Programming Project

Instructions:

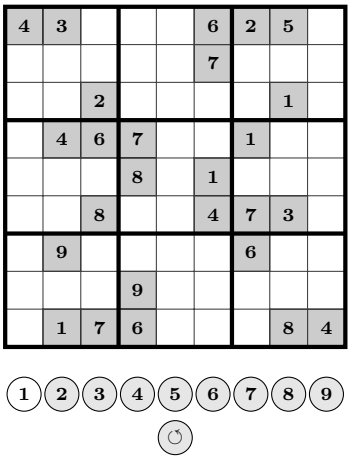
- **The project requires completing tasks by December 06, 2024, at 4:00 pm.**
- **The Sudoku game consists of a 9 by 9 grid. Its objective is to populate the entire grid with the digits 1 through 9 so that no duplicate digits are in any row, column, or grid box. Furthermore, the game begins with a few given values in random cells that cannot be changed as illustrated.**

Terminal Display

```
   0  1  2    3  4  5    6  7  8
 ---------------------------------
0 :(4)(3)[ ]::[ ][ ](6)::(2)(5)[ ]:
1 :[ ][ ][ ]::[ ][ ](7)::[ ][ ][ ]:
2 :[ ][ ](2)::[ ][ ][ ]::[ ](1)[ ]:
 ---------------------------------
3 :[ ](4)(6)::(7)[ ][ ]::(1)[ ][ ]:
4 :[ ][ ][ ]::(8)[ ](1)::[ ][ ][ ]:
5 :[ ][ ](8)::[ ][ ](4)::(7)(3)[ ]:
 ---------------------------------
6 :[ ](9)[ ]::[ ][ ][ ]::(6)[ ][ ]:
7 :[ ][ ][ ]::(9)[ ][ ]::[ ][ ][ ]:
8 :[ ](1)(7)::(6)[ ][ ]::[ ](8)(4):
 ---------------------------------

 :::[1](2)(3)(4)(5)(6)(7)(8)(9):::
```

GUI Display



Your objective is to create two identical Sudoku game programs using different programming languages. They can be both terminal or GUI or a mixture of the two. The games must adhere to the following criteria and constraints:

- ☐ **The programs must use at least 5 hard-coded puzzles where a hard-coded puzzle consists of the completed grid and the locations of the given values. For instance, the above puzzle would be**

  ```
  439186257581327469762549813246735198973861542158294736895473621624918375317652984
  110001110000001000001000010011100100000101000001001110010000100000100000011100011
  ```

  where the first row is the completed grid and the second row is a Boolean representation of locations of the given values in row order.
- ☐ **Besides allowing the player to select digits and cells, the player should be able to undo previous turns.**
- ☐ **Both programs must display a grid and a list of digits as illustrated above.**
- ☐ **For terminal programs, the cells that contain given values must be enclosed in parentheses while the other cells are enclosed in square braces.**
- ☐ **For GUI programs, the background color of the cells that contain given values must be gray while the background color of the other cells must be white.**
- ☐ **For terminal programs, the selected digit in the list must be enclosed in square braces while the others are enclosed in parentheses.**
- ☐ **For GUI programs, the background color of the button of the selected digit in the list must differ from the background color of the remaining buttons.**
- ☐ **The selected digit must stay the same until the player changes it. Likewise, the default selected digit must be 1.**
- ☐ **For terminal programs, if an invalid input for a digit or a cell is provided, an error message must be displayed and the game display must be reloaded.**
- ☐ **For terminal programs, if an attempt to populate a cell is illegal, display an error message that states exactly one conflict.**
- ☐ **For GUI programs, if an attempt to populate a cell is illegal, make exactly one outline of a conflicting cell red which should remain until another event occurs.**
- ☐ **For terminal programs, an option to quit the game must be provided.**

- **You may work in groups of at most two members. Each member must define the game in a different language. However, the tasks' algorithms must be identical for both languages.**
- **The game must be defined in two languages to receive credit regardless of the group size.**
- **A typed document must be submitted by October 25 that provides the group members' names, their programming languages, and interface types (terminal or GUI).**

- **Defining additional functions besides the task functions as helper functions are allowed.**
- **Only if a task is completed accurately will it receive points.**
- **You must present your programs on December 6. Your presentation must explain your game attributes and the function algorithms (in pseudocode).**
- **Cheating of any kind is prohibited and will not be tolerated.**
- **Violating and failing to follow any rules will result in an automatic zero (0) for the project.**

Game Grading

Task	Points	Earned
01	1	
02	1	
03	1	
04	1	
05	1	
06	1	
07	1	
08	1	
09	1	
10	1	
Total	10	

- Tasks highlighted in **red** must be submitted by November 8. Their grades are final.
- Tasks highlighted in **orange** must be submitted by November 22. Their grades are final.

Tasks:

1. A function named `select_puzzle()` that randomly selects one of the hard-coded puzzles.
2. A function named `initialize()` that creates and initializes the game objects and runs the game loop.
3. A function named `has_won()` that returns true if the game grid is filled.
4. A function named `is_valid_row()` that checks if a conflict occurs when assigning the selected digit to a specified cell within the row of the selected cell.
5. A function named `is_valid_column()` that checks if a conflict occurs when assigning the selected digit to a specified cell within the column of the selected cell.
6. A function named `is_valid_box()` that checks if a conflict occurs when assigning the selected digit to a specified cell within the box of the selected cell.
7. A function named `undo()` that undoes the previous change to the game grid.
8. A function named `select_digit()` that modifies the selected digit.
9. A function named `populate_cell()` that tries to assign the selected digit to a cell.
10. A function named `display()` that displays (or modifies) the game grid and digit list.

Hints & Suggestions:

- Certain representations of the game objects will make it possible for every function except for `Initialize()` and `Display()` to have a constant runtime.
- Using a single or multiple stacks will be required; a single stack will be needed depending on the representation of the game objects.
- Every two-dimensional array can be transposed to a one-dimensional array with a simple hash function.
- The boxes of the Sudoku can be interpreted as a 3 by 3 grid. Mapping the row and column indices of the Sudoku grid separately to a cell of a 3 by 3 grid can be done with two simple hash functions.
- Declare a single or three Boolean two-dimensional (or one-dimensional) arrays that track digits in each row, column, and box. Offset values are needed to get to the correct indices whenever a single array is used.
- Boolean values can be represented as characters. Hence, char arrays can be used in place of Boolean arrays.