**Old Phone Pad Program Documentation**

**Overview**

The program processes a sequence of key presses and returns the correct letters.

**Description**

* The **OldPhonePad** **method** accepts the parameter value of input string that has numbers as string of key press and execute to return the corresponding letters. The function handle key presses, backspace using ( \* ) and send button with ( # ) character.
* **Dictionary** is created for the key, value mapping. Key is number on the keypad and value is letters of relevant key.
* **StringBuilder** is used to return the letters.
* First, **checking** whether **#** character is included or not. No code execute unless the input doesn’t have # character at the end. It will response the message.
* **Index** variable is created to loop the each number of the input string.
* **While** loop is used to iterate the each character of input as long as the index is not greater than the input length.
* **Currentkey** variable is created to keep the current string from the index number of the input.
* **Checking # in current key** is to escape from the executable code as # is just the end of input.
* **Checking \* in current key** is to handle backspace to remove the last character in the output is removed if the character is present.
* **Checking currentkey** is included in keypad that is object of dictionary. If it contains in keypad,

**While** loop including input length checking, duplicate key checking (means same key) and space checking will execute to increase count and index of the key. Count variable is used to know how many duplicate key is existing in the input string.

* Using **modulus** operator to handle the wrapping around of characters when user presses a key more times than the number of available characters mapped to that key.

**Eg**. Press 4 times 2222# = 4 -1 % 3 = 3 % 3= 0 = A

**Conclusion**

The program simulates input from old phone keypad using dictionaries, loops and basic string operations to generate correct output.