KENDRIYA VIDYALAYA RAILWAY GANDHIDHAM

Project Work In
Computer Science (Code No -083)
2020 – 2021

Submitted By:
Name:- Vishal Meena
Roll No:- 34
Class:- 11th

CERTIFICATE

This is to certify that the **COMPUTER SCIENCE** Project work has been successfully completed by Master **Vishal Meena** Roll No. **34** of class **11th** during the academic session **2020 – 2021**.

Examiner No:

Date:

Teacher In-Charge External Examiner

Principal

ACKNOWLEDGEMENT

It is profound privilege to express my deep sense of gratitude towards our school **Kendriya Vidyalaya Railway Gandhidham**.

I would also like to thank our respected Principal **Mr. Chandan Singh Pilkhwal** for their encouragement and appreciation.

I wish to extend my gratitude in all sincerity to our teacher **Mr.Vinay Kumar Chauhan** who gave me an opportunity to work under his guidance. He has been a constant source of inspiration and without his valuable guidance the project would not have been successful. I would also like to thank our other teachers and staff members who provided me help and support regarding this project. At last, I am also thankful to all my classmates who helped me with wholeheartedness during the development of the project.

Introduction to Python

Python is an interpreted, high level and general purpose programming language. It is Dynamically typed. Python was created in the late 1980s, and first released in 1991, by Guido Van Rossum as a successor to the ABC programming language.
As of January 2021, Python ranks third in TIOBE's index of most popular programming languages, behind C and Java having previously gained second place and their award for the most popularity gain for 2020.
Python was conceived in the late 1980s by Guido Van Rossun at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.
Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.
Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture, for example, the metasyntactic variables often used in Python literature are *spam* and *eggs* instead of the traditional *foo* and *bar*. The official Python documentation also contains various references to Monty Python routines.

Python 2.0 was released on 16 October 2000 with many major new features.
Python 3.0 was released on 3 December 2008.

Modules used

Tkinter module :-

The *tkinter* package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and *tkinter* are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)

To make a widget in Tk, the command is always of the form:

```
>>> classCommand newPathname options
```

- *classCommand*

  Denotes which kind of widget to make (a button, a label, a menu…)

- *newPathname*

  Is the new name for this widget. All names in Tk must be unique. To help enforce this, widgets in Tk are named with *pathnames*, just like files in a file system. The top level widget, the *root*, is called `.` (period) and children are delimited by more periods. For example, `.myApp.controlPanel.okButton` might be the name of a widget

- *options*

  configure the widget’s appearance and in some cases, its behavior. The options come in the form of a list of flags and values. Flags are preceded by a ‘-‘, like Unix shell command flags, and values are put in quotes if they are more than one word.

PROJECT ON:-

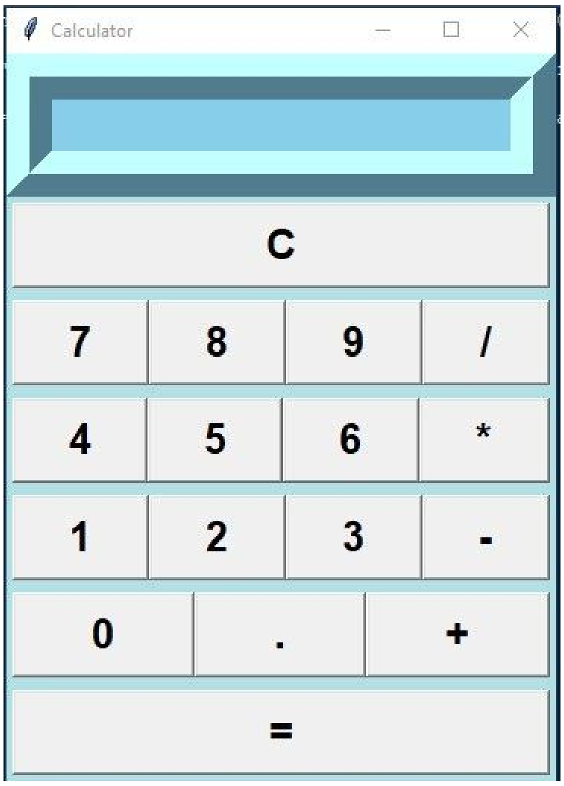
G.U.I.
CALCULATOR

SOURCE CODE

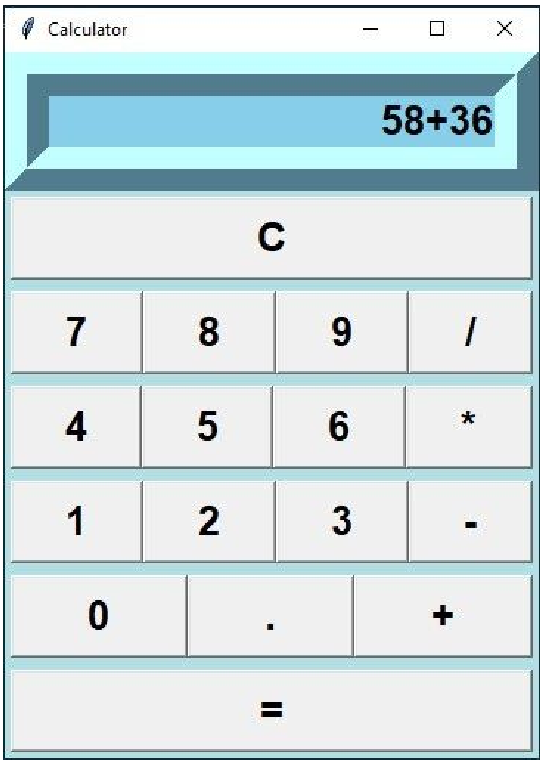
```
# Importing Tkinter module
from tkinter import *

# Creating the main interface / window for calculator
def iCalc(source, side):
    storeObj = Frame(source, borderwidth=4, bd=4, bg="powder blue")
    storeObj.pack(side=side, expand =YES, fill =BOTH)
    return storeObj

def button(source, side, text, command=None):
    storeObj = Button(source, text=text, command=command)
    storeObj.pack(side=side, expand = YES, fill=BOTH)
    return storeObj

class app(Frame):
    def __init__(self):
        Frame.__init__(self)
        self.option_add('*Font', 'arial 20 bold')
        self.pack(expand = YES, fill =BOTH)
        self.master.title('Calculator')

# Adding Display Widget
        display = StringVar()
        Entry(self, relief=RIDGE, textvariable=display,
          justify='right'
          , bd=30, bg="sky blue").pack(side=TOP,
                                        expand=YES, fill=BOTH)

# Adding Clear Button Widget
        for clearButton in (["C"]):
            erase = iCalc(self, TOP)
            for ichar in clearButton:
                button(erase, LEFT, ichar, lambda
                    storeObj=display, q=ichar: storeObj.set(''))

# Adding Numbers And Symbols Widget
        for numButton in ("789/", "456*", "123-", "0.+"):
         FunctionNum = iCalc(self, TOP)
         for iEquals in numButton:
            button(FunctionNum, LEFT, iEquals, lambda
                storeObj=display, q=iEquals: storeObj
                   .set(storeObj.get() + q))

# Adding Equal Button
        EqualButton = iCalc(self, TOP)
        for iEquals in "=":
            if iEquals == '=':
                btniEquals = button(EqualButton, LEFT, iEquals)
                btniEquals.bind('<ButtonRelease-1>', lambda e,s=self,
                                storeObj=display: s.calc(storeObj), '+')



            else:
                btniEquals = button(EqualButton, LEFT, iEquals,
                                    lambda storeObj=display, s=' %s ' % iEquals: storeObj.set
                                    (storeObj.get() + s))

# Applying Event Trigger On Widgets
    def calc(self, display):
            try:
                display.set(eval(display.get()))
            except:
                display.set("ERROR")


if __name__=='__main__':
 app().mainloop()
```

OUTPUT

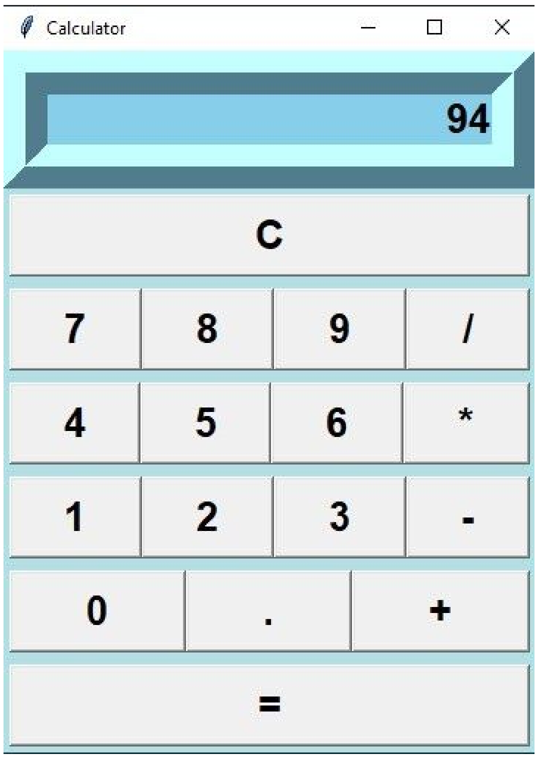
1. Main Window



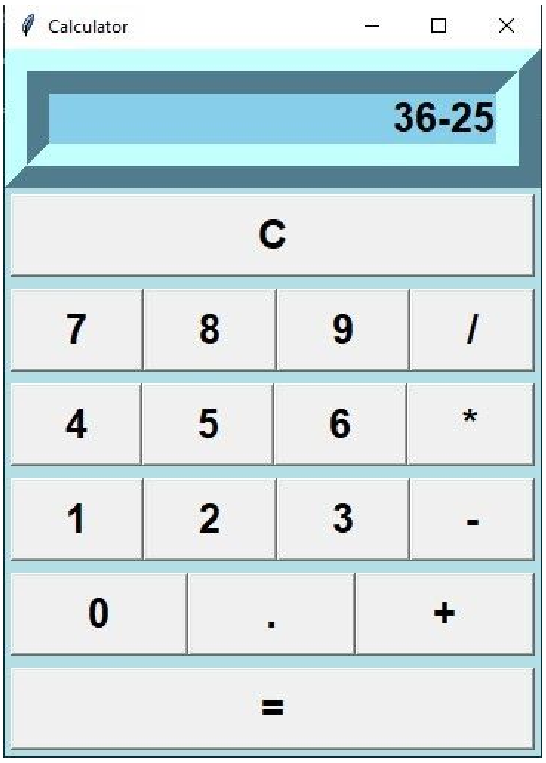
2. Addition



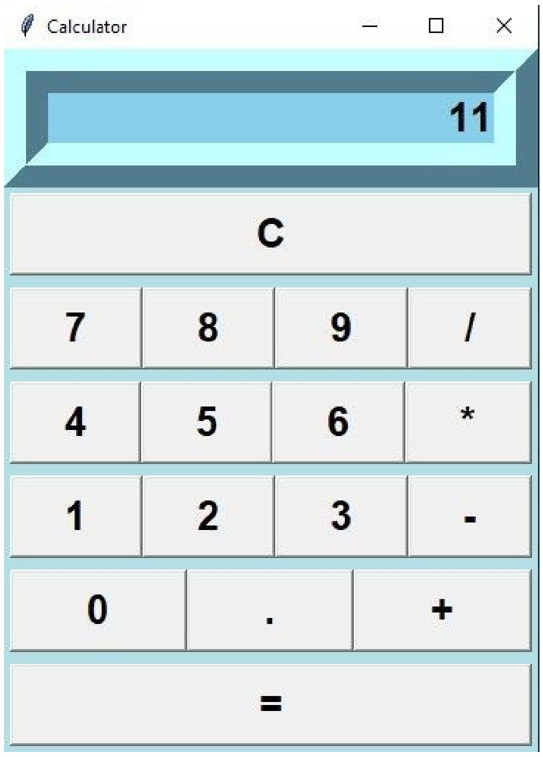
Result



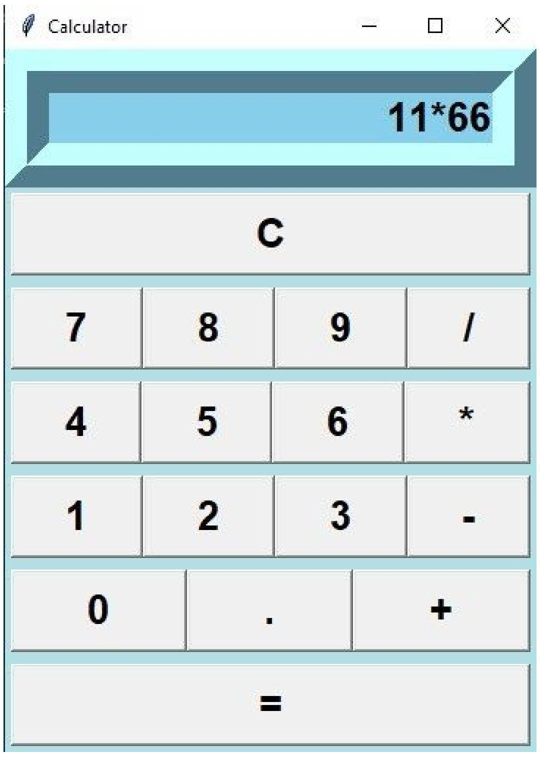
3. Subtraction



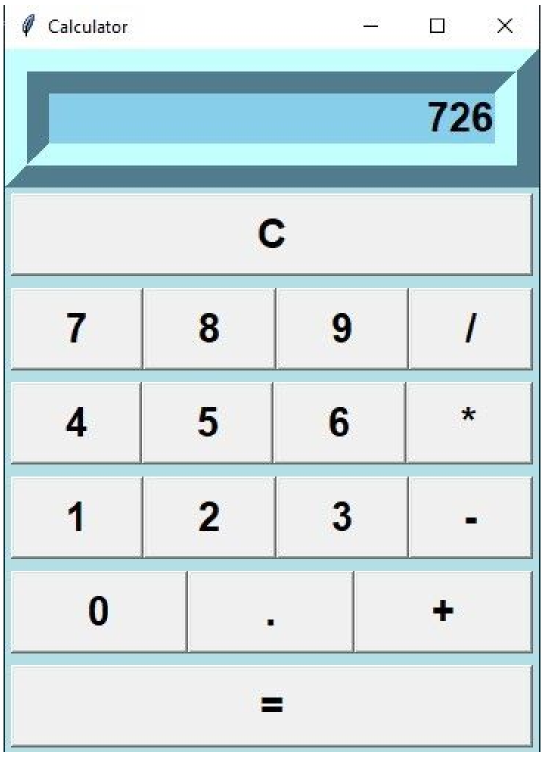
Result



4. Multiplication

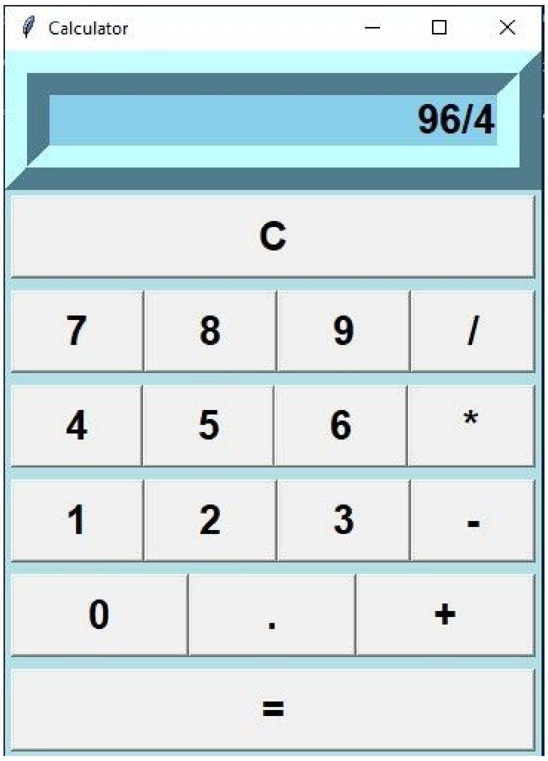


Result

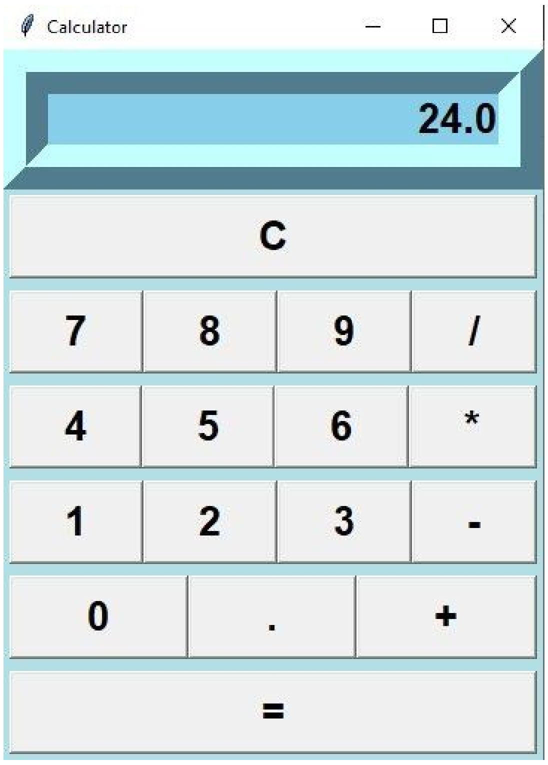


5. Division

Result

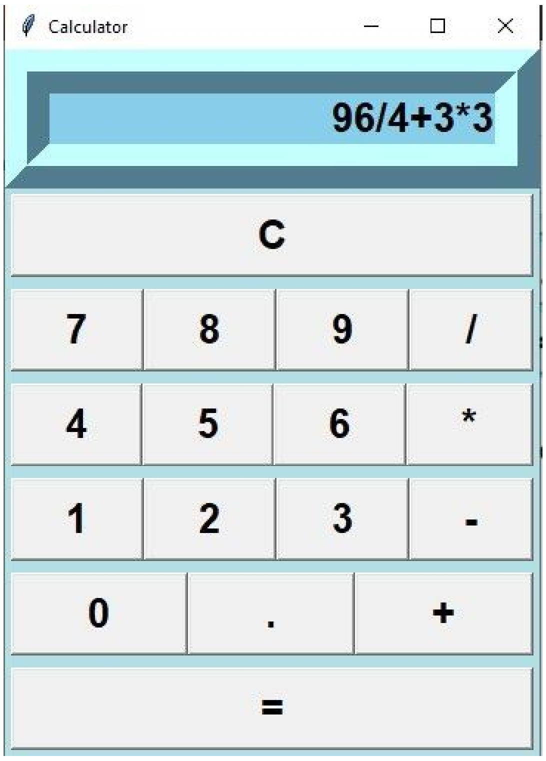

Calculator

96/4

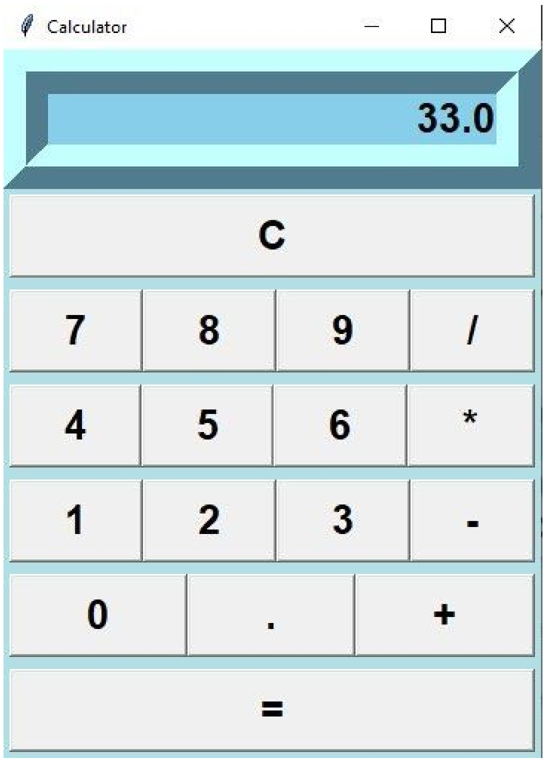

Calculator

24.0

6. Complex problems (involving more than one operator)

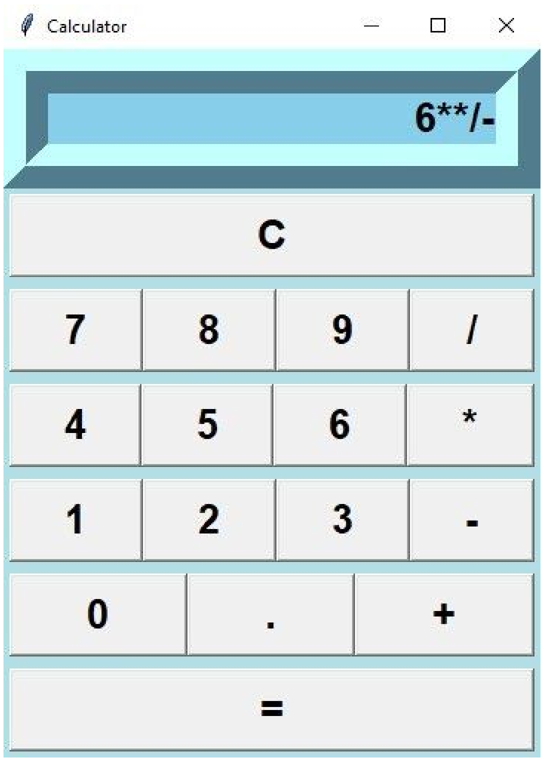

Calculator

96/4+3*3

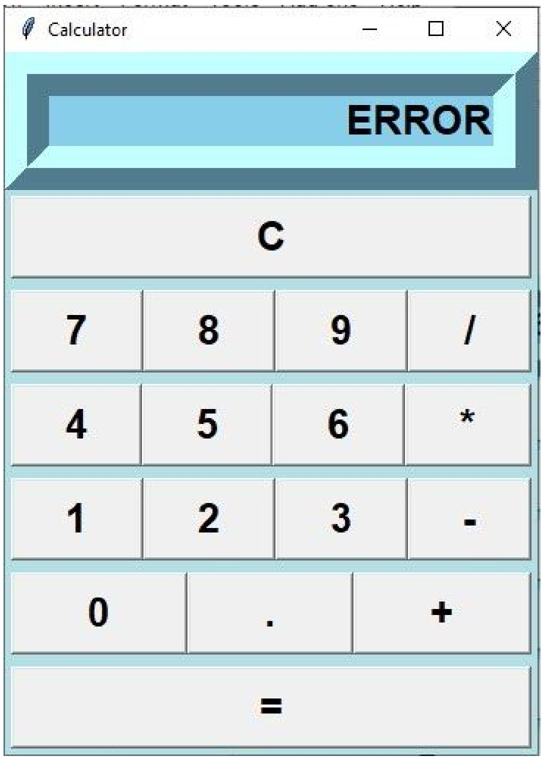

Calculator

33.0

7. Invalid Input



Result



Bibliography

Made by :- Vishal Meena,11A, K.V. Rly GIM
Editor :- Amit Meena, 11A, K.V. Rly GIM

Modulus used; Python documentation on tkinter

https://docs.python.org/3/library/tkinter.html , A (Very) Quick Look at Tcl/Tk.

Introduction to python; Wikipedia page on Python (Programming language)

https://en.wikipedia.org/wiki/Python_(programming_language) ,

Help in source code;

GeeksForGeeks Simple GUI Calculator using Tkinter,

https://www.geeksforgeeks.org/python-simple-gui-calculator-using-tkinter/

Simplified Python GUI Calculator,

https://www.simplifiedpython.net/python-calculator/

DATE :- 12-02-2020