

Tarea 1

```
def generate_palindromes_recursive(alphabet, n):
    def helper(current_length):
        # Caso base: Longitud 0 → Devuelve una lista con la palabra vacía
        if current_length == 0:
            return [""]

        # Caso base: Longitud 1 → Cada carácter del alfabeto es un palíndromo
        if current_length == 1:
            return alphabet

        # Genera palíndromos para la longitud actual
        smaller_palindromes = helper(current_length - 2)
        palindromes = []

        for char in alphabet:
            for middle in smaller_palindromes:
                palindromes.append(char + middle + char)

        return palindromes

    # Generar todos los palíndromos para longitudes de 0 a n
    result = []
    for length in range(n + 1):
        result.extend(helper(length))

    return result

# Entrada
alphabet = input("Ingresa el alfabeto (caracteres separados por espacios): ").split()
n = int(input("Ingresa la longitud máxima de las palabras (n): "))
```

```
# Generación de palíndromos
palindromes = generate_palindromes_recursive(alphabet, n)

# Salida
print("Palíndromos generados:")
for palindrome in palindromes:
    print(palindrome)
```