<script>

// Javascript program to implement the

// above approach

let hashTable = [ "", "", "abc", "def", "ghi", "jkl",

                  "mno", "pqrs", "tuv", "wxyz" ];

// A recursive function to print all possible

// words that can be obtained by input number[]

// of size n. The output words are one by one

// stored in output[]

function printWordsUtil(number, curr, output, n)

{

    // Base case, if current output

    // word is prepared

if (curr == n)

      {

        document.write(output.join("") + "<br>")

        return;

      }

      // Try all 3 possible characters for current

      // digir in number[] and recur for remaining digits

    for(let i = 0;

            i < hashTable[number[curr]].length;

            i++)

    {

        output.push(hashTable[number[curr]][i]);

 printWordsUtil(number, curr + 1, output, n);

        output.pop();

        if(number[curr] == 0 || number[curr] == 1)

            return

    }

}

// A wrapper over printWordsUtil(). It creates

// an output array and calls printWordsUtil()

function printWords(numbers, n)

{

printWordsUtil(number, 0, [], n);

}

// Driver code

let number = [ 2, 3, 4 ];

let n = number.length;

printWords(number, n);

// This code is contributed by avanitrachhadiya2155

</script>