INTERPOLACIÓN POLINÓMICA DE LAGRANGE EN R APLICADO A LA INTRODUCCIÓN DE DATOS MANUAL

El código presentado a continuación es una implementación en **R** del **método de interpolación de Lagrange**, una técnica de interpolación polinómica que permite encontrar un polinomio de grado n−1 que pasa exactamente por un conjunto de puntos dados. A continuación, se explica paso a paso lo que hace el código:

```r
L=function(x,s,n){
  L=c()
  for (i in 1:n){
    L[i]=1
    for (j in 1:n){
      if(j!=i){
        L[i]=L[i]*((x-s[j])/(s[i]-s[j]))
      }
    }
  }
  print("A continuación se muestran las bases de Lagrange para el valor deseado")
  return(L)
}
```

Primera función: L

Esta función calcula las **bases de Lagrange** Li(x). Estas bases son los términos que, al combinarse linealmente con los valores yi, producen el polinomio interpolador.

Parámetros:

- x: el valor en el que se quiere evaluar el polinomio interpolado.
- s: un vector que contiene los nodos o puntos de soportes: (los valores de x conocidos).
- n: la longitud del vector s, es decir, el número de nodos.(No haría falta pedir esta variable como parte de la función pero se ha decidido incluirla para ganar más claridad en la explicación.

Algoritmo:

1. Crea un vector vacío L para almacenar las bases de Lagrange.
2. Recorre cada índice i de 1 a n..
    - Inicializa L[i]=1
    - Para cada j distinto de i (que también va de 1 a n)
3. Devuelve el vector L, que contiene todas las bases de Lagrange evaluadas en x.

$$L_j(x) = \prod_{\substack{1 \le k \le n \\ k \neq j}} \frac{x - x_k}{x_j - x_k}.$$

NOTA:. En esta imagen las xk son nuestras si

Salida:

Imprime un mensaje indicando que se han calculado las bases de Lagrange y devuelve el vector L.

Segunda función: `funcioninterpolL`

```
funcioninterpolL=function(y,l,n){
  s1=0
  for(i in 1:n){
    s1=s1+(y[i]*l[i])
  }
  print("A continuación se muestra el valor x interpolado")
  print(s1)
}
```

Esta función calcula el valor del polinomio interpolador en x utilizando las bases de Lagrange y los valores de y.

Parámetros:

- y: un vector que contiene los valores de las imágenes y correspondientes a los nodos si.
- l: el vector de las bases de Lagrange calculadas previamente.
- n: el número de nodos.

Algoritmo:

1. Inicializa una suma acumulativa `s1 = 0`.
2. Recorre cada índice i de 1 a n:

Finalmente, imprime el valor

$$P(x) = \sum_{i=1}^{n} y_i \cdot L_i(x)$$

interpolado.

Bloque principal

```
x=as.numeric(readline(prompt = "Inserte el valor a interpolar"))
n=as.numeric(readline(prompt = "Inserte la longitud del vector del soporte"))
for(i in 1:n){
  s[i]=as.numeric(readline(prompt = "Inserte 1 por 1 los valores del soporte"))
}
if(length(s)!=n){
  print("Debe conservar la longitud del soporte")
}
for(i in 1:n){
  y[i]=as.numeric(readline(prompt = "Inserte 1 por 1 los valores de la imagen del soporte"))
}
if(length(y)!=n){
  print("Debe conservar la longitud del soporte")
}
l=L(x,s,n)
funcioninterpolL(y,l,n)

```

El resto del código se encarga de recoger los datos de entrada del usuario y realizar la interpolación.

1. Solicita al usuario:
    - `x`: el valor de x en el que se desea interpolar.
    - `n`: el número de nodos.
2. Crea un vector `s` para los nodos y pide al usuario que introduzca sus valores uno por uno.
    - Si el número de valores ingresados no coincide con `n`, muestra un mensaje de error.
3. Crea un vector `y` para los valores yicorrespondientes a los nodos y los solicita uno por uno.
    - También verifica que el número de valores coincida con `n`.
4. Llama a la función `L` para calcular las bases de Lagrange.
5. Llama a `funcioninterpolL` para calcular e imprimir el valor interpolado.

Conclusión:

Este código es una implementación básica del método de interpolación de Lagrange. Es útil para entender los fundamentos de este método y cómo calcular manualmente un polinomio interpolador.