In interpolazione, poiché computazionalmente meno costosa, il polinomio scritto con le basi di Lagrange, viene spesso riscritto in forma baricentrica, ovvero

$$p_n(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x-x_i} y_i}{\sum_{i=0}^{n} \frac{w_i}{x-x_i}}$$

con i pesi w_i definiti come

$$w_i = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{1}{x_i - x_j}$$

Si implementi una function `interpol_bary` che calcola il polinomio in forma baricentrica e che prende come input

- `x`: vettore (riga) dei nodi
- `y`: vettore (riga) dei valori della funzione nei nodi
- `s`: vettore (colonna) dei punti di valutazione del polinomio

e restituisce in output un vettore `t` con i valori del polinomio negli elementi di `s`.
Attenzione: Se gli elementi di `x` sono contenuti in `s` avremo una divisione per 0 nella implementazione, anche se in realtà $\lim_{x\to x_i} p_n(x) = y_i$ perchè è zero anche il numeratore, quindi in questi casi i valori vanno specificati a mano e non con la formula sopra. Per fare ciò si può usare la funzione `ismember` di Matlab e definire tali valori come

```
[idy,locs] = ismember(x,s);
t(locs(locs~=0)) = y(idy);
```

In seguito applicare la function in uno script per interpolare la funzione $f(x) = \cos(x) - \sin(x)$ in $I = [-1, 1]$ utilizzando come nodi sia i punti equispaziati, sia i punti di Chebyshev-Lobatto e variando il grado del polinomio $n = 1 : 30$. Si valuti il polinomio in 500 punti all'interno dell'intervallo I

Calcolare poi, per ciascun n l'errore assoluto massimo tra la funzione e il polinomio all'interno della discretizzazione dell'intervallo.

Infine, fare il grafico in scala semilogaritmica del'errore assoluto massimo in funzione del grado n sovrapponendo nello stesso grafico l'errore con il polinomio che utilizza i nodi equispaziati (in blu) e l'errore che con il polinomio che utilizza i nodi di Chebyshev-Lobatto (in magenta). Si assegni al grafico il titolo "Errori "e si aggiunga una legenda.