**Esercizi di esempio per bash di linux su redirezione I/O**

**Esercizio 1**

exec 6<&0 #Copio il descrittore dell’input in 6

exec < file-dati #Da queso momento l’input avviene da file-dati

set `wc -l file-dati` #Recupero il numero di righe di file-dati

nr=$1 # nr contiene il numero di righe di file-dati

# Con il ciclo seguente leggo e visualizzo a monitor le linee di file-dati

for (( i=0; i<nr; i++ ))

do

read -r riga

echo $riga

done

**Esercizio 2**

set `wc -l file-dati 2>> ERRORE.log` > log #Se si ferifica un errore il messaggio viene rediretto in ERRORE.log

# e l'output di set in file log

nr=$1 #Recupero il numero di righe di file-dati

# Con il seguente ciclo mando in uscita sul file output le righe lette dal file file-dati

for (( i=0; i<nr; i++ ))

do

read -r riga

echo $riga >> output

done < file-dati # Il blocco for chiuso in questo modo fa si che lo standard input al suo interno è file-dati

**Esercizio 3 (Esempio generale)**

#!/bin/bash

# Redirigere lo stdin usando 'exec'.

exec 6<&0 # Collega il descrittore di file nr.6 allo stdin.

# Salva lo stdin.

exec < file-dati # lo stdin viene sostituito dal file "file-dati"

read a1 # Legge la prima riga del file "file-dati".

read a2 # Legge la seconda riga del file "file-dati."

echo

echo "Le righe lette dal file."

echo "------------------------"

echo $a1

echo $a2

echo; echo; echo

exec 0<&6 6<&-

# È stato ripristinato lo stdin dal df nr.6, dov'era stato salvato,

#+ e chiuso il df nr.6 ( 6<&- ) per renderlo disponibile per un altro processo.

#

# <&6 6<&- anche questo va bene.

echo -n "Immetti dei dati "

read b1 # Ora "read" funziona come al solito, leggendo dallo stdin.

echo "Input letto dallo stdin."

echo "------------------------"

echo "b1 = $b1"

echo

exit 0