CODERPAD CODE:

```
/*
You're given a vector of vectors of words, e.g.:
[['quick', 'lazy'], ['brown', 'black', 'grey'], ['fox', 'dog']].

Write a generalized function that prints all combinations of one word from the first vector, one word
from the second vector, etc.
The solution may not use recursion.
NOTE: the number of vectors and number of elements within each vector may vary.

For the input above, it should print (in any order):
quick brown fox
quick brown dog
quick black fox
quick black dog
...
lazy grey dog
*/
```

```
import java.io.*;
import java.util.*;

/*
 * To execute Java, please define "static void main" on a class
 * named Solution.
 *
 * If you need more classes, simply define them inline.
 */

/*class Solution {
  public static void main(String[] args) {
    ArrayList<String> strings = new ArrayList<String>();
    strings.add("Hello, World!");
    strings.add("Welcome to CoderPad.");
    strings.add("This pad is running Java 8.");

    for (String string : strings) {
      System.out.println(string);
    }
  }
```

```
}

*/

import java.util.*;

public class Solution {


 //this is the generalized function for the required Output
  public void finalAns(Vector<Vector<String>> input)
  {
    int totalCount=1;//stores the number of total possible combinations
    int noElementsInput=0;//stores the number of vectors the given input has
    for(Vector<String> i:input)
    {
       totalCount=totalCount*i.size();
       noElementsInput=noElementsInput+1;
    }
    //ArrayList of final output of combination of words
    ArrayList<ArrayList<String>> output=new ArrayList<ArrayList<String>>();
    output.add(firstRepeatArray(input.get(0), totalCount));
     int betweenNo=1;
    int x=noElementsInput;
    while(betweenNo<=noElementsInput-2)
    {
      output.add(repeatArray(input.get(betweenNo), totalCount,input.get(x-1).size()));
      betweenNo++;
      x=x-1;
    }
    output.add(lastRepeatArray((input.get(input.size()-1)),totalCount));
    display(output,totalCount);

  }

  //Creates arrayList taking the first vector<String> of input based on a pattern
  public static ArrayList<String> firstRepeatArray(Vector<String> v,int totalCount)
  {
    ArrayList<String> ans=new ArrayList<String>();
    int c=0;
    for(int i=0;i<totalCount;)
    {

      for(int j=0;j<(totalCount/v.size());j++)
       {
         ans.add(i, v.get(c));
         i++;
```

```
     }
    c++;

  }
  return ans;
 }
//Creates arrayList taking the last vector<String> of input based on a pattern
 public static ArrayList<String> lastRepeatArray(Vector<String> v,int totalCount)
 {
  ArrayList<String> ans=new ArrayList<String>();
  int vecSize=v.size();
  int c=0;
  for(int i=0;i<totalCount;)
  {
   for(int j=0;j<vecSize;j++)
   {
    ans.add(i, v.get(c));
    i++;
    c++;
   }

    c=0;
  }
  return ans;
 }
//Creates arrayList taking the intermediate vector<String> of input based on a pattern
 public static ArrayList<String> repeatArray(Vector<String> v,int totalCount,int x)
 {
  ArrayList<String> ans=new ArrayList<String>();
  int vecSize=v.size();
  int c=0;
  for(int i=0;i<totalCount;)
  {
   for(int j=0;j<x;j++)
   {
    ans.add(i, v.get(c));
    i++;
   }
   c=c+1;
   if(c>=vecSize)
    c=0;
  }
  return ans;
 }
//Function to display the final output
 public static void display(ArrayList<ArrayList<String>> l,int totalCount)
{
  for(int i=0;i<totalCount;i++)
```

```
  {
    for(ArrayList<String> list:l)
    {
      System.out.print(list.get(i)+" ");
    }
    System.out.println();

  }
 }

 public static void main(String[] args)
 {
   Solution f=new Solution();
  Vector<Vector<String>> input=new Vector<Vector<String>>();
  Vector<String> v1=new Vector<String>();
  Vector<String> v2=new Vector<String>();
  Vector<String> v3=new Vector<String>();
  //Vector<String> v4=new Vector<String>();
 v1.add(0, "quick");
  v1.add(1,"lazy");
  v2.add(0,"brown");
  v2.add(1,"black");
  v2.add(2,"grey");
  //v4.add(0,"Kusum");
  //v4.add(1,"Rudrayya);
  v3.add(0,"fox");
  v3.add(1,"dog");
  input.add(v1);
  input.add(v2);
  //input.add(v4);
  input.add(v3);

  f.finalAns(input);

 }
}
```