

# Esercizio 1

Creare uno script che abbia la sintassi

**./elabora F**

dove **F** è il path assoluto di un file.

Lo script deve:

- richiedere all'utente e **leggere da standard input** un numero intero **N**.
- **controllare** che **N** sia un intero positivo
- **controllare** che **F** sia un path assoluto e corrisponda al nome di un file esistente e leggibile.
- scrivere in un **file di output le ultime N linee** del file **F ordinate in ordine lessicografico inverso**.

Il file di output sarà memorizzato nella home directory dell'utente che ha invocato lo script e dovrà avere il nome:

**results\_<uname>.out**

dove <uname> è il nome dello USER che ha invocato lo script

# Esercizio 1: suggerimenti (1/2)

## Lettura da standard input:

- `read var1 var2`

Le stringhe in ingresso vengono attribuite alle variabili a seconda della corrispondenza posizionale

## Test di file:

- `test -f <path>` Esistenza del file. Alternativa `[ -f <path> ]`
- `test -d <path>` Esistenza del direttorio
- `test -r <path>` Diritto di lettura (allo stesso modo, `-w` e `-x`)

## Test di N:

- `[[ ]]` Più comodo di `test` per testare regular expressions
-

# Esercizio 1: suggerimenti (2/2)

## Filtraggio delle ultime N linee di un file:

- **tail** → quale opzione per filtrare le ultime N linee? (v. man)

## Ordinamento delle linee di un file:

- **sort** → quale opzione per ordinamento inverso? (v. man ) [perchè sort e non rev?]

## Redirezione I/O

- **comando > F** st. output redirezionato sul file F path(>> per append)
- **comando < F** st. input preso dal file F

L'output del tail deve essere elaborato dal sort → **piping di comandi**

## Occorrono anche:

- home directory dell'utente che ha invocato lo script
- username dell'utente

→ vedere le **variabili di ambiente**

---