

Aplicaciones de Permutaciones y Combinaciones.

Revisar la web, libros, publicaciones y otros, buscar problemas propuestos sobre Combinaciones y Permutaciones con y sin repeticiones de elementos. Aplicar los algoritmos de `combinacionesSR()`, `combinacionesCR()`, `permutacionesSR()`, `permutacionesCR()` o modificarlos para dar solución a los problemas propuestos en esas fuentes de información. Aplicar al menos 1 problemas cualesquiera para cada Algoritmo, por cada integrante de grupo. Citar Fuentes.

I. Combinaciones sin Repeticiones

Ejercicio 1:

Este programa realiza una Lista de Listas con todas las combinaciones sin repetición en una Lista de n elementos.

```
public static void allPermutations(LinkedList<Integer> L,
    LinkedList<LinkedList<Integer>> ANS){

    for(int i=0;i<L.size();++i){
        LinkedList<Integer> L2 = new LinkedList<>();
        combiSR(L,L2,ANS,i+1,0);
    }

    System.out.println(ANS);
}

public static void combiSR(LinkedList<Integer> L1,
    LinkedList<Integer> L2,
    LinkedList<LinkedList<Integer>> ANS,int r, int i) {
    if (L2.size() == r) {

        ANS.add((LinkedList<Integer>)L2.clone());
        //System.out.println(L2);
        return;
    }
    int k = i;
    while (k < L1.size()) {
        L2.add(L1.get(k));
        combiSR(L1, L2,ANS, r, k + 1);
        L2.removeLast();
        k = k + 1;
    }
}
```

Ejercicio 2:

Un chef va a preparar una ensalada de verduras con tomate, zanahoria, papa y brócoli. ¿De cuántas formas se puede preparar la ensalada usando solo 2 ingredientes?

Referencia(s)

<https://matemovil.com/permutaciones-y-combinaciones>

```
public static void ensaladaSR(LinkedList<String> L1,
LinkedList<String> L2, int cantidad, int n, int inicio) {
    if (L2.size() == n) {
        System.out.println(L2);
        c++;
        return;
    }
    int k = inicio;
    while (k < cantidad) {
        L2.add(L1.get(k));
        ensaladaSR(L1, L2, cantidad, n, k + 1);
        L2.removeLast();
        k = k + 1;
    }
}
```