SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya
TRUST

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

Let's Play SUDOKU

Online Edition!

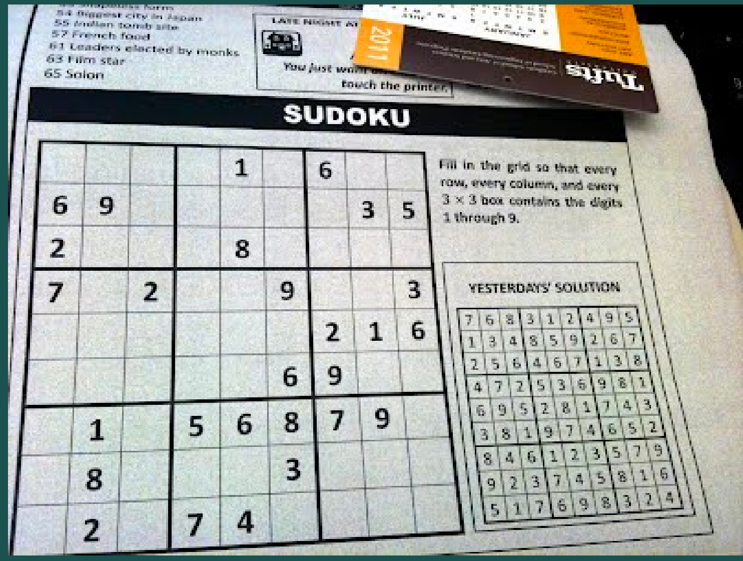
1]Aryaan Saraiya-16010221023
2]Adithe Shivaram-16010221024
3]Eshan Trivedi-16010221038

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

What is sudoku?

Su Doku, popular form of number game. In its simplest and most common configuration, sudoku consists of a 9 × 9 grid with numbers appearing in some of the squares. The object of the puzzle is to fill the remaining squares, using all the numbers 1–9 exactly once in each row, column, and the nine 3 × 3 subgrids. Sudoku is based entirely on logic, without any arithmetic involved, and the level of difficulty is determined by the quantity and positions of the original numbers. The puzzle, however, raised interesting combinatorial problems for mathematicians, two of whom proved in 2005 that there are 6,670,903,752,021,072,936,960 possible sudoku grids.


SUDOKU
Fill in the grid so that every row, every column, and every 3 × 3 box contains the digits 1 through 9.
YESTERDAYS' SOLUTION

How to play sudoku and basic tips

1]Only use the numbers 1 to 9,
2]Avoid trying to guess the solution to the puzzle,
3]Only use each number once in each row, column,grid,
4]Use the process of elimination as a tactic,
5]Use cross-hatching and penciling in techniques.

Problem Statement

The task is to solve the sudoku puzzle in as much less time as possible.It can be done by investigating different techniques for solving sudoku and comparing them for the most efficient solution. Sudoku itself can be solved using brute-force in a reasonable amount of time in most cases, but there are special cases where it takes a long time to brute-force. Therefore our task is to try to find effcient algorithms for all instances of the problem and evaluate them while using the optimal solver to get the solution for thesudoku puzzle

Tkinter in python

Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.

Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

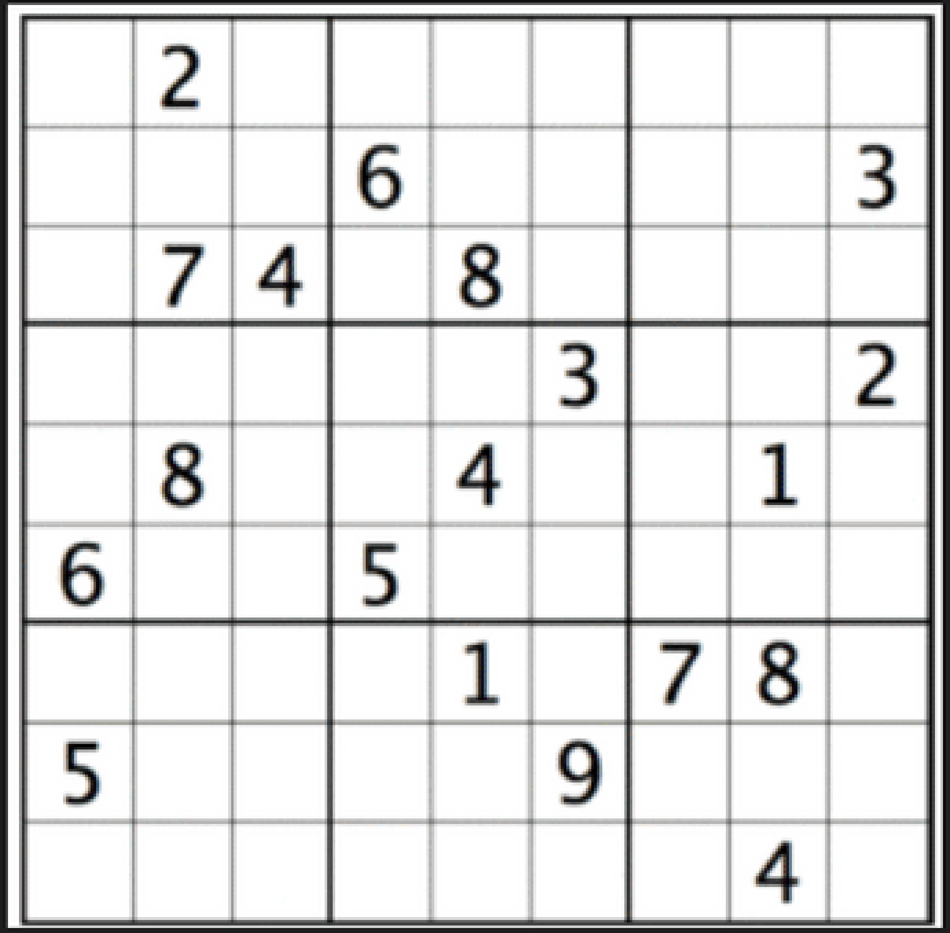
1]Import the Tkinter module.

2]Create the GUI application main window.

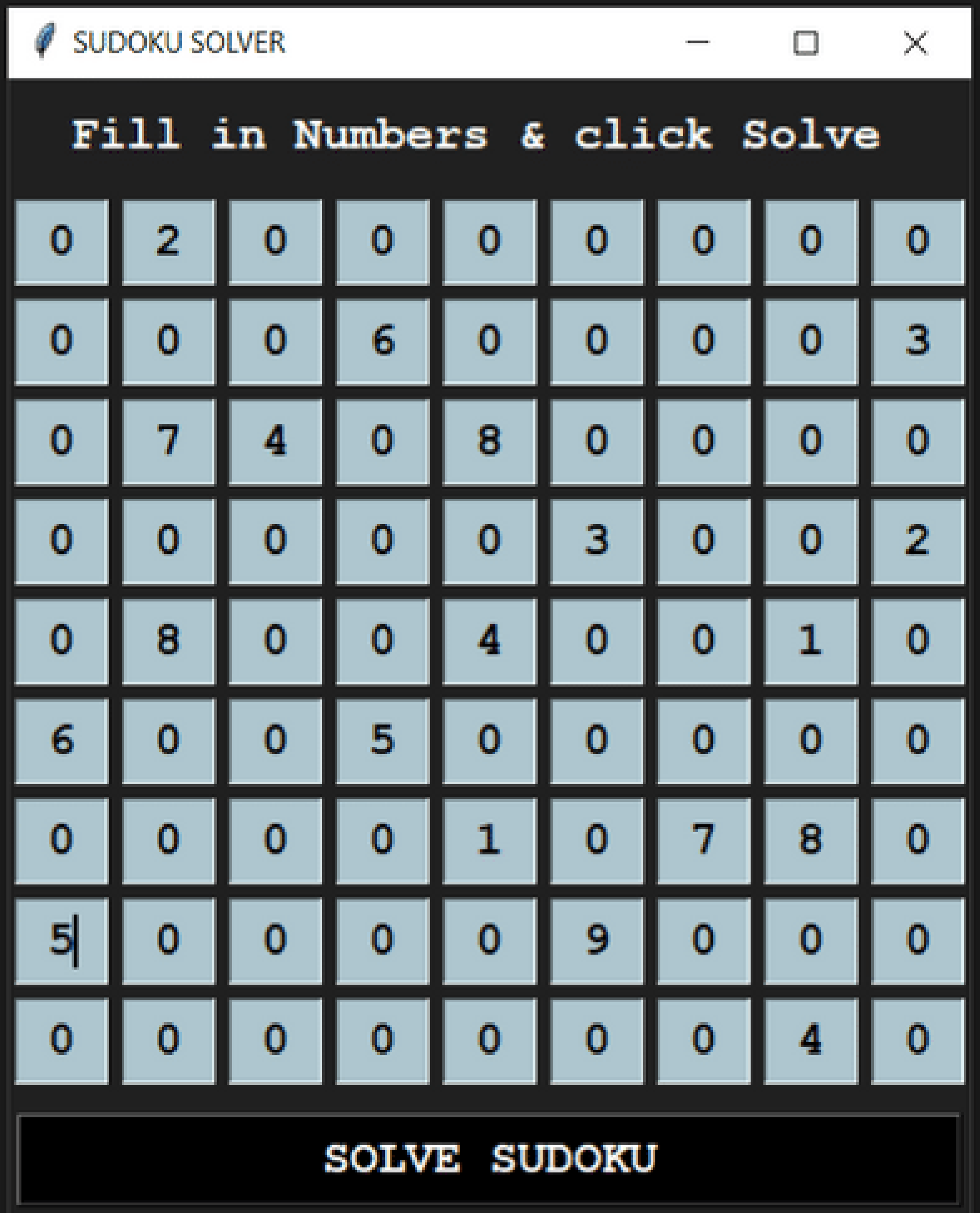
3]Add one or more of the above-mentioned widgets to the GUI application.

4]Enter the main event loop to take action against each event triggered by the user.

Preview and Question



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

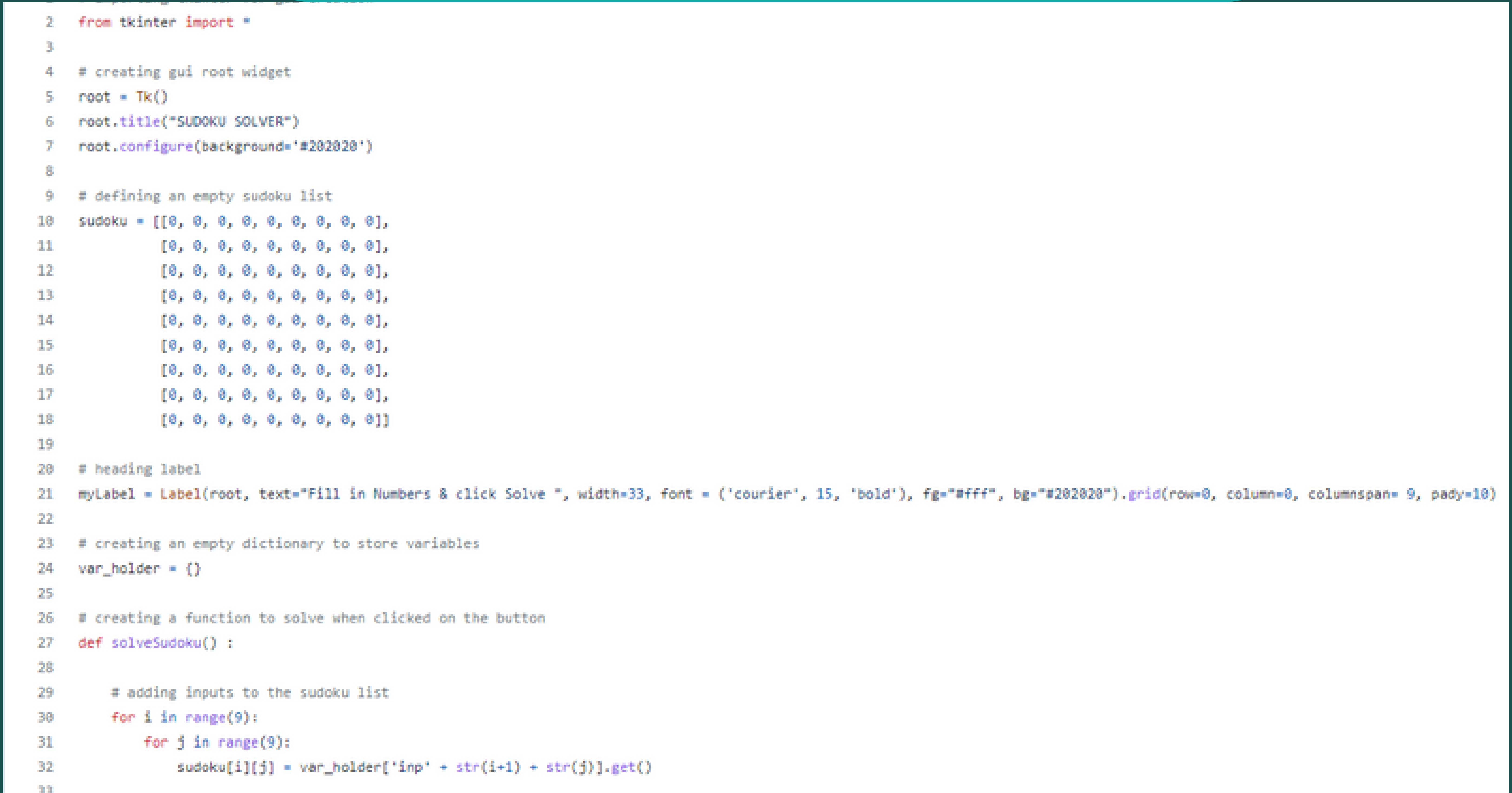


SUDOKU SOLVER

Fill in Numbers & click Solve

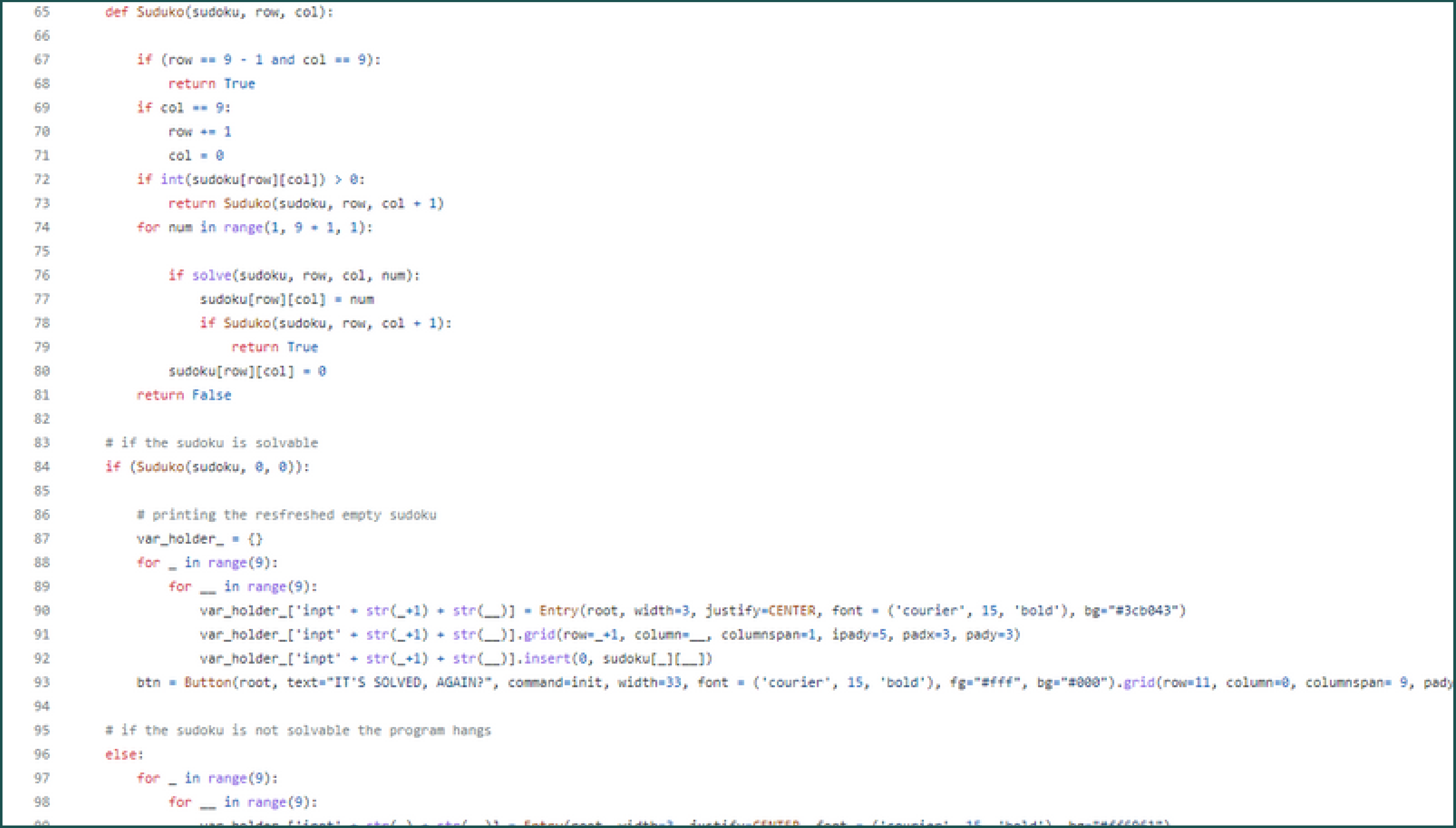
0	2	0	0	0	0	0	0	0
0	0	0	6	0	0	0	0	3
0	7	4	0	8	0	0	0	0
0	0	0	0	0	3	0	0	2
0	8	0	0	4	0	0	1	0
6	0	0	5	0	0	0	0	0
0	0	0	0	1	0	7	8	0
5	0	0	0	0	9	0	0	0
0	0	0	0	0	0	0	4	0

Code/Input

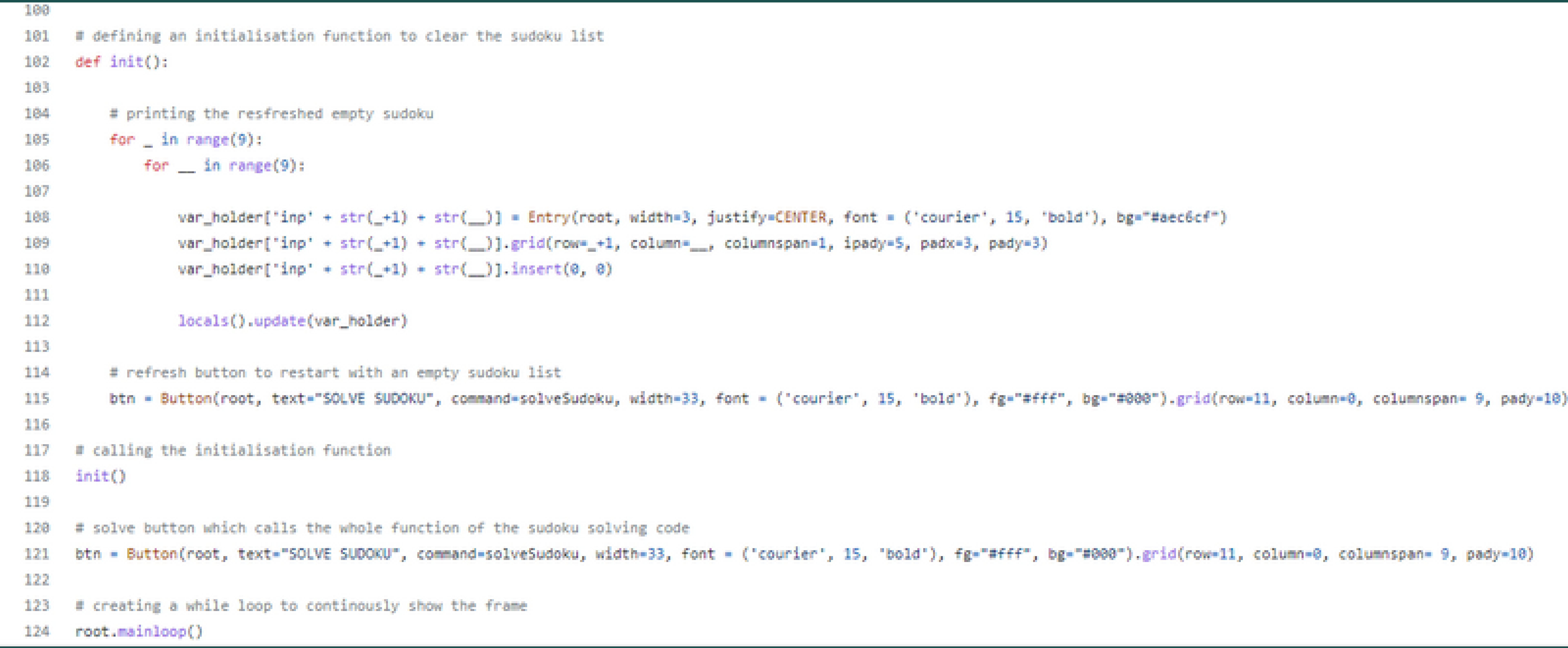


```python
from tkinter import *

# creating gui root widget
root = Tk()
root.title("SUDOKU SOLVER")
root.configure(background='#202020')

# defining an empty sudoku list
sudoku = [[0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0]]

# heading label
myLabel = Label(root, text="Fill in Numbers & click Solve ", width=33, font = ('courier', 15, 'bold'), fg="#fff", bg="#202020").grid(row=0, column=0, columnspan= 9, pady=10)

# creating an empty dictionary to store variables
var_holder = {}

# creating a function to solve when clicked on the button
def solveSudoku() :

    # adding inputs to the sudoku list
    for i in range(9):
        for j in range(9):
            sudoku[i][j] = var_holder['inp' + str(i+1) + str(j)].get()
```

```python
# printing the the sudoku matrix without using numpy
def puzzle(a):
    for i in range(9):
        for j in range(9):
            print(a[i][j],end = " ")
        print()

# creating a function to check the possibility of a correct answer at a particular position
def solve(sudoku, row, col, num):

    # does the number appear in the row ?
    for x in range(9):
        if int(sudoku[row][x]) == num:
            return False

    # does the number appear in the column ?
    for x in range(9):
        if int(sudoku[x][col]) == num:
            return False

    # does the number appear in the 3X3 square box ?
    startRow = row - row % 3
    startCol = col - col % 3
    for i in range(3):
        for j in range(3):
            if int(sudoku[i + startRow][j + startCol]) == num:
                return False

    return True
```



```python
def Suduko(sudoku, row, col):

    if (row == 9 - 1 and col == 9):
        return True
    if col == 9:
        row += 1
        col = 0
    if int(sudoku[row][col]) > 0:
        return Suduko(sudoku, row, col + 1)
    for num in range(1, 9 + 1, 1):

        if solve(sudoku, row, col, num):
            sudoku[row][col] = num
            if Suduko(sudoku, row, col + 1):
                return True
        sudoku[row][col] = 0
    return False

# if the sudoku is solvable
if (Suduko(sudoku, 0, 0)):

    # printing the resfreshed empty sudoku
    var_holder_ = {}
    for _ in range(9):
        for __ in range(9):
            var_holder_['inpt' + str(_+1) + str(__)] = Entry(root, width=3, justify=CENTER, font = ('courier', 15, 'bold'), bg="#3cb043")
            var_holder_['inpt' + str(_+1) + str(__)].grid(row=_+1, column=__, columnspan=1, ipady=5, padx=3, pady=3)
            var_holder_['inpt' + str(_+1) + str(__)].insert(0, sudoku[_][__])
    btn = Button(root, text="IT'S SOLVED, AGAIN?", command=init, width=33, font = ('courier', 15, 'bold'), fg="#fff", bg="#000").grid(row=11, column=0, columnspan= 9, pady

# if the sudoku is not solvable the program hangs
else:
    for _ in range(9):
        for __ in range(9):
            var_holder_['inpt' + str(_) + str(__)] = Entry(root, width=3, justify=CENTER, font = ('courier', 15, 'bold'), bg="[illegible]")
```

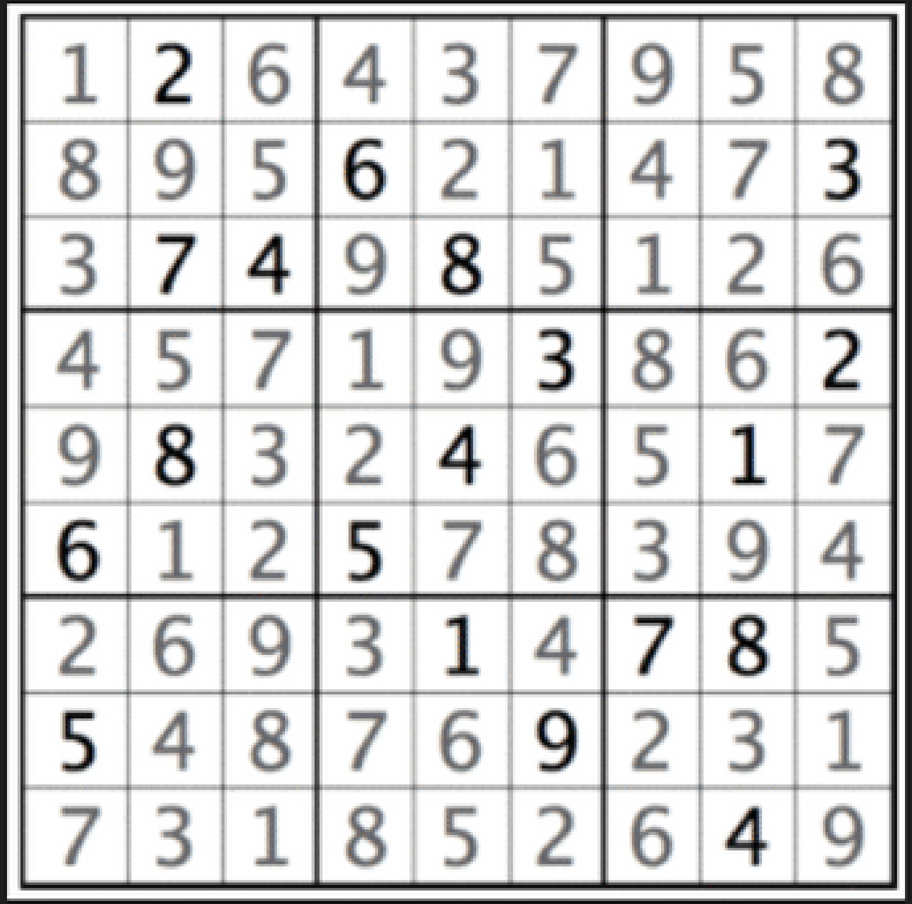


```python

# defining an initialisation function to clear the sudoku list
def init():

    # printing the resfreshed empty sudoku
    for _ in range(9):
        for __ in range(9):

            var_holder['inp' + str(_+1) + str(__)] = Entry(root, width=3, justify=CENTER, font = ('courier', 15, 'bold'), bg="#aec6cf")
            var_holder['inp' + str(_+1) + str(__)].grid(row=_+1, column=__, columnspan=1, ipady=5, padx=3, pady=3)
            var_holder['inp' + str(_+1) + str(__)].insert(0, 0)

            locals().update(var_holder)

    # refresh button to restart with an empty sudoku list
    btn = Button(root, text="SOLVE SUDOKU", command=solveSudoku, width=33, font = ('courier', 15, 'bold'), fg="#fff", bg="#000").grid(row=11, column=0, columnspan= 9, pady=10)

# calling the initialisation function
init()

# solve button which calls the whole function of the sudoku solving code
btn = Button(root, text="SOLVE SUDOKU", command=solveSudoku, width=33, font = ('courier', 15, 'bold'), fg="#fff", bg="#000").grid(row=11, column=0, columnspan= 9, pady=10)

# creating a while loop to continously show the frame
root.mainloop()
```

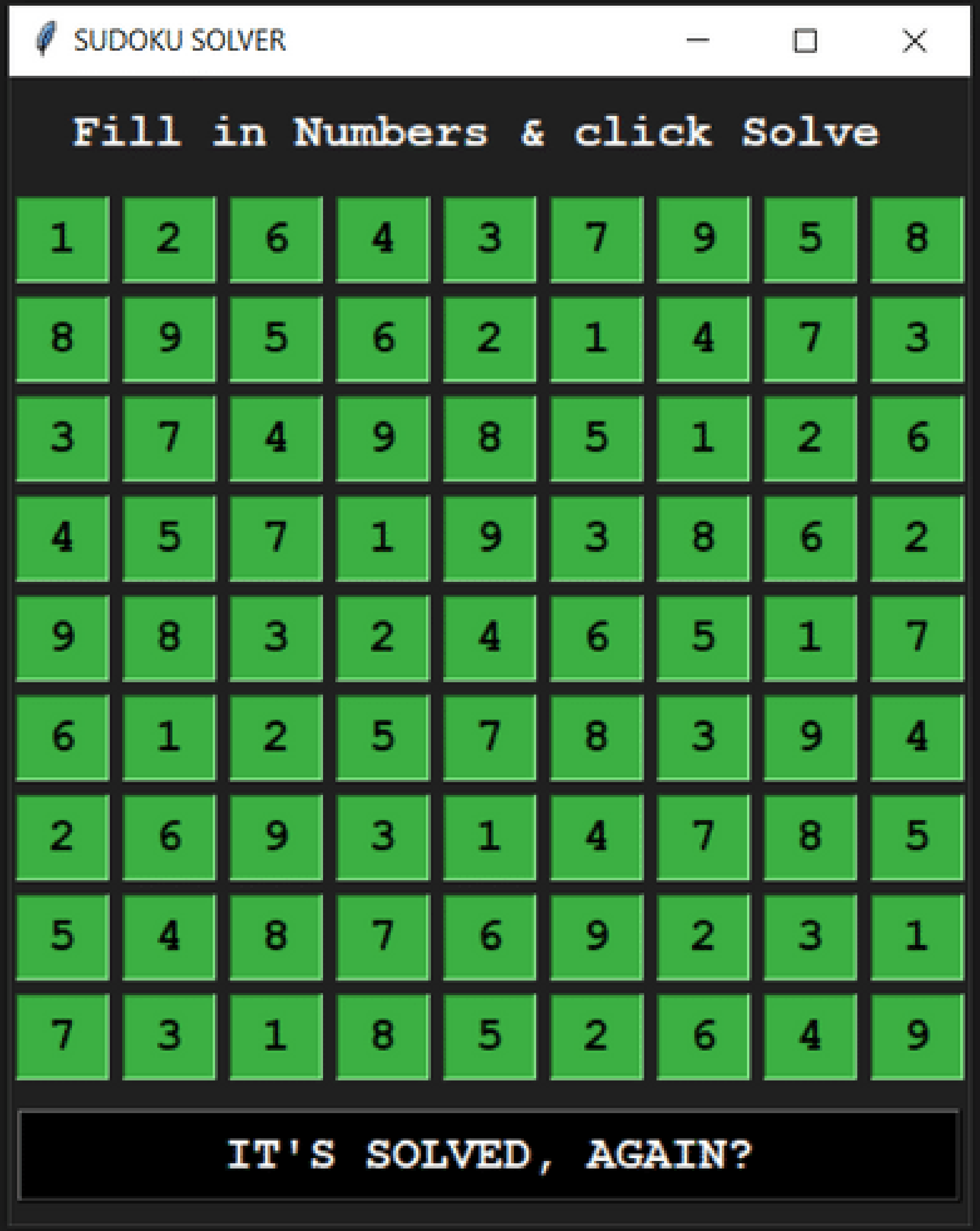
SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Output



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



Fill in Numbers & click Solve

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

IT'S SOLVED, AGAIN?

Conclusion

The algorithm is an appropriate method to find a solution faster and more efficient

- We tried to apply our in class developed knowledge about python as language in making a useful project.
-
- The project taught us logic building ,team coordination and working as a team .
- We also learnt about creating stuff more organised and adding unique features to project to make it more amazing