**OldPhonePadDecoder**

**Overview**

The **OldPhonePadDecoder** is a C# console application that decodes numeric sequences entered using the traditional mobile phone keypad (T9 input) into text. This emulates the way users used to compose messages before smartphones, using multi-tap input on numeric keys.

**How It Works**

The application takes input strings representing sequences of key presses and decodes them into letters using the following rules:

* Numbers 2 through 9 map to their respective letters as on a standard mobile keypad.
* Repeated digits indicate multiple key presses to select different letters.
* A space ' ' indicates a pause between letters (i.e., separate character input).
* An asterisk '\*' functions as a backspace (deletes the last letter entered).
* A hash '#' indicates the end of the input.
* Unknown characters (letters, punctuation, or other symbols) are ignored.

**Keypad Mapping**

| **Key** | **Letters** |
| --- | --- |
| 2 | ABC |
| 3 | DEF |
| 4 | GHI |
| 5 | JKL |
| 6 | MNO |
| 7 | PQRS |
| 8 | TUV |
| 9 | WXYZ |

**Example Inputs and Outputs**

Here are a few examples of how inputs are decoded:

| **Input** | **Output** | **Explanation** |
| --- | --- | --- |
| 2# | A | Single press of 2 gives 'A' |
| 7777 666 555 3# | SEEK | 7777 = S, 666 = O, 555 = L, 3 = D |
| 8 88777444666\*664# | TURING | Simulates typing and correcting |
| 22 33 44 55 66 77 88 99# | BEHKNQUX | Maps each key to 2nd letter |
| 2 2 2\*2\*2# | AAA | Inputs 3 As with deletes |

**Code Structure**

**OldPhonePadDecoder Class**

**Static Members:**

* Keypad: A dictionary mapping number keys to their associated characters.

**Static Methods:**

* OldPhonePad(string input):  
  Main decoding function that:
  + Parses input string.
  + Interprets digits, backspaces (\*), pauses ( ), and termination (#).
  + Outputs the decoded message.
* StringBlock(String Builder stringblock, String Builder result):  
  Helper function that:
  + Converts a block of repeated digits into a single character based on the keypad mapping.
  + Appends the character to the result.

**Main () Method:**

* Contains a set of predefined test inputs to demonstrate decoding functionality.
* Outputs the original input alongside the resulting decoded message.

**How to Run**

1. Open the project in any C# development environment (e.g., Visual Studio or Rider).
2. Build and run the program.
3. Observe decoded outputs for each test case in the console.

**Limitations**

* Only processes digits 2-9, \*, #, and spaces.
* Ignores invalid inputs (like 0, 1, letters, or symbols).
* No support for predictive T9 input (just multi-tap).