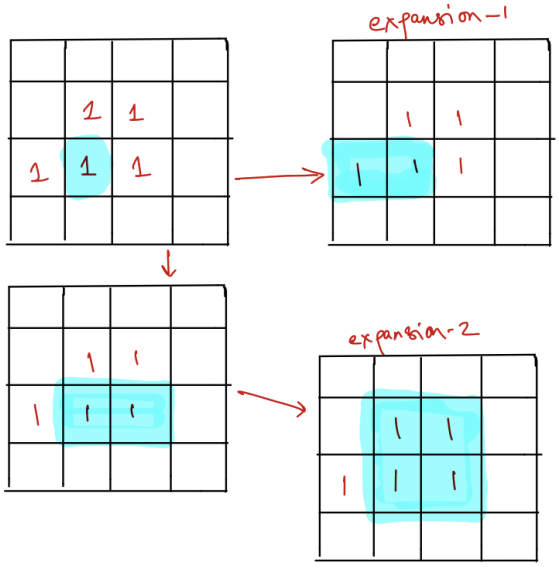
COL215P: Software Assignment - 2

Boolean Function Expansion

Chinmay Mittal(2020CS10336) and Dhruv Tyagi(2020CS10341)

Do all expansions result in an identical set of terms ?

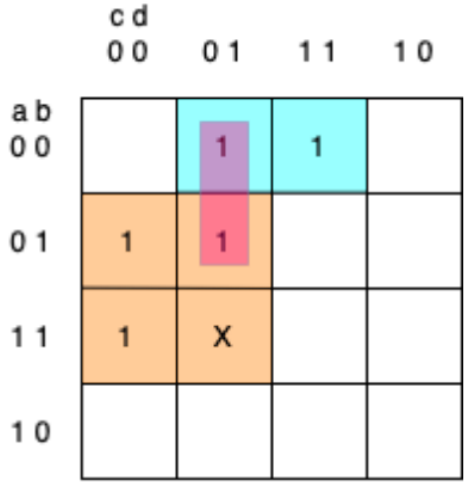
All expansions do not give the same results and thus different expansion sequences can lead to different answers some of which can be non-optimal i.e. they are not the largest region enclosing the term though they are maximal i.e. they cannot be expanded further. This is shown by the following case. Thus all expansions are not equally good. Some expansions allow further expansions whereas others lead to maximal regions which can not be further expanded.

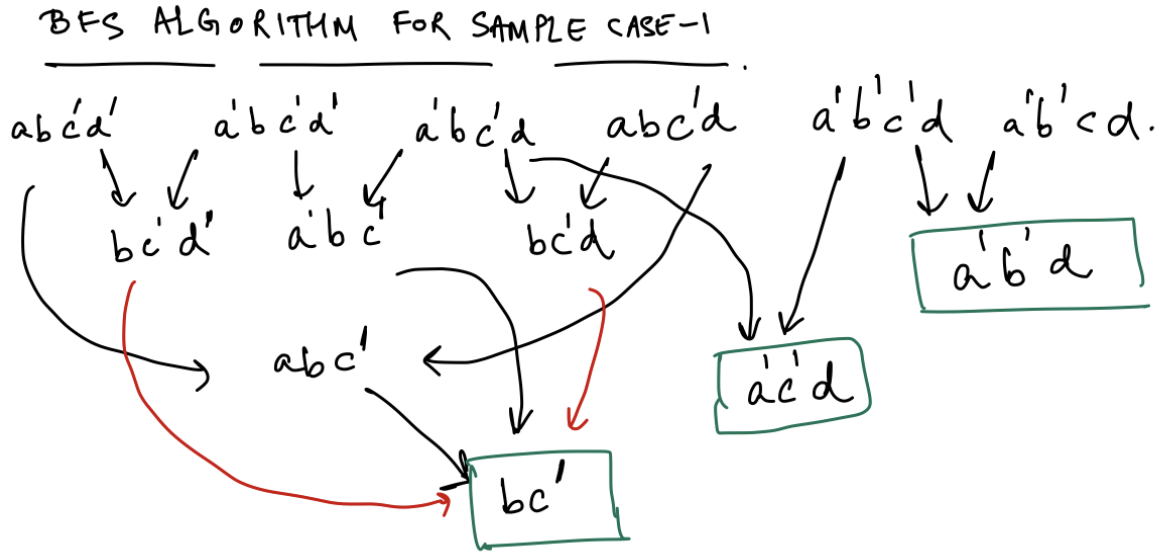


Algorithm for region Expansion

We essentially use a Multi-Source Breadth First Search Algorithm to maximally expand regions. As the depth increases, we move to larger regions that encapsulate more terms. We use this BFS algorithm to find all legal regions. At depth=0 we have all legal regions of size one. These correspond to the regions given by the true and don't care terms. For each region we try all possible expansions i.e. variable eliminations. For eg. to expand the term *abcde* we first check variable a. This region can be expanded by eliminating variable a if *a'bcde* is also a legal region, this will give a region of twice the size at the next depth namely *bcde*. We repeat this process

depth wise. For each legal region at each depth we consider all variables for expansion and check if the corresponding region for expansion is also a legal region. If so, we can eliminate the variable and achieve a legal region of larger size at the next depth. If we cannot expand a region by eliminating any variable for a term then that means that the corresponding region is maximal. This BFS algorithm can then compute all legal regions which can be used later. Once we know all legal regions. For each term we can do a DFS on the graph. This DFS is similar; it tries all expansions and since we already know all legal regions we can directly check if an expansion is possible or not. Leaf nodes represent maximally enclosing regions for the original term. We return the optimal region among all the possible maximal regions. Our algorithm is optimal since it computes the largest region enclosing every term. Also during BFS we use memoization i.e. we do not re-process a node we have processed before this might happen since different regions might combine to give the same region, in these cases we ensure that this region is processed only once.

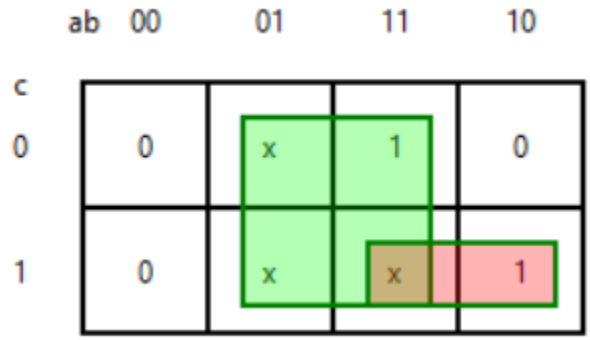




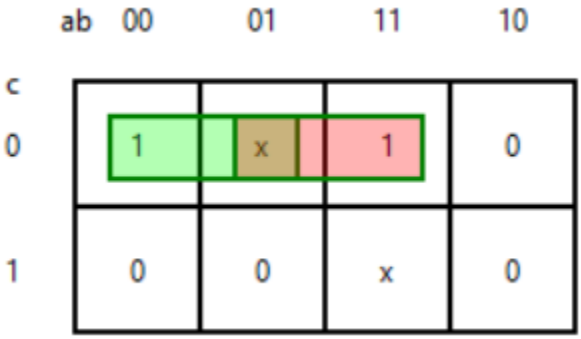
Time Complexity

Let us consider the worst case when we are given all the terms in the list i.e. the true and don't care terms cover the entire k-map. If there are N variables then the number of variables in the func_true and func_dc list will be 2^N. Our BFS traversal will then generate all regions which are 3^N (for every variable we have 3 choices of whether it doesn't appear or is one or is zero). Thus the number of nodes in the BFS graph are 3^N. Every node has an edge corresponding to trying a variable for expansion hence there are 3^N. N edges in the graph. Thus the time complexity of the BFS is *O(E + V)* which is O(3^N N) which is O($M^{1.58}$ logM) (1.58 is log 3 to the base 2). Thus our algorithm is polynomial time in the size of the input list.

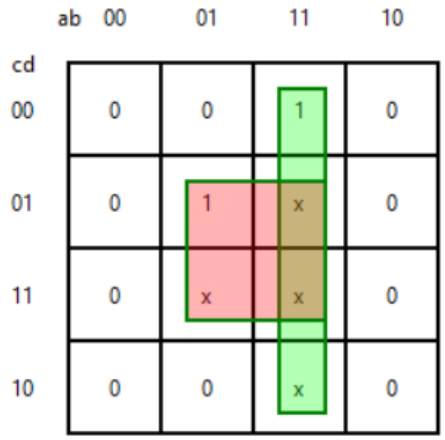
Test Cases



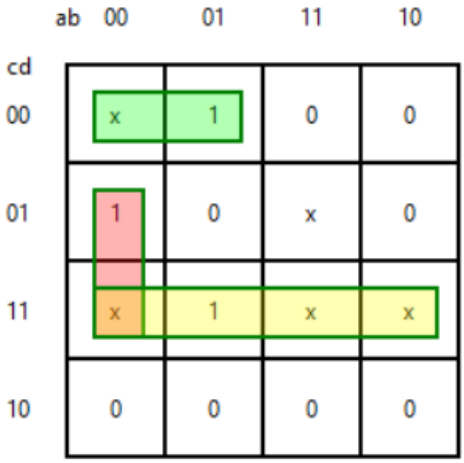

```
func_TRUE = ["a'bc'd'", "abc'd'", "a'b'c'd", "a'bc'd", "a'b'cd"]
func_DC = ["abc'd"]
output = ['b', 'ac']
```




```
func_TRUE = ["a'b'c'", "abc'" ]
func_DC = ["a'bc'" , "abc" ]
output =  ["a'c'", "bc'"]
```




```
func_TRUE = ["abc'd'", "a'bc'd"]
func_DC = [ "abc'd", "a'bcd", "abcd", "abcd'"]
output = ['ab', 'bd']
```




```
func_TRUE = ["a'bc'd'", "a'b'c'd", "a'bcd"]
func_DC = [ "a'b'c'd'", "a'b'cd", "abcd", "abc'd", "ab'cd"]
output = ["a'c'd'", "a'b'd", 'cd']
```

Additional Test Cases (Without K-MAP)

```
Input:
func_TRUE = ["a'b'"]
func_DC = ["a'b"]

Output:
["a'"]

Input:
func_TRUE = ["ab'", "a'b"]
func_DC = ["a'b'"]

Output:
["b'", "a'"]

Input:
func_TRUE = ["abc", "ab'c"]
func_DC = ["abc'", "ab'c'", "a'bc'", "a'b'c"]

Output:
["a", "a"]

Input:
func_TRUE = ["a'b'c", "a'b'c'", "ab'c"]
func_DC = ["ab'c'", "a'bc", "a'bc'"]

Output:
["b'", "b'", "b'"]

Input:
func_TRUE = ["a'bcd", "a'bcd'", "a'b'c'd'", "a'bc'd"]
func_DC = ["abcd", "abcd'", "abc'd", "abc'd'", "ab'cd'", "ab'c'd",
"a'bc'd'", "a'b'cd", "a'b'c'd"]

Output:
["b", "b", "a'c'", "b"]

Input:
```

```
func_TRUE = ["a'b'cd", "a'bcd'", "ab'c'd'", "a'bc'd", "abc'd"]
func_DC = ["abcd", "abcd'", "ab'cd", "ab'cd'", "ab'c'd"]

Output:
["b'cd", "bcd'", "ab'", "bc'd", "ad"]

Input:
func_TRUE = ["a'bc'de'", "a'bc'de", "a'bc'd'e", "a'bcd'e", "ab'c'd'e'",
"abc'd'e'", "a'b'c'de", "abc'd'e"]
func_DC = ["abcd'e", "abc'de", "ab'cde'", "ab'cd'e'", "ab'c'de",
"ab'c'd'e", "a'bcde'", "a'b'cde'", "a'b'cd'e"]

Output:
["a'bde'", "c'de", "bd'e", "bd'e", "ac'd'", "ac'd'", "c'de", "bd'e"]

Input:
func_TRUE = ["a'b'c'd'e", "a'bcde", "ab'cd'e", "ab'cde'", "abc'd'e'",
"ab'cde", "abcde", "a'bcd'e", "a'bcde'", "abc'de"]
func_DC = ["abc'd'e", "ab'cd'e'", "ab'c'de'", "ab'c'd'e", "ab'c'd'e'",
"a'bc'de", "a'bc'd'e", "a'bc'd'e'", "a'b'c'de'", "a'b'c'd'e'"]

Output:
["c'd'", "bde", "ab'd'", "ab'e'", "c'd'", "ab'c", "bde", "a'be", "a'bcd",
"bde"]

Input:
func_TRUE = ["a'b'c'def'", "abc'de'f'", "a'bc'd'e'f", "a'b'cd'e'f'",
"ab'cdef'", "ab'cd'ef", "abcdef", "a'b'cdef", "ab'c'def'", "ab'c'd'e'f'",
"ab'cde'f'", "abcd'ef", "a'bcd'e'f'", "ab'c'd'ef", "a'b'c'de'f",
"a'bc'def'", "ab'c'def", "a'b'c'd'e'f'", "abc'def", "abc'def'", "a'bcdef",
"a'b'cd'ef"]
func_DC = ["abcde'f", "abcde'f'", "abcd'ef'", "a'bcd'ef'", "a'bcd'e'f",
"a'bc'd'ef'", "ab'cd'ef'", "ab'cd'e'f", "a'b'c'd'ef'"]

Output:
["c'def'", "abde'f'", "a'bd'e'f", "a'cd'e'f'", "ab'def'", "acd'e",
"bcdef", "a'cdef", "c'def'", "b'c'd'e'f'", "acde'f'", "acd'e",
"a'cd'e'f'", "ab'd'ef", "a'b'c'de'f", "c'def'", "ac'de", "b'c'd'e'f'",
"ac'de", "c'def'", "bcdef", "b'cd'ef"]
```

```
Input:
func_TRUE = ["ab'cde'f'", "a'b'c'd'ef", "a'b'c'de'f", "a'bc'def'",
"a'b'cde'f", "a'b'c'd'e'f'", "a'bcd'e'f", "a'bc'de'f"]
func_DC = ["abcdef'", "abcde'f", "abcde'f'", "abcd'ef", "ab'cdef'",
"ab'cd'e'f", "ab'c'de'f", "ab'c'd'ef'", "ab'c'd'e'f", "a'bcd'e'f'",
"a'bc'd'ef", "a'b'cdef", "a'b'cdef'", "a'b'cde'f'", "a'b'cd'e'f",
"a'b'c'def", "a'b'c'def'", "a'b'c'd'ef'", "a'b'c'd'e'f"]

Output:
["b'cdf'", "a'b'c'f", "b'c'e'f", "a'c'def'", "a'b'e'f", "a'b'c'd'",
"a'cd'e'f", "a'c'de'f"]

Input:
func_TRUE = ["ab'cd'e'f'g'", "a'b'cd'e'f'g'", "a'bcd'efg", "abcd'e'f'g",
"ab'cd'e'f'g", "a'bc'd'e'fg'", "a'b'cd'efg'", "ab'c'd'efg'",
"a'b'c'de'fg", "ab'cd'ef'g'", "abcdefg'", "a'b'c'd'e'f'g", "a'bcd'ef'g'",
"a'b'c'def'g'", "a'bc'd'efg'", "abcd'e'fg'", "a'b'c'd'efg'"]
func_DC = ["abcdefg", "abcdef'g", "abcdef'g'", "abcde'fg", "abcde'fg'",
"abcde'f'g", "abcde'f'g'", "abcd'efg", "abcd'efg'", "abcd'ef'g",
"abcd'ef'g'", "abcd'e'fg", "abcd'e'f'g'", "abc'defg", "abc'defg'",
"abc'def'g", "abc'def'g'", "abc'de'fg", "abc'de'fg'", "abc'de'f'g",
"abc'de'f'g'", "abc'd'efg", "abc'd'efg'", "abc'd'ef'g", "abc'd'ef'g'",
"abc'd'e'fg", "abc'd'e'fg'", "abc'd'e'f'g", "abc'd'e'f'g'", "ab'cdefg",
"ab'cdefg'", "ab'cdef'g", "ab'cdef'g'", "ab'cde'fg", "ab'cde'fg'",
"ab'cde'f'g", "ab'cde'f'g'", "ab'cd'efg", "ab'cd'efg'", "ab'cd'ef'g", "ab'cd'e'fg",
"ab'cd'e'fg'", "ab'c'defg", "ab'c'def'g", "ab'c'def'g'", "ab'c'de'fg",
"ab'c'de'fg'", "ab'c'de'f'g'", "ab'c'd'efg", "ab'c'd'ef'g", "ab'c'd'e'fg",
"ab'c'd'e'fg'", "ab'c'd'e'f'g", "ab'c'd'e'f'g'", "a'bcdefg", "a'bcdef'g",
"a'bcdef'g'", "a'bcde'fg", "a'bcde'fg'", "a'bcde'f'g", "a'bcde'f'g'",
"a'bcd'efg'", "a'bcd'ef'g", "a'bcd'e'fg", "a'bcd'e'fg'", "a'bcd'e'f'g'",
"a'bc'defg", "a'bc'defg'", "a'bc'def'g", "a'bc'def'g'", "a'bc'de'fg",
"a'bc'de'fg'", "a'bc'de'f'g", "a'bc'd'efg", "a'bc'd'ef'g", "a'bc'd'ef'g'",
"a'bc'd'e'fg", "a'bc'd'e'f'g", "a'bc'd'e'f'g'", "a'b'cdefg", "a'b'cdefg'",
"a'b'cdef'g", "a'b'cde'fg", "a'b'cde'fg'", "a'b'cde'f'g", "a'b'cd'efg",
"a'b'cd'ef'g'", "a'b'cd'e'fg", "a'b'cd'e'fg'", "a'b'cd'e'f'g",
"a'b'c'defg", "a'b'c'defg'", "a'b'c'def'g", "a'b'c'de'fg'",
"a'b'c'de'f'g", "a'b'c'de'f'g'", "a'b'c'd'ef'g", "a'b'c'd'ef'g'",
"a'b'c'd'e'fg", "a'b'c'd'e'fg'", "a'b'c'd'e'f'g'"]

Output:
```

```
["d'e'g'", "d'e'g'", "bfg", "ab", "b'd'e'", "e'f", "d'fg'", "d'fg'",
"e'f", "cd'g'", "ab", "c'd'e'", "cd'g'", "c'def'", "d'fg'", "e'f",
"d'fg'"]

Input:
func_TRUE = ["a'b'cd'e'fg'", "abcdefg'", "a'b'cdef'g'", "a'bc'd'ef'g'",
"a'b'cdef'g", "a'b'cdefg", "a'bcd'e'fg'", "ab'c'd'efg", "ab'cd'ef'g",
"a'b'c'defg'", "a'bcd'e'fg", "abc'def'g", "a'bc'd'e'fg'", "a'bc'd'e'f'g'",
"ab'c'de'fg'", "a'bc'd'ef'g", "abc'de'fg", "a'b'c'de'fg"]
func_DC = ["abcdef'g", "abcdef'g'", "abcde'fg", "abcde'fg'", "abcde'f'g",
"abcde'f'g'", "abcd'efg", "abcd'efg'", "abcd'ef'g", "abcd'ef'g'",
"abcd'e'f'g", "abcd'e'f'g'", "abc'defg", "abc'defg'", "abc'def'g'",
"abc'de'fg'", "abc'de'f'g", "abc'de'f'g'", "abc'd'efg", "abc'd'efg'",
"abc'd'ef'g", "abc'd'e'fg'", "abc'd'e'f'g", "abc'd'e'f'g'", "ab'cdefg",
"ab'cdef'g", "ab'cdef'g'", "ab'cde'fg", "ab'cde'fg'", "ab'cde'f'g'",
"ab'cd'efg", "ab'cd'efg'", "ab'cd'ef'g'", "ab'cd'e'fg'", "ab'cd'e'f'g",
"ab'cd'e'f'g'", "ab'c'defg", "ab'c'defg'", "ab'c'def'g", "ab'c'def'g'",
"ab'c'de'fg", "ab'c'de'f'g", "ab'c'de'f'g'", "ab'c'd'efg'", "ab'c'd'ef'g",
"ab'c'd'e'fg", "ab'c'd'e'fg'", "ab'c'd'e'f'g", "a'bcdefg'", "a'bcde'fg",
"a'bcde'fg'", "a'bcde'f'g", "a'bcde'f'g'", "a'bcd'ef'g", "a'bcd'ef'g'",
"a'bcd'e'f'g", "a'bcd'e'f'g'", "a'bc'defg", "a'bc'defg'", "a'bc'def'g'",
"a'bc'de'fg", "a'bc'de'f'g", "a'bc'de'f'g'", "a'bc'd'efg'", "a'bc'd'e'fg",
"a'bc'd'e'f'g", "a'b'cdefg'", "a'b'cde'fg", "a'b'cde'fg'", "a'b'cde'f'g",
"a'b'cde'f'g'", "a'b'cd'efg", "a'b'cd'ef'g'", "a'b'cd'e'fg",
"a'b'cd'e'f'g", "a'b'cd'e'f'g'", "a'b'c'defg", "a'b'c'def'g",
"a'b'c'def'g'", "a'b'c'de'f'g", "a'b'c'de'f'g'", "a'b'c'd'efg",
"a'b'c'd'efg'", "a'b'c'd'ef'g", "a'b'c'd'ef'g'", "a'b'c'd'e'fg",
"a'b'c'd'e'fg'", "a'b'c'd'e'f'g"]

Output:
["a'ce'", "abdg'", "b'df'g'", "a'c'eg'", "b'deg", "b'efg", "a'ce'",
"b'c'g", "aef'g", "c'efg'", "a'e'g", "ac'd", "c'd'fg'", "be'f'", "ac'd",
"c'd'f'g", "ac'd", "b'c'g"]

Input:
func_TRUE = ["a'bc'd'e'f'gh", "ab'cdefg'h'", "a'bc'd'ef'g'h'",
"abcde'f'gh'", "a'b'cdef'g'h", "ab'c'def'gh", "ab'c'd'ef'gh",
"a'bcdefgh'", "a'bcd'e'fgh", "a'b'cd'e'fg'h'", "a'b'cd'e'f'gh'",
"a'bcd'e'f'g'h", "ab'c'de'f'gh'", "a'bc'd'efgh", "ab'cde'fgh",
```

```
"abcd'efg'h'", "a'bc'de'fg'h'", "a'b'cd'e'f'gh", "abcde'fg'h",
"ab'cdef'g'h'"]
func_DC = ["abcdefgh", "abcdef'gh", "abcdef'gh'", "abcde'fgh'",
"abcde'fg'h'", "abcde'f'gh", "abcd'ef'gh'", "abcd'ef'g'h", "abcd'e'fgh'",
"abcd'e'fg'h'", "abcd'e'f'gh'", "abcd'e'f'g'h", "abcd'e'f'g'h'",
"abc'defg'h", "abc'defg'h'", "abc'def'gh'", "abc'def'g'h'", "abc'de'f'gh",
"abc'de'f'gh'", "abc'de'f'g'h'", "abc'd'efgh'", "abc'd'efg'h",
"abc'd'ef'gh", "abc'd'e'fgh", "abc'd'e'fg'h", "abc'd'e'f'gh'",
"abc'd'e'f'g'h", "ab'cdef'g'h", "ab'cde'fg'h", "ab'cde'f'gh'",
"ab'cd'efgh", "ab'cd'efgh'", "ab'cd'ef'g'h'", "ab'cd'e'fg'h'",
"ab'cd'e'f'gh'", "ab'c'defg'h'", "ab'c'def'g'h'", "ab'c'de'fgh'",
"ab'c'de'fg'h", "ab'c'de'f'g'h", "ab'c'd'efgh", "ab'c'd'efgh'",
"ab'c'd'e'fgh", "ab'c'd'e'fgh'", "ab'c'd'e'f'gh'", "ab'c'd'e'f'g'h'",
"a'bcdefgh", "a'bcdefg'h", "a'bcdef'gh'", "a'bcdef'g'h'", "a'bcde'fgh",
"a'bcde'fgh'", "a'bcde'fg'h'", "a'bcde'f'gh", "a'bcde'f'gh'",
"a'bcde'f'g'h", "a'bcd'efgh", "a'bcd'ef'g'h", "a'bcd'e'fgh'",
"a'bc'defgh", "a'bc'def'gh", "a'bc'def'gh'", "a'bc'de'f'gh'",
"a'bc'de'f'g'h", "a'bc'd'efg'h'", "a'bc'd'ef'gh", "a'bc'd'ef'g'h",
"a'bc'd'e'fgh'", "a'bc'd'e'fg'h", "a'bc'd'e'fg'h'", "a'bc'd'e'f'g'h'",
"a'b'cdef'gh'", "a'b'cdef'g'h'", "a'b'cde'f'g'h'", "a'b'cd'ef'gh",
"a'b'cd'ef'gh'", "a'b'cd'e'fgh", "a'b'cd'e'fgh'", "a'b'cd'e'fg'h",
"a'b'c'defgh", "a'b'c'defg'h", "a'b'c'def'gh", "a'b'c'def'g'h'",
"a'b'c'de'fgh", "a'b'c'de'fgh'", "a'b'c'de'f'g'h'", "a'b'c'd'efgh",
"a'b'c'd'ef'g'h", "a'b'c'd'e'fgh'", "a'b'c'd'e'fg'h"]

Output:
["a'bc'd'f'gh", "ab'deg'h'", "a'bc'd'g'h'", "bdf'gh'", "b'cdef'g'",
"b'c'def'gh", "ac'd'ef'gh", "a'bcdgh'", "a'cd'e'fg", "a'b'cd'e'f",
"a'b'cd'f'g", "bcd'f'g'h", "ae'f'gh'", "a'c'efgh", "ab'cde'fh",
"abcd'fg'h'", "a'bde'fg'h'", "a'b'cd'f'g", "acde'fg'h", "b'def'g'h'"]

Input:
func_TRUE = ["abc'de'f'g'h'", "a'bcde'f'g'h'", "a'bc'd'e'f'gh'",
"a'b'c'd'efg'h", "ab'cde'f'gh", "abcd'e'f'gh'", "ab'c'd'ef'g'h",
"ab'c'd'efgh'", "abc'def'gh'", "a'bcdefgh'", "ab'cd'e'f'g'h'",
"abcd'e'fgh", "a'b'c'd'ef'gh", "abcde'f'gh", "a'b'cdef'gh",
"abc'd'e'f'g'h", "abc'd'ef'gh", "a'bc'd'efg'h'", "abcd'e'fg'h",
"ab'c'de'fg'h'", "a'bcd'e'f'gh'", "a'b'cde'f'g'h", "ab'c'def'gh",
"ab'cdefgh", "a'bc'de'fgh'", "abcd'efgh'", "a'b'c'defg'h", "ab'cdefg'h",
"a'bc'defg'h'", "abc'd'efgh", "ab'cd'e'fgh", "abc'd'ef'g'h'",
```

```
"abc'def'g'h", "a'bcdefg'h'", "a'b'c'de'fgh", "a'b'cde'fg'h",
"a'b'cd'ef'gh'", "a'bcd'e'fg'h", "ab'cd'ef'g'h", "abcde'fgh"]
func_DC = ["abcdefgh", "abcdefg'h", "abcdefg'h'", "abcdef'gh'",
"abcdef'g'h", "abcde'fg'h", "abcde'f'gh'", "abcde'f'g'h", "abcde'f'g'h'",
"abcd'efgh", "abcd'efg'h", "abcd'efg'h'", "abcd'ef'gh", "abcd'ef'gh'",
"abcd'e'fg'h'", "abcd'e'f'gh", "abcd'e'f'g'h", "abc'defgh'", "abc'defg'h",
"abc'defg'h'", "abc'def'gh", "abc'def'g'h'", "abc'de'fgh", "abc'de'fgh'",
"abc'de'fg'h", "abc'de'fg'h'", "abc'de'f'gh", "abc'de'f'g'h",
"abc'd'efgh'", "abc'd'efg'h'", "abc'd'ef'gh'", "abc'd'e'fg'h",
"abc'd'e'fg'h'", "abc'd'e'f'gh", "abc'd'e'f'gh'", "ab'cdefgh'",
"ab'cdefg'h'", "ab'cdef'gh", "ab'cdef'g'h", "ab'cdef'g'h'", "ab'cde'fgh'",
"ab'cde'f'gh'", "ab'cde'f'g'h'", "ab'cd'efgh", "ab'cd'efg'h'",
"ab'cd'ef'gh", "ab'cd'ef'gh'", "ab'cd'ef'g'h'", "ab'cd'e'fg'h",
"ab'cd'e'fg'h'", "ab'cd'e'f'gh'", "ab'cd'e'f'g'h", "ab'c'defgh",
"ab'c'defg'h", "ab'c'defg'h'", "ab'c'def'gh'", "ab'c'def'g'h",
"ab'c'de'fgh", "ab'c'de'fgh'", "ab'c'de'f'gh'", "ab'c'd'efgh",
"ab'c'd'efg'h'", "ab'c'd'ef'gh", "ab'c'd'ef'gh'", "ab'c'd'ef'g'h'",
"ab'c'd'e'fgh", "ab'c'd'e'fgh'", "ab'c'd'e'fg'h'", "ab'c'd'e'f'g'h'",
"a'bcdefg'h", "a'bcdef'g'h'", "a'bcde'fgh", "a'bcde'fgh'", "a'bcde'fg'h",
"a'bcde'f'g'h", "a'bcd'efgh", "a'bcd'efgh'", "a'bcd'efg'h",
"a'bcd'efg'h'", "a'bcd'ef'gh'", "a'bcd'ef'g'h'", "a'bcd'e'fgh",
"a'bcd'e'fgh'", "a'bcd'e'fg'h'", "a'bcd'e'f'gh", "a'bcd'e'f'g'h",
"a'bcd'e'f'g'h'", "a'bc'defgh'", "a'bc'def'gh", "a'bc'def'g'h",
"a'bc'def'g'h'", "a'bc'de'fgh", "a'bc'de'fg'h", "a'bc'de'fg'h'",
"a'bc'de'f'g'h", "a'bc'de'f'g'h'", "a'bc'd'efgh", "a'bc'd'efgh'",
"a'bc'd'efg'h", "a'bc'd'ef'gh", "a'bc'd'ef'gh'", "a'bc'd'ef'g'h",
"a'bc'd'ef'g'h'", "a'bc'd'e'fgh", "a'bc'd'e'fgh'", "a'bc'd'e'fg'h",
"a'bc'd'e'fg'h'", "a'bc'd'e'f'gh", "a'bc'd'e'f'g'h", "a'bc'd'e'f'g'h'",
"a'b'cdefgh", "a'b'cdefgh'", "a'b'cdef'gh'", "a'b'cdef'g'h",
"a'b'cdef'g'h'", "a'b'cde'fgh", "a'b'cde'f'gh", "a'b'cde'f'gh'",
"a'b'cde'f'g'h'", "a'b'cd'efgh", "a'b'cd'efg'h", "a'b'cd'efg'h'",
"a'b'cd'ef'gh", "a'b'cd'ef'g'h", "a'b'cd'ef'g'h'", "a'b'cd'e'fgh",
"a'b'cd'e'fgh'", "a'b'cd'e'fg'h", "a'b'cd'e'fg'h'", "a'b'cd'e'f'gh",
"a'b'cd'e'f'g'h", "a'b'cd'e'f'g'h'", "a'b'c'defgh", "a'b'c'defgh'",
"a'b'c'defg'h'", "a'b'c'def'gh", "a'b'c'def'gh'", "a'b'c'def'g'h'",
"a'b'c'de'fgh'", "a'b'c'de'fg'h'", "a'b'c'de'f'gh'", "a'b'c'de'f'g'h",
"a'b'c'd'efgh", "a'b'c'd'ef'gh'", "a'b'c'd'e'fgh", "a'b'c'd'e'fgh'",
"a'b'c'd'e'fg'h", "a'b'c'd'e'fg'h'", "a'b'c'd'e'f'gh", "a'b'c'd'e'f'g'h'"]

Output:
```

```
["bde'f'g'", "bde'f'g'", "a'bd'h'", "a'd'fh", "b'cdf'gh", "bd'f'gh'",
"ab'ef'h", "ac'd'eh'", "ac'ef'g", "a'bfgh'", "b'd'e'g'h'", "cd'fgh",
"b'egh", "abe'f'h", "b'egh", "be'g'h", "c'ef'gh", "a'bd'h'", "be'g'h",
"c'e'fg'h'", "a'bd'h'", "a'ce'g'h", "b'egh", "b'egh", "c'de'fh'",
"bd'egh'", "b'c'defh", "adeg'h", "c'dfg'h'", "d'efgh", "cd'fgh",
"bc'eg'h'", "bc'df'g'", "befg'h'", "c'de'fg", "a'ce'g'h", "d'ef'gh'",
"be'g'h", "b'cef'h", "bde'fh"]


Input:
func_TRUE = ["a'bcd'ef'ghi", "a'b'cdef'gh'i'", "abcde'fghi'",
"abcd'ef'gh'i", "a'b'c'def'g'h'i'", "a'b'cde'fg'h'i'", "ab'cd'efgh'i'",
"abc'd'efgh'i", "abc'd'e'f'g'hi", "a'bcd'ef'ghi'", "ab'c'de'f'ghi'",
"a'bcd'ef'gh'i'", "a'b'cd'e'fg'hi'", "ab'cd'e'f'gh'i'",
"a'b'cd'e'f'g'h'i", "abcd'ef'gh'i'", "a'b'cd'ef'g'h'i'", "abcd'efg'h'i'",
"a'bcd'e'fg'hi'", "a'b'cde'f'g'h'i'", "ab'cd'efghi'", "a'bc'd'ef'gh'i'",
"a'bc'def'ghi", "abc'd'e'f'g'hi'", "a'b'c'd'e'f'g'hi", "a'b'c'de'fg'h'i",
"abc'de'fghi", "ab'cd'e'fg'hi", "a'b'cd'e'fg'hi", "ab'cde'f'gh'i'",
"a'bc'd'e'f'gh'i'", "a'b'cd'e'f'gh'i", "a'b'c'de'fg'h'i'",
"a'b'cd'efg'h'i", "ab'cd'e'f'g'hi", "ab'c'd'ef'g'hi", "a'bcde'f'g'h'i",
"a'bc'd'e'f'gh'i", "a'b'c'def'ghi", "abcdef'g'hi'", "a'bc'de'fg'hi'",
"ab'c'd'e'fg'h'i", "a'bc'd'efg'h'i'", "a'bcde'fghi'", "abc'de'f'g'h'i",
"a'bcde'fg'hi", "ab'cd'e'f'g'hi'", "a'bcd'e'f'g'h'i'",
"a'b'c'd'e'f'g'h'i", "ab'c'd'efghi'"]
func_DC = ["abcdefghi", "abcdefghi'", "abcdefgh'i", "abcdefgh'i'",
"abcdefg'hi", "abcdefg'hi'", "abcdefg'h'i", "abcdefg'h'i'", "abcdef'ghi",
"abcdef'ghi'", "abcdef'gh'i", "abcdef'gh'i'", "abcdef'g'hi",
"abcdef'g'h'i", "abcdef'g'h'i'", "abcde'fghi", "abcde'fgh'i",
"abcde'fgh'i'", "abcde'fg'hi", "abcde'fg'hi'", "abcde'fg'h'i",
"abcde'fg'h'i'", "abcde'f'ghi", "abcde'f'ghi'", "abcde'f'gh'i",
"abcde'f'gh'i'", "abcde'f'g'hi", "abcde'f'g'hi'", "abcde'f'g'h'i",
"abcde'f'g'h'i'", "abcd'efghi", "abcd'efghi'", "abcd'efgh'i",
"abcd'efgh'i'", "abcd'efg'hi", "abcd'efg'hi'", "abcd'efg'h'i",
"abcd'ef'ghi", "abcd'ef'ghi'", "abcd'ef'g'hi", "abcd'ef'g'hi'",
"abcd'ef'g'h'i", "abcd'ef'g'h'i'", "abcd'e'fghi", "abcd'e'fghi'",
"abcd'e'fgh'i", "abcd'e'fgh'i'", "abcd'e'fg'hi", "abcd'e'fg'hi'",
"abcd'e'fg'h'i", "abcd'e'fg'h'i'", "abcd'e'f'ghi", "abcd'e'f'ghi'",
"abcd'e'f'gh'i", "abcd'e'f'gh'i'", "abcd'e'f'g'hi", "abcd'e'f'g'hi'",
"abcd'e'f'g'h'i", "abcd'e'f'g'h'i'", "abc'defghi", "abc'defghi'",
"abc'defgh'i", "abc'defgh'i'", "abc'defg'hi", "abc'defg'hi'",
```

```
"abc'defg'h'i", "abc'def'ghi", "abc'def'ghi'", "abc'def'gh'i",
"abc'def'gh'i'", "abc'def'g'hi", "abc'def'g'hi'", "abc'def'g'h'i",
"abc'def'g'h'i'", "abc'de'fghi'", "abc'de'fgh'i", "abc'de'fgh'i'",
"abc'de'fg'hi", "abc'de'fg'hi'", "abc'de'fg'h'i", "abc'de'fg'h'i'",
"abc'de'f'ghi", "abc'de'f'ghi'", "abc'de'f'gh'i", "abc'de'f'gh'i'",
"abc'de'f'g'hi", "abc'de'f'g'hi'", "abc'de'f'g'h'i'", "abc'd'efghi",
"abc'd'efghi'", "abc'd'efgh'i'", "abc'd'efg'hi", "abc'd'efg'hi'",
"abc'd'efg'h'i", "abc'd'efg'h'i'", "abc'd'ef'ghi", "abc'd'ef'ghi'",
"abc'd'ef'gh'i", "abc'd'ef'gh'i'", "abc'd'ef'g'hi", "abc'd'ef'g'hi'",
"abc'd'ef'g'h'i", "abc'd'ef'g'h'i'", "abc'd'e'fghi", "abc'd'e'fghi'",
"abc'd'e'fgh'i", "abc'd'e'fgh'i'", "abc'd'e'fg'hi", "abc'd'e'fg'h'i",
"abc'd'e'fg'h'i'", "abc'd'e'f'ghi", "abc'd'e'f'ghi'", "abc'd'e'f'gh'i",
"abc'd'e'f'gh'i'", "abc'd'e'f'g'h'i", "abc'd'e'f'g'h'i'", "ab'cdefghi",
"ab'cdefghi'", "ab'cdefgh'i", "ab'cdefgh'i'", "ab'cdefg'hi",
"ab'cdefg'hi'", "ab'cdefg'h'i", "ab'cdef'ghi", "ab'cdef'ghi'",
"ab'cdef'gh'i", "ab'cdef'gh'i'", "ab'cdef'g'hi", "ab'cdef'g'hi'",
"ab'cdef'g'h'i", "ab'cdef'g'h'i'", "ab'cde'fghi", "ab'cde'fghi'",
"ab'cde'fgh'i", "ab'cde'fgh'i'", "ab'cde'fg'hi", "ab'cde'fg'hi'",
"ab'cde'fg'h'i", "ab'cde'fg'h'i'", "ab'cde'f'ghi", "ab'cde'f'ghi'",
"ab'cde'f'gh'i", "ab'cde'f'g'hi", "ab'cde'f'g'hi'", "ab'cde'f'g'h'i",
"ab'cde'f'g'h'i'", "ab'cd'efghi", "ab'cd'efgh'i", "ab'cd'efg'hi",
"ab'cd'efg'hi'", "ab'cd'efg'h'i", "ab'cd'efg'h'i'", "ab'cd'ef'ghi",
"ab'cd'ef'ghi'", "ab'cd'ef'gh'i", "ab'cd'ef'gh'i'", "ab'cd'ef'g'hi",
"ab'cd'ef'g'hi'", "ab'cd'ef'g'h'i", "ab'cd'ef'g'h'i'", "ab'cd'e'fghi",
"ab'cd'e'fghi'", "ab'cd'e'fgh'i", "ab'cd'e'fgh'i'", "ab'cd'e'fg'hi'",
"ab'cd'e'fg'h'i", "ab'cd'e'fg'h'i'", "ab'cd'e'f'ghi", "ab'cd'e'f'ghi'",
"ab'cd'e'f'gh'i", "ab'cd'e'f'g'h'i", "ab'cd'e'f'g'h'i'", "ab'c'defghi",
"ab'c'defghi'", "ab'c'defgh'i", "ab'c'defgh'i'", "ab'c'defg'hi",
"ab'c'defg'hi'", "ab'c'defg'h'i", "ab'c'defg'h'i'", "ab'c'def'ghi",
"ab'c'def'ghi'", "ab'c'def'gh'i", "ab'c'def'gh'i'", "ab'c'def'g'hi",
"ab'c'def'g'hi'", "ab'c'def'g'h'i", "ab'c'def'g'h'i'", "ab'c'de'fghi",
"ab'c'de'fghi'", "ab'c'de'fgh'i", "ab'c'de'fgh'i'", "ab'c'de'fg'hi",
"ab'c'de'fg'hi'", "ab'c'de'fg'h'i", "ab'c'de'f'ghi", "ab'c'de'f'gh'i",
"ab'c'de'f'gh'i'", "ab'c'de'f'g'hi", "ab'c'de'f'g'h'i",
"ab'c'de'f'g'h'i'", "ab'c'd'efghi", "ab'c'd'efgh'i'", "ab'c'd'efg'hi",
"ab'c'd'efg'hi'", "ab'c'd'efg'h'i", "ab'c'd'efg'h'i'", "ab'c'd'ef'ghi",
"ab'c'd'ef'ghi'", "ab'c'd'ef'gh'i", "ab'c'd'ef'gh'i'", "ab'c'd'ef'g'hi'",
"ab'c'd'ef'g'h'i", "ab'c'd'ef'g'h'i'", "ab'c'd'e'fghi", "ab'c'd'e'fghi'",
"ab'c'd'e'fgh'i", "ab'c'd'e'fgh'i'", "ab'c'd'e'fg'hi", "ab'c'd'e'fg'hi'",
"ab'c'd'e'fg'h'i'", "ab'c'd'e'f'ghi", "ab'c'd'e'f'ghi'",
```

```
"ab'c'd'e'f'gh'i", "ab'c'd'e'f'gh'i'", "ab'c'd'e'f'g'hi",
"ab'c'd'e'f'g'hi'", "ab'c'd'e'f'g'h'i", "ab'c'd'e'f'g'h'i'", "a'bcdefghi",
"a'bcdefghi'", "a'bcdefgh'i", "a'bcdefgh'i'", "a'bcdefg'hi",
"a'bcdefg'hi'", "a'bcdefg'h'i", "a'bcdefg'h'i'", "a'bcdef'ghi",
"a'bcdef'ghi'", "a'bcdef'gh'i", "a'bcdef'gh'i'", "a'bcdef'g'hi",
"a'bcdef'g'hi'", "a'bcdef'g'h'i", "a'bcdef'g'h'i'", "a'bcde'fghi",
"a'bcde'fgh'i", "a'bcde'fgh'i'", "a'bcde'fg'hi'", "a'bcde'fg'h'i",
"a'bcde'fg'h'i'", "a'bcde'f'ghi", "a'bcde'f'ghi'", "a'bcde'f'gh'i",
"a'bcde'f'gh'i'", "a'bcde'f'g'hi", "a'bcde'f'g'hi'", "a'bcde'f'g'h'i'",
"a'bcd'efghi", "a'bcd'efghi'", "a'bcd'efgh'i", "a'bcd'efgh'i'",
"a'bcd'efg'hi", "a'bcd'efg'hi'", "a'bcd'efg'h'i", "a'bcd'efg'h'i'",
"a'bcd'ef'gh'i", "a'bcd'ef'g'hi", "a'bcd'ef'g'hi'", "a'bcd'ef'g'h'i",
"a'bcd'ef'g'h'i'", "a'bcd'e'fghi", "a'bcd'e'fghi'", "a'bcd'e'fgh'i",
"a'bcd'e'fgh'i'", "a'bcd'e'fg'hi", "a'bcd'e'fg'h'i", "a'bcd'e'fg'h'i'",
"a'bcd'e'f'ghi", "a'bcd'e'f'ghi'", "a'bcd'e'f'gh'i", "a'bcd'e'f'gh'i'",
"a'bcd'e'f'g'hi", "a'bcd'e'f'g'hi'", "a'bcd'e'f'g'h'i", "a'bc'defghi",
"a'bc'defghi'", "a'bc'defgh'i", "a'bc'defgh'i'", "a'bc'defg'hi",
"a'bc'defg'hi'", "a'bc'defg'h'i", "a'bc'defg'h'i'", "a'bc'def'ghi'",
"a'bc'def'gh'i", "a'bc'def'gh'i'", "a'bc'def'g'hi", "a'bc'def'g'hi'",
"a'bc'def'g'h'i", "a'bc'def'g'h'i'", "a'bc'de'fghi", "a'bc'de'fghi'",
"a'bc'de'fgh'i", "a'bc'de'fgh'i'", "a'bc'de'fg'hi", "a'bc'de'fg'h'i",
"a'bc'de'fg'h'i'", "a'bc'de'f'ghi", "a'bc'de'f'ghi'", "a'bc'de'f'gh'i",
"a'bc'de'f'gh'i'", "a'bc'de'f'g'hi", "a'bc'de'f'g'hi'", "a'bc'de'f'g'h'i",
"a'bc'de'f'g'h'i'", "a'bc'd'efghi", "a'bc'd'efghi'", "a'bc'd'efgh'i",
"a'bc'd'efgh'i'", "a'bc'd'efg'hi", "a'bc'd'efg'hi'", "a'bc'd'efg'h'i",
"a'bc'd'ef'ghi", "a'bc'd'ef'ghi'", "a'bc'd'ef'gh'i", "a'bc'd'ef'g'hi",
"a'bc'd'ef'g'hi'", "a'bc'd'ef'g'h'i", "a'bc'd'ef'g'h'i'", "a'bc'd'e'fghi",
"a'bc'd'e'fghi'", "a'bc'd'e'fgh'i", "a'bc'd'e'fgh'i'", "a'bc'd'e'fg'hi",
"a'bc'd'e'fg'hi'", "a'bc'd'e'fg'h'i", "a'bc'd'e'fg'h'i'",
"a'bc'd'e'f'ghi", "a'bc'd'e'f'g'hi", "a'bc'd'e'f'g'hi'",
"a'bc'd'e'f'g'h'i", "a'bc'd'e'f'g'h'i'", "a'b'cdefghi", "a'b'cdefghi'",
"a'b'cdefgh'i", "a'b'cdefgh'i'", "a'b'cdefg'hi", "a'b'cdefg'hi'",
"a'b'cdefg'h'i", "a'b'cdefg'h'i'", "a'b'cdef'ghi", "a'b'cdef'ghi'",
"a'b'cdef'gh'i", "a'b'cdef'g'hi", "a'b'cdef'g'hi'", "a'b'cdef'g'h'i'",
"a'b'cde'fghi", "a'b'cde'fghi'", "a'b'cde'fgh'i", "a'b'cde'fgh'i'",
"a'b'cde'fg'hi", "a'b'cde'fg'hi'", "a'b'cde'fg'h'i", "a'b'cde'f'ghi",
"a'b'cde'f'ghi'", "a'b'cde'f'gh'i", "a'b'cde'f'gh'i'", "a'b'cde'f'g'hi",
"a'b'cde'f'g'hi'", "a'b'cde'f'g'h'i", "a'b'cd'efghi", "a'b'cd'efghi'",
"a'b'cd'efgh'i", "a'b'cd'efgh'i'", "a'b'cd'efg'hi", "a'b'cd'efg'hi'",
"a'b'cd'efg'h'i'", "a'b'cd'ef'ghi", "a'b'cd'ef'ghi'", "a'b'cd'ef'gh'i",
```

```
"a'b'cd'ef'gh'i'", "a'b'cd'ef'g'hi", "a'b'cd'ef'g'h'i", "a'b'cd'e'fghi",
"a'b'cd'e'fghi'", "a'b'cd'e'fgh'i", "a'b'cd'e'fgh'i'", "a'b'cd'e'fg'h'i",
"a'b'cd'e'fg'h'i'", "a'b'cd'e'f'ghi", "a'b'cd'e'f'ghi'",
"a'b'cd'e'f'gh'i'", "a'b'cd'e'f'g'hi", "a'b'cd'e'f'g'h'i'",
"a'b'c'defghi", "a'b'c'defghi'", "a'b'c'defgh'i", "a'b'c'defgh'i'",
"a'b'c'defg'hi", "a'b'c'defg'hi'", "a'b'c'defg'h'i", "a'b'c'def'ghi'",
"a'b'c'def'gh'i", "a'b'c'def'gh'i'", "a'b'c'def'g'hi'", "a'b'c'def'g'h'i",
"a'b'c'de'fghi", "a'b'c'de'fghi'", "a'b'c'de'fgh'i", "a'b'c'de'fgh'i'",
"a'b'c'de'fg'hi", "a'b'c'de'fg'hi'", "a'b'c'de'f'ghi", "a'b'c'de'f'ghi'",
"a'b'c'de'f'gh'i", "a'b'c'de'f'gh'i'", "a'b'c'de'f'g'hi",
"a'b'c'de'f'g'hi'", "a'b'c'de'f'g'h'i", "a'b'c'de'f'g'h'i'",
"a'b'c'd'efghi", "a'b'c'd'efghi'", "a'b'c'd'efgh'i", "a'b'c'd'efgh'i'",
"a'b'c'd'efg'hi", "a'b'c'd'efg'hi'", "a'b'c'd'efg'h'i",
"a'b'c'd'efg'h'i'", "a'b'c'd'ef'ghi", "a'b'c'd'ef'ghi'",
"a'b'c'd'ef'gh'i", "a'b'c'd'ef'gh'i'", "a'b'c'd'ef'g'hi",
"a'b'c'd'ef'g'hi'", "a'b'c'd'ef'g'h'i", "a'b'c'd'ef'g'h'i'",
"a'b'c'd'e'fghi", "a'b'c'd'e'fghi'", "a'b'c'd'e'fgh'i",
"a'b'c'd'e'fgh'i'", "a'b'c'd'e'fg'hi", "a'b'c'd'e'fg'hi'",
"a'b'c'd'e'fg'h'i", "a'b'c'd'e'fg'h'i'", "a'b'c'd'e'f'ghi",
"a'b'c'd'e'f'ghi'", "a'b'c'd'e'f'gh'i", "a'b'c'd'e'f'gh'i'",
"a'b'c'd'e'f'g'hi'", "a'b'c'd'e'f'g'h'i'"]

Output:
["cg", "dg", "dg", "cg", "f'h'i'", "e'h'i'", "cg", "bi", "bi", "cg", "dg",
"cg", "cfh", "cg", "e'f'i", "cg", "f'h'i'", "bc", "bc", "f'h'i'", "cg",
"gh'i'", "dg", "bf'g'", "e'hi", "a'fi", "dg", "fhi", "fhi", "dg", "gh'i'",
"cg", "e'h'i'", "d'g'i", "e'hi", "d'hi", "bi", "bi", "dg", "bc", "dfh",
"d'g'i", "d'h'i'", "dg", "bi", "bi", "ad'f'", "bc", "e'f'i", "fgi'"]
```