
    ui->radioButton_3->setText(answer3);
    ui->radioButton_4->setText(answer4);
    ui->lineEdit_3->setText(correctAnswer);
    ui->lineEdit_3->setVisible(false);

}

MainWindow::~MainWindow()
{
    delete ui;
}
void MainWindow::on_pushButton_2_clicked()
{
    QString selectedAnswer;
    if (ui->radioButton->isChecked()) {
    selectedAnswer = ui->radioButton->text();
    } else if (ui->radioButton_2->isChecked()) {
    selectedAnswer = ui->radioButton_2->text();
    } else if (ui->radioButton_3->isChecked()) {
    selectedAnswer = ui->radioButton_3->text();
    } else if (ui->radioButton_4->isChecked()) {
    selectedAnswer = ui->radioButton_4->text();
    }

    QString correctAnswer = ui->lineEdit_3->text();
    if (selectedAnswer == correctAnswer) {
    score++;
```

```
        ui->lineEdit_2->setText("Correct");
        ui->lineEdit_2->setStyleSheet("background-color:green;color:white;");
        ui->lineEdit->setText(QString::number(score));
        } else {
        ui->lineEdit_2->setText("Wrong");
        ui->lineEdit_2->setStyleSheet("background-color:red;color:white;");
        ui->lineEdit->setText(QString::number(score));
        }

        ui->lineEdit_3->setText(correctAnswer);
        ui->lineEdit_3->setVisible(true);
        QMessageBox::information(this, "Result", "Your answer: " + selectedAnswer +
"\nCorrect answer: " + correctAnswer + "\nYour score: " + QString::number(score));
        ui->radioButton->setChecked(false);
        ui->radioButton_2->setChecked(false);
        ui->radioButton_3->setChecked(false);
        ui->radioButton_4->setChecked(false);
}
```

Output:



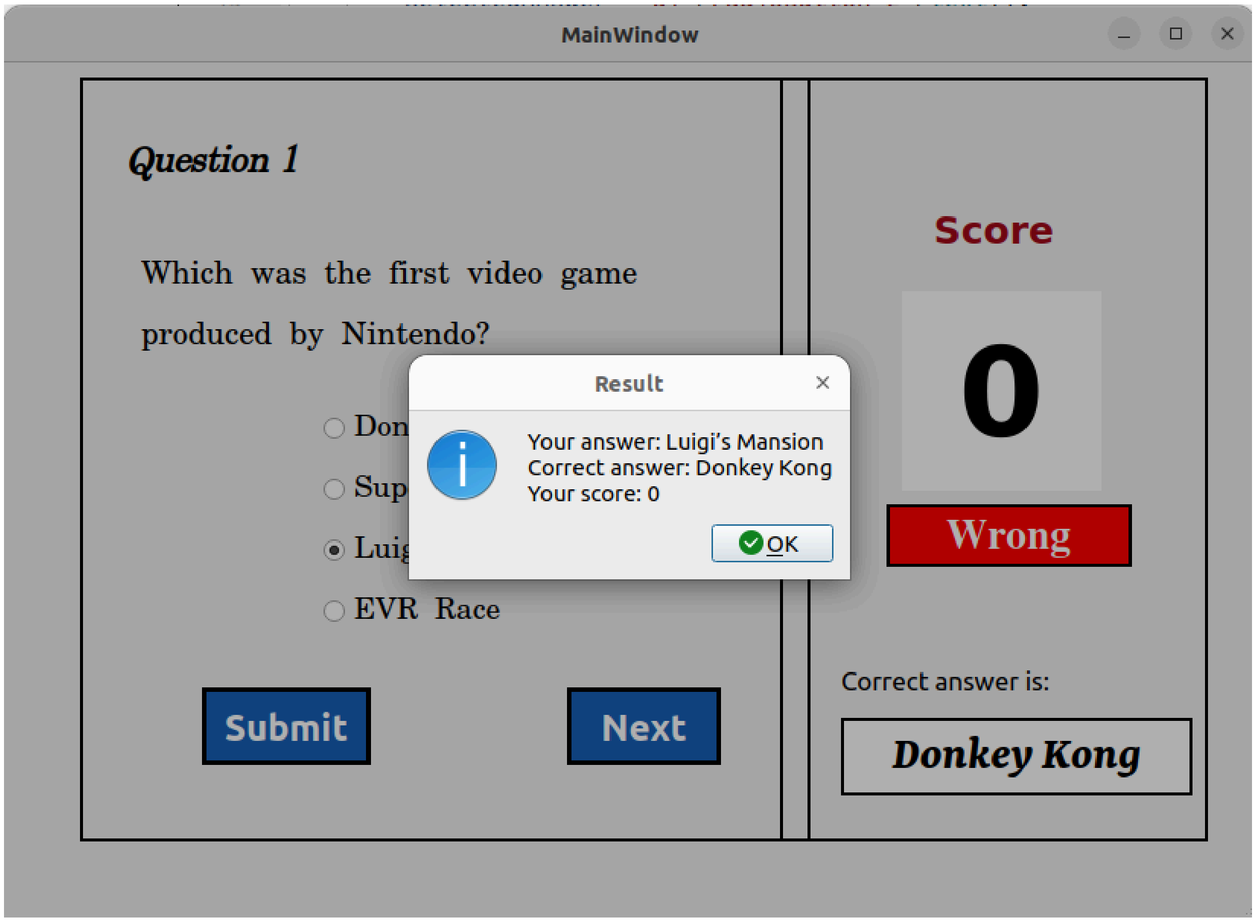
Which was the first video game produced by Nintendo?

- [✓ Don]
- Sup
- Luig
- EVR Race

```
Your answer: Donkey Kong
Correct answer: Donkey Kong
Your score: 1
```

Correct answer is:

Donkey Kong



Question 1

Which was the first video game produced by Nintendo?

- ○ Don
- ○ Sup
- [✓ Luig]
- ○ EVR Race

Your answer: Luigi’s Mansion
Correct answer: Donkey Kong
Your score: 0

Score

0

Wrong

Correct answer is:

Donkey Kong