

1- Recursión (40%)

Dada una lista de n letras distintas, escribir un programa en Scala usando recursión que enumere todos los subconjuntos de estas letras (sin invertir letras). No es necesario enumerar el conjunto vacío. Ejm. List('a','b','c') retorna **EN UNA LISTA** a ab ac abc b bc c

Probar con val lista1 : List[Char] = List('a','b','c','d','e')

2- Recursión(20%)

Se tienen 5 estudiantes sentados de izquierda a derecha:

"Juan","Anamaria","Pablo","Sofía","Alejandro". Escribir un programa en Scala que haga uso de la recursión mostrando todas las posibles posiciones en las que se pueden sentar los 5

estudiantes **EN UNA LISTA**.

3- Recursión(40%)

Dada la siguiente sucesión:

$$a_{n+1} = \frac{a_n^2 + a_n - 1}{a_n + c}; c \in \mathbb{C}$$

donde el primer elemento es 0, muestre los n primeros números usando recursión simple y **tail-recursion**

(recursión de cola). Ejemplo: Para n = 6 y c = 0.5 el resultado debe quedar en UNA LISTA:

List(8.374131132596311, 8.020830448762329, 7.67375886524823, 7.333333333333336, -0.6666666666666666, -2.0, 0.0)

Adicionalmente, debe mostrar el tiempo de ejecución para c/u cuando n=100.