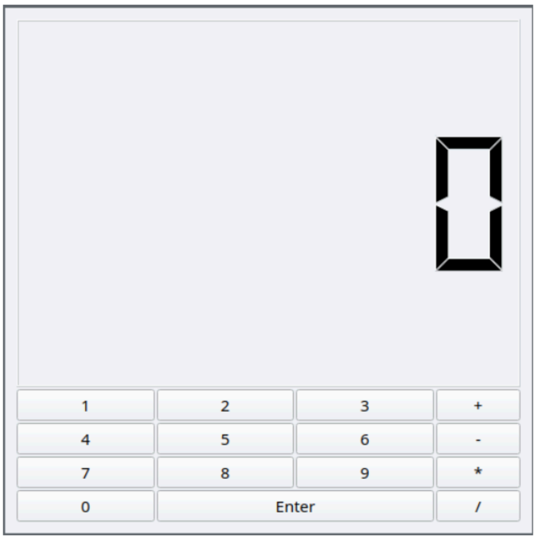
Mohamed Abdallaoui
Tarek Kamel
Instructor’s Name
Anass Belcaid
10 November 2021

Calculator



Preview of our basic Calculator

Table of contents :

1.Introduction

The goal is to use Signals and Slots to simulate a basic calculator behavior. The supported operations are `∗, +, −, /`. Our role is to create the connections between the different button to add, subtract, divide and multiply the numbers at the input of the user

2. Code Explanation

We first started buy adding the two following slots as mentioned in the Homework

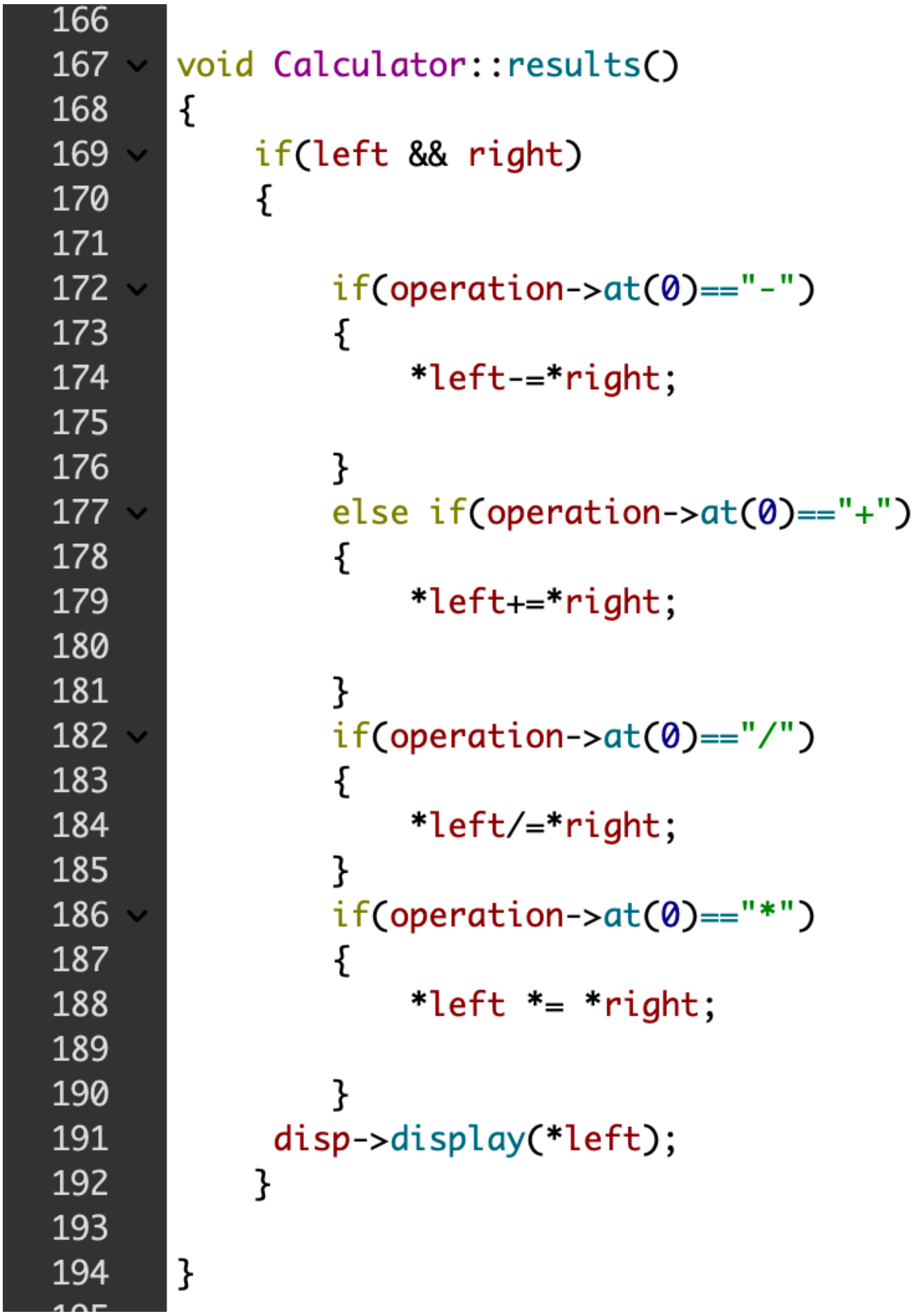
```
public slots:
    void changeOperation();  //Slot to handle the
click on operations
    void newDigit();
```

Next we implemented them in the cpp file

We added a new method results() that implements the computing function of (addition ,subs-traction ,division and multiplication) on each corresponding button

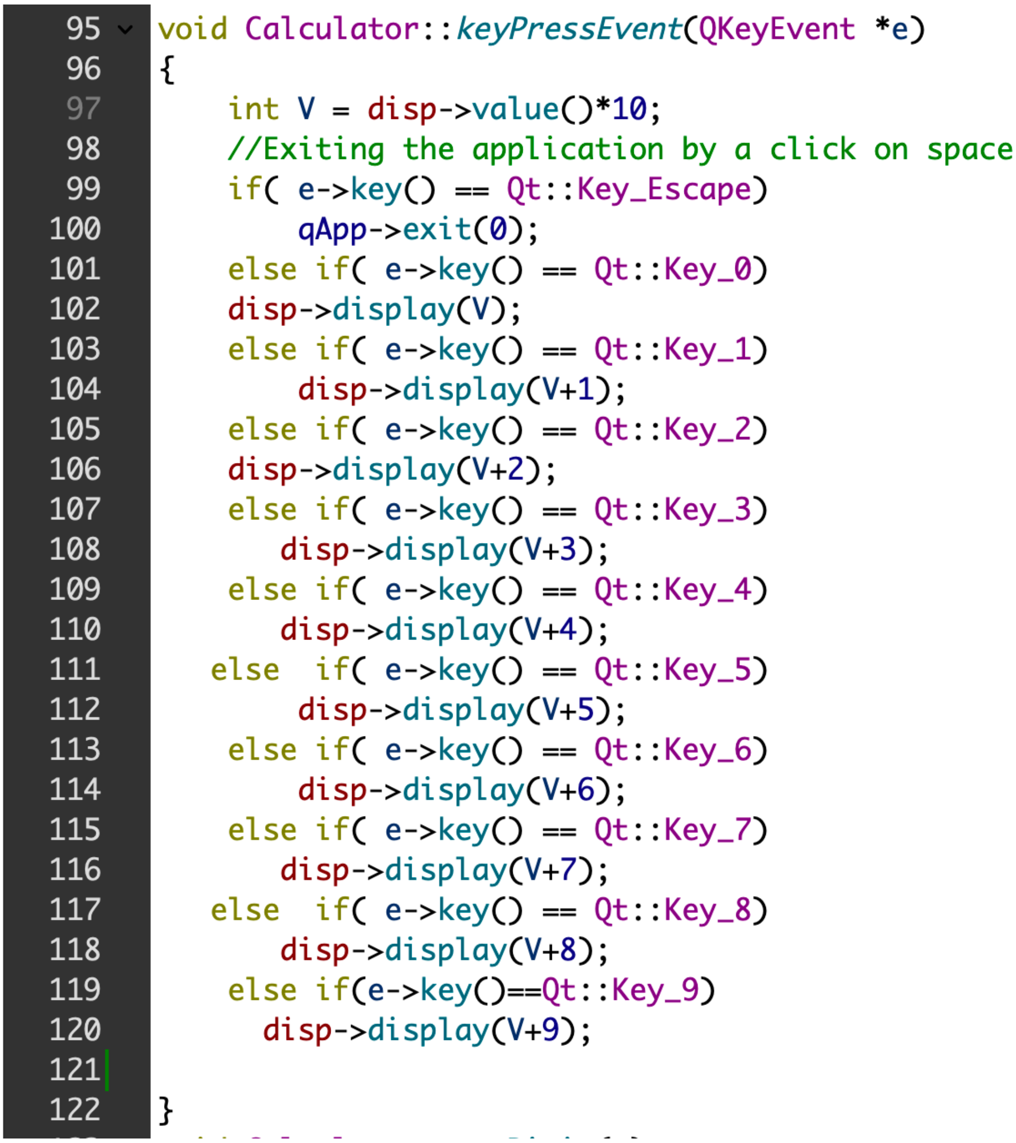
When the user enters the left and right value, this method checks if the two values are set to true then executes the operations.

Result below



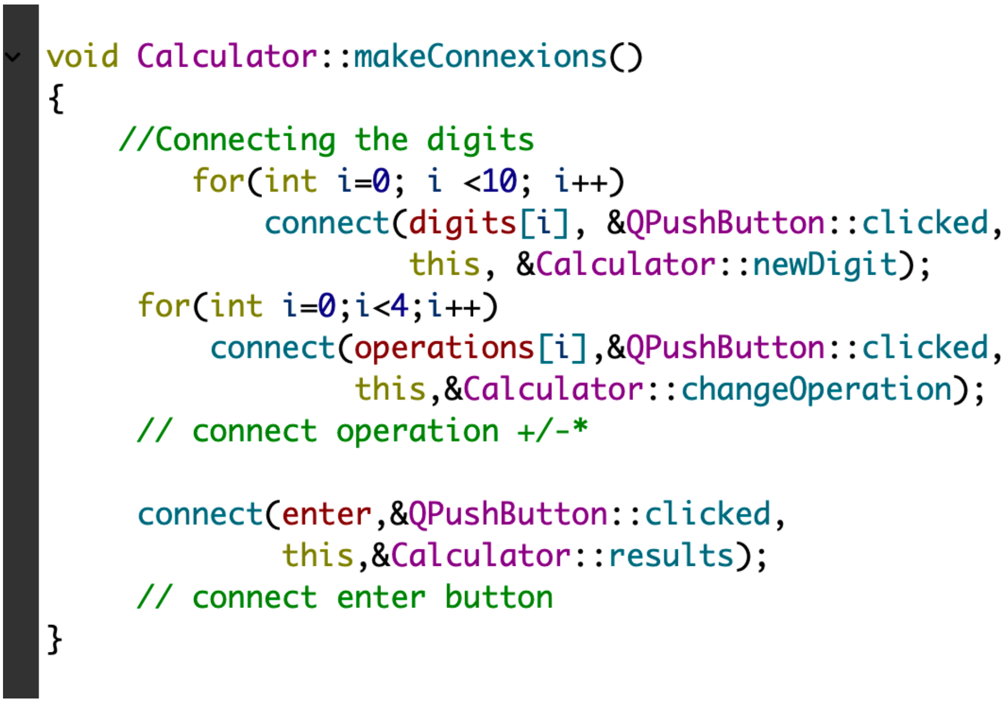
```cpp
void Calculator::results()
{
    if(left && right)
    {

        if(operation->at(0)=="-")
        {
            *left-=*right;

        }
        else if(operation->at(0)=="+")
        {
            *left+=*right;

        }
        if(operation->at(0)=="/")
        {
            *left/=*right;
        }
        if(operation->at(0)=="*")
        {
            *left *= *right;

        }
     disp->display(*left);
    }

}
```

We also worked on new events using just the key board as follows :



```
void Calculator::keyPressEvent(QKeyEvent *e)
{
    int V = disp->value()*10;
    //Exiting the application by a click on space
    if( e->key() == Qt::Key_Escape)
        qApp->exit(0);
    else if( e->key() == Qt::Key_0)
    disp->display(V);
    else if( e->key() == Qt::Key_1)
        disp->display(V+1);
    else if( e->key() == Qt::Key_2)
    disp->display(V+2);
    else if( e->key() == Qt::Key_3)
       disp->display(V+3);
    else if( e->key() == Qt::Key_4)
       disp->display(V+4);
   else  if( e->key() == Qt::Key_5)
        disp->display(V+5);
    else if( e->key() == Qt::Key_6)
        disp->display(V+6);
    else if( e->key() == Qt::Key_7)
       disp->display(V+7);
   else  if( e->key() == Qt::Key_8)
       disp->display(V+8);
    else if(e->key()==Qt::Key_9)
      disp->display(V+9);

}
```

Moving on to the connections as we added two more connections one for the Enter Button and another on for the operations Buttons

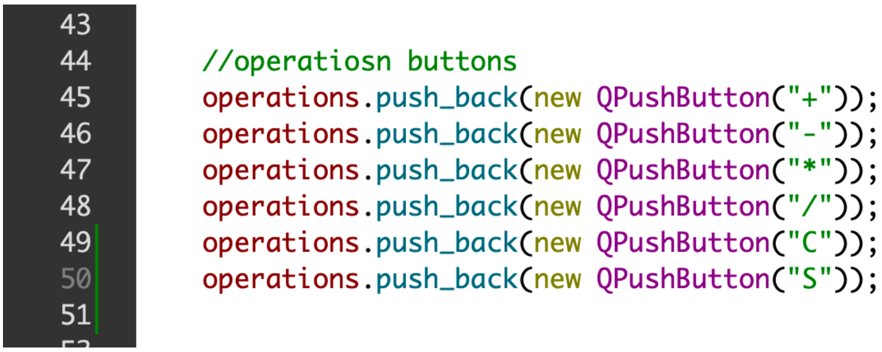


```
void Calculator::makeConnexions()
{
    //Connecting the digits
        for(int i=0; i <10; i++)
            connect(digits[i], &QPushButton::clicked,
                    this, &Calculator::newDigit);
     for(int i=0;i<4;i++)
         connect(operations[i],&QPushButton::clicked,
                 this,&Calculator::changeOperation);
     // connect operation +/-*

     connect(enter,&QPushButton::clicked,
             this,&Calculator::results);
     // connect enter button
}
```

Enhancement

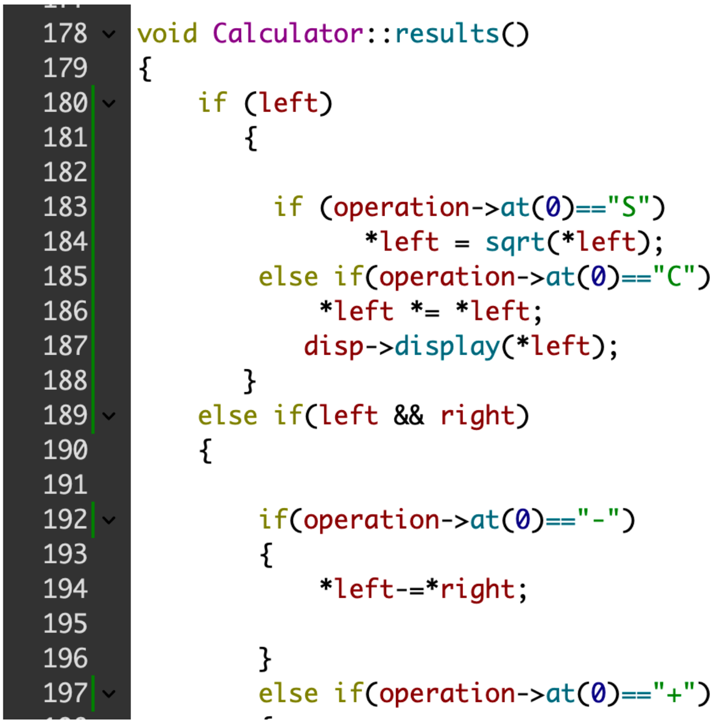
To add more functionalities to our Calculator we added Three more Buttons :

- Cancel button : that resets all the value to NULL
- S Button : Which returns the square root of a number
- C Button : Which returns the Square of a given number



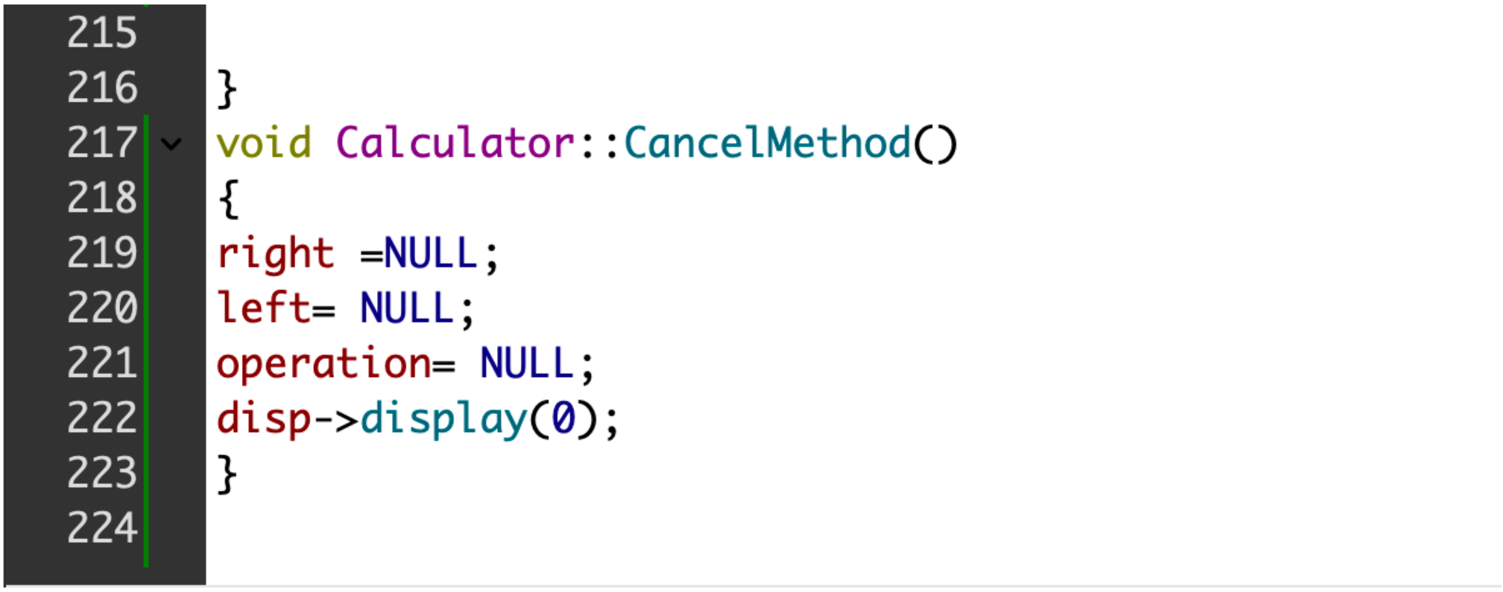
```
//operatiosn buttons
operations.push_back(new QPushButton("+"));
operations.push_back(new QPushButton("-"));
operations.push_back(new QPushButton("*"));
operations.push_back(new QPushButton("/"));
operations.push_back(new QPushButton("C"));
operations.push_back(new QPushButton("S"));
```

Above we created two new Pushbuttons (S and C) to the operations vector



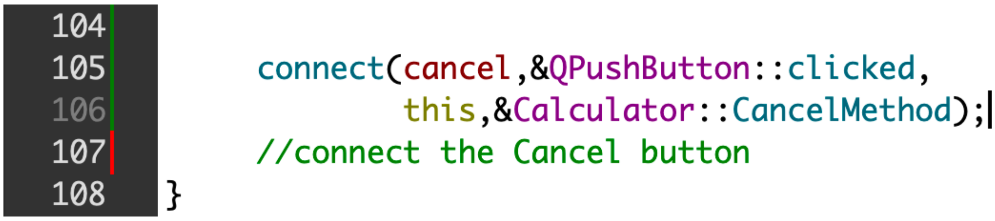
```
void Calculator::results()
{
    if (left)
       {

         if (operation->at(0)=="S")
               *left = sqrt(*left);
        else if(operation->at(0)=="C")
            *left *= *left;
           disp->display(*left);
       }
    else if(left && right)
    {

        if(operation->at(0)=="-")
        {
            *left-=*right;

        }
        else if(operation->at(0)=="+")
```

Now we have implemented a new condition to the results() method that only checks if left is full then computes two new operations notably Square-root and the square of a number



```

}
void Calculator::CancelMethod()
{
right =NULL;
left= NULL;
operation= NULL;
disp->display(0);
}

```

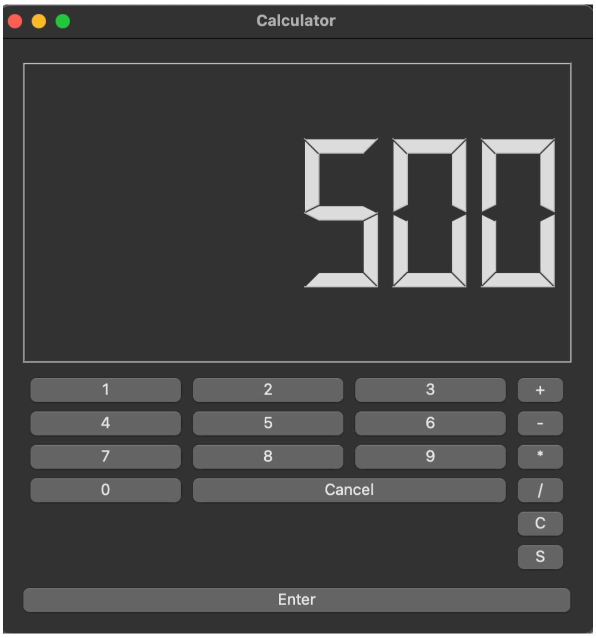
We have created a new slot and connected it to the QPushButton “Cancel" when the signal is clicked



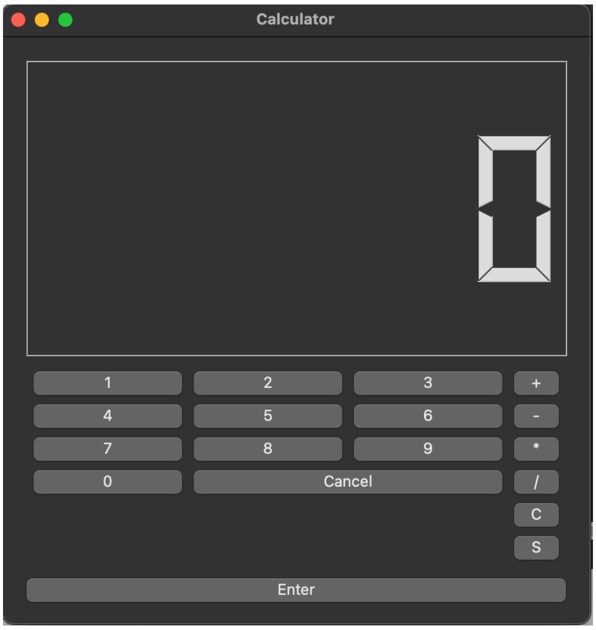
```

    connect(cancel,&QPushButton::clicked,
            this,&Calculator::CancelMethod);
    //connect the Cancel button
}
```

Final Output

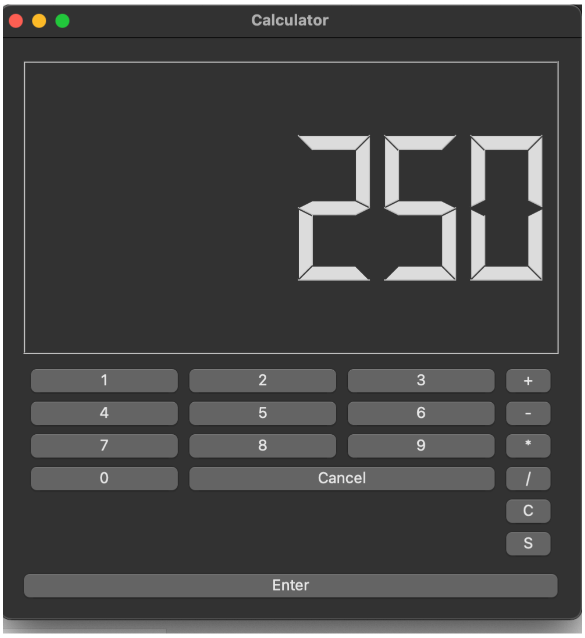
This is our results after computing the following operation :
250 - 500 = -250



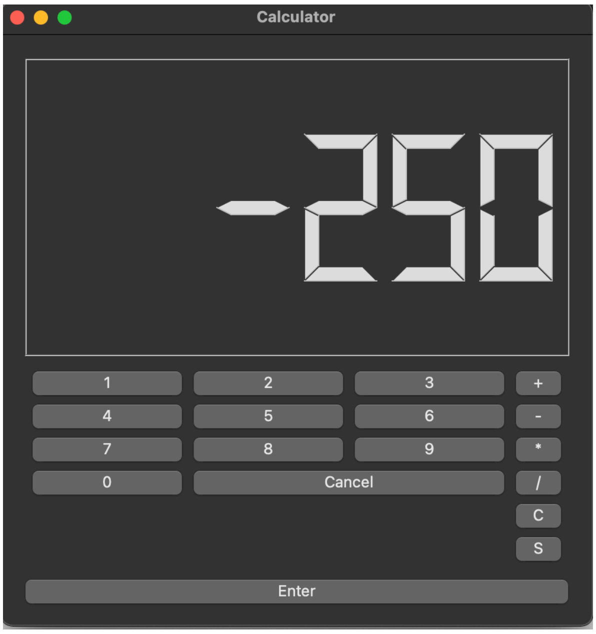
500



0

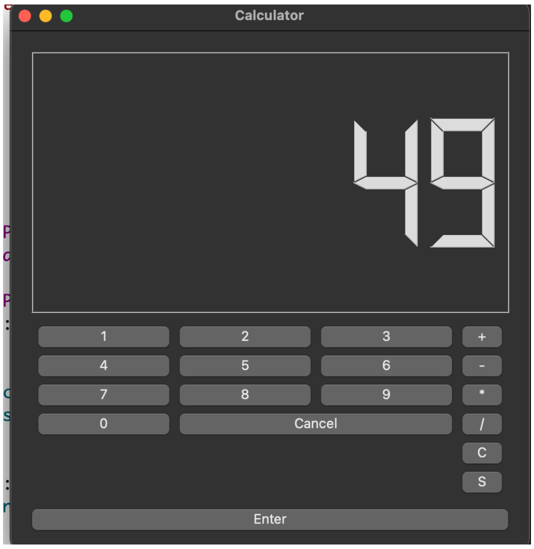


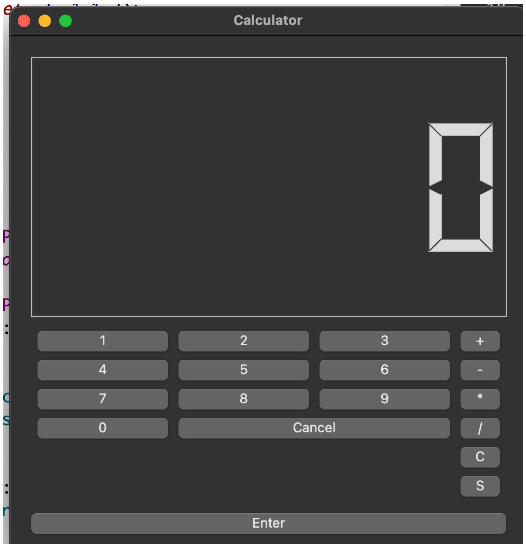
250

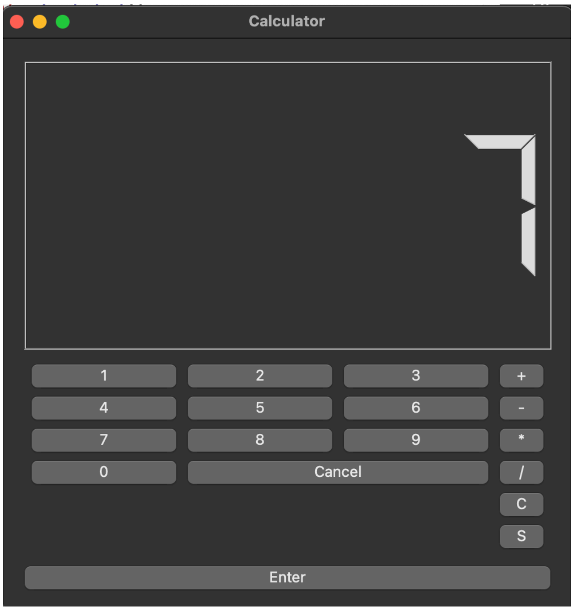


-250

This is our results after computing the following operation :
SQRT(49) = 7


Calculator
49
1
2
3
+
4
5
6
-
7
8
9
*
0
Cancel
/
C
S
Enter


Calculator
0
1
2
3
+
4
5
6
-
7
8
9
*
0
Cancel
/
C
S
Enter


Calculator
7
1
2
3
+
4
5
6
-
7
8
9
*
0
Cancel
/
C
S
Enter