

Pseudocódigo Kmp

Función Prefija (n)

```
Vector<int> prefija (string patron) {  
    Vector<int> lista (patron.size()) // Para almacenar valores de n  
    int len = 0 // Valores que coinciden de ambos lados  
    for (int i = 0; i < patron.size(); i++) {  
        if (i == 0) lista[i] = 0 // El primer elemento de n siempre es 0  
        else {  
            // Mientras no coincidan, retroceder  
            while (len > 0 && patron[i] != patron[len]) {  
                len = lista[len-1]  
            }  
            // Si son iguales, incrementar len  
            if (patron[i] == patron[len]) {  
                len++  
            }  
            lista[i] = len // Guardar longitud  
        }  
    }  
    return lista  
}
```

Función KMP → índices de inicio : fin

```
vector<pair<int,int>> kmp (string texto, string patron) {
```

```
// Primero obtenemos la función prefijo del patron  
vector<int> lista = prefija (patron)
```

```
// Contadores para texto, patron  
int i=0, int j=0
```

```
// Indices (resultado)  
vector<pair<int,int>> indices
```

```
while (i < texto.size()) {
```

```
    if (texto[i] == patron[j+1]) { // Si j==j+1, ambos avanzan  
        i++, j++  
    }
```

```
    if (j == patron.size()-1) { // Si j==patron, regresar res  
        indices.push_back (i - patron.size()-1, i)  
        j = lista[j]  
    }
```

```
    if (texto[i] != patron[j+1]) { // regresar
```

```
        if (j==0) i++  
        else j = lista[j]
```

```
    }  
    return indices
```

```
}
```