

Soluzioni prima parte Sistemi Operativi di Letizia Leonardi

07/07/2023

La parte in Shell deve prevedere un numero variabile di parametri **Q+1** (con Q maggiore o uguale a 2): il primo parametro deve essere considerato un **numero intero (X) strettamente positivo e strettamente minore di 4**, mentre gli altri Q devono essere **nomi assoluti di directory** che identificano Q gerarchie (G1, G2, ...) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in Q fasi, una per ogni gerarchia. Il programma, per ognuna delle Q fasi, deve esplorare la gerarchia G corrispondente - tramite un file comandi ricorsivo, FCR.sh – e deve cercare tutte le directory che si trovano a livello corrispondente al numero X. Si riporti il nome assoluto di tali directory sullo standard output. In ogni directory trovata, si deve invocare la parte in C passando come parametri i nomi di tutti i **file leggibili che NON siano vuoti** (F1, F2, ...) presenti in tale direttorio. **NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:**

- una variabile di nome X per contenere il primo parametro di FCP.sh;
- una variabile di nome G per le singole gerarchie di ognuna delle Q fasi;
- una variabile di nome F per identificare, via via, i singoli file delle directory trovate a livello X. **OSSERVAZIONE:** se per provare la parte shell, si commenta la chiamata alla parte C, ricordarsi di togliere il commento prima della consegna!

Tag: tre parametri, 2+1 parametri, intero positivo minore di X, file leggibili non vuoti, file lunghezza zero 0,

FCP.sh

```
#!/bin/bash

#Redirigo tutti gli errori a standard error,
#fornisco un codice di errore progressivo

#Controllo che i parametri non siano meno di 3
if test $# -lt 3
then
    echo Numero parametri incorretto, almeno 3 >&2
    exit 1
fi
X=$1
case $X in
    #Controllo che il primo par sia intero 0<=X<4
    0|1|2|3)    echo Primo parametro corretto
                ;;
    *)          echo Primo parametro non corretto, deve essere numero intero e
                minore di 4 >&2
                exit 2
                ;;
esac

#Per rimuovere il primo parametro
shift

#Per tutte le gerarchie passate
for G in $*
```

```

do
case $G in
    #Testo se il path è assoluto, e nel caso
    #se è una cartella attraversabile
    /*      )if test -d $G -a -x $G
            then
                echo $G parametro corretto
            else
                echo Parametro $G non corretto, deve essere una dir attraversabile
>&2

                exit 3
            fi

            ;;

    #Se il path non è assoluto
    *      )   echo Parametro $G non corretto, deve essere un nome assoluto di
directory >&2

                exit 4
            ;;

esac
done

#Esporto il ricorsivo per essere chiamato ovunque
#nelle sottocartelle
PATH="`pwd`: $PATH"
export PATH

#Assumo di partire dal livello zero visto che
# #0 appare essere un parametro accettabile dal
#testo del compito
LIVELLO=0
for G in $*
do
    #Su ogni gerarchia chiamo il ricorsivo dandogli
    #la cartella da esplorare il livello richiesto
    #e il livello attuale
    FCR.sh $G $X $LIVELLO
done

```

FCR.sh

```

# FCR.sh $G $X $LIVELLO
#Rinomino come richiesto
G=$1
X=$2
LIVELLO=$3

#Entro nella cartella, virgolette per impedire
#errore con gli spazi o altri caratteri
cd "$G"

#Aumento di livello perché sono entrato in una cartella
LIVELLO=$((LIVELLO + 1))
#Scritta di debug
echo La cartella `pwd` si trova al livello $LIVELLO , richiesto $X

```

```

#Se siamo al livello richiesto
if test $LIVELLO -eq $X
then
    #Stampo il nome assoluto della cartella
    echo CARTELLA TROVATA: `pwd`
    file_leggibili=
    for F in *
    do
        #Se leggibile e non vuoto
        if test -f "$F" -a -r "$F" -a -s "$F"
        then
            file_leggibili="$file_leggibili $F"
        fi
    done
    #Se ho trovato almeno un file, le virgolette
    #per impedire problemi con gli spazi
    if test "$file_leggibili"
    then
        echo Chiamerei main.c con $file_leggibili
    else
        echo Nessun file a specifica, non chiamo main.c
    fi
    exit 0
fi

#Se dobbiamo ancora arrivare al livello desiderato
if test $LIVELLO -lt $X
then
    for D in *
    do
        #Se il file nella cartella corrente
        #è una dir ed è attraversabile
        #virgolette per impedire problemi con gli
        #spazi
        if test -d "$D" -a -x "$D"
        then
            #Richiamo il ricorsivo aumentando il
            #numero del livello esplorato
            FCR.sh "$D" "$X" $LIVELLO
        fi
    done
fi

```

13/04/2022

La parte in Shell deve prevedere un numero variabile di parametri $Q+1$ (con Q maggiore o uguale a 2): il primo parametro deve essere considerato un numero intero strettamente positivo (X), mentre gli altri Q devono essere nomi assoluti di directory che identificano Q gerarchie (G_1, G_2, \dots) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in Q fasi, una per ogni gerarchia. Il programma, per ognuna delle Q fasi, deve esplorare la gerarchia G corrispondente - tramite un file comandi ricorsivo, FCR.sh – e deve contare **globalmente** tutti i file leggibili la cui lunghezza in linee sia uguale a X . Al termine di ognuna delle Q fasi, si deve riportare sullo standard output **per la gerarchia corrispondente** il numero totale di file trovati (che soddisfano la condizione sopra indicata). Quindi, al termine di tutte le Q fasi, si deve procedere ad un controllo sui file trovati: tale controllo deve prendere sempre a riferimento i file trovati nella prima gerarchia (G_1) e quindi per ognuno di tali file si deve controllare se

il contenuto è uguale al contenuto di ognuno dei file trovati nelle altre Q-1 gerarchie, riportando sullo standard output una opportuna frase di spiegazione per l'utente, ad esempio: I file /home/utente/ger1/d2/d3/f8 e /home/utente/ger2/d21/d32/f82 sono uguali
NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome X per contenere il primo parametro di FCP.sh;
- il nome /tmp/nomiAssoluti per la parte iniziale del nome dei file temporanei
- una variabile di nome G per le singole gerarchie di ognuna delle Q fasi;
- una variabile di nome F per identificare, via via, i singoli file delle directory esplorate e CANCELLATO(della directory selezionata;)

Tag: differenza, multipli file, lunghezza

FCP.sh

```
#!/bin/sh
#Soluzione della Prima Prova in itinere del 13 Aprile 2022
#versione che usa diversi file temporanei e usa FCR.sh per tutte le fasi

#controllo sul numero di parametri: deve essere maggiore o uguale a 3
case $# in
0|1|2) echo Errore: numero parametri $# quindi pochi parametri. Usage is $0 numero
dirass1 dirass2 ...
        exit 1;;
*)      echo OK: da qui in poi proseguiamo con $# parametri ;;
esac

#dobbiamo controllare che il primo parametro sia un numero e sia strettamente
maggiore di 0
case $1 in
*[^0-9]*) echo $1 non numerico o non positivo
        exit 2;;
*)      echo primo parametro giusto $1;;
esac

X=$1      #salviamo il numero nella variabile indicata dal testo e poi facciamo shift
shift

#ora in $* abbiamo solo i nomi delle gerarchie e quindi possiamo fare i controlli
sui nomi assoluti e sulle directory in un for
for G
do
    case $G in
/*) if test ! -d $G -o ! -x $G
        then
            echo $G non directory o non attraversabile
            exit 3
        fi;;
*)  echo $G non nome assoluto; exit 4;;
esac
done

#controlli sui parametri finiti possiamo passare alle Q fasi dopo aver settato ed
esportato il path
PATH=`pwd`: $PATH
export PATH

n=1 #serve per dare un nome diverso ai file temporanei e, volendo, per dire in che
```

```

fase siamo
for G
do
    > /tmp/nomiAssoluti$$-$n          #creiamo/azzeriamo il file temporaneo: N.B.
    l'indicazione $$ si poteva anche omettere, ma in questo caso era coerente con la
    specifica che indicava che la PARTE INIZIALE del nome dei vari file temporanei!
    echo DEBUG-fase $n per $G
    #invochiamo il file comandi ricorsivo con la gerarchia, il numero X e il
    file temporaneo
    FCR.sh $G $X /tmp/nomiAssoluti$$-$n
    #la specifica indicava che doveva essere riportato su standard output il
    conteggio di OGNI SINGOLA gerarchia!
    echo Il numero di file trovati in questa fase e\' `wc -l <
/tmp/nomiAssoluti$$-$n`
    n=`expr $n + 1`
done

#terminate tutte le ricerche ricorsive cioe' le Q fasi
#dobbiamo procedere con il confronto richiesto
for file1 in `cat /tmp/nomiAssoluti$$-1`          #per ogni file trovato nella prima
gerarchia
do
    n=1      #idem come prima: ATTENZIONE QUESTA INIZIALIZZAZIONE E' DA FARE
    ALL'INTERNO DI QUESTO CICLO!
    for i    #per ognuna delle gerarchie
    do
        if test $n -gt 1          #saltiamo il file temporaneo della prima
gerarchia; N.B. in alternativa prima di iniziare questa parte di confronto si
poteva usare il comando shift e quindi avere in $* solo le gerarchie a partire
dalla G2
            then
                for file2 in `cat /tmp/nomiAssoluti$$-$n`
                do
                    #Stampa di DEBUG
                    echo DEBUG-STIAMO PER CONTROLLARE il file $file1 e
il file $file2

                    if diff $file1 $file2 > /dev/null 2>&1 #basta
usare il comando diff che riporta 0 se i file sono uguali, altrimenti un numero
diverso da 0: da notare sia la ridirezione dello standard output che dello standard
error su /dev/null

                        then      echo I file $file1 e $file2 sono uguali
                        #else      echo I file $file1 e $file2 sono diversi
#a seconda dei testi doveva essere riportato che i file fossero uguali o diversi e
nel secondo caso era da fare questo else andando a non fare nulla nello then o
mettendo una condizione inversa
                            fi
                        done
                    fi
                done
                n=`expr $n + 1`
            done
        done
    done

done

#ci vuole la fase di cancellazione di tutti i file temporanei
n=1 #idem come prima
for G
do
    #eliminiamo il file temporaneo n-esimo
    rm /tmp/nomiAssoluti$$-$n

```

n=`expr \$n + 1`

done

FCR.sh

```
#!/bin/sh
#FCR.sh
#file comandi ricorsivo che scrive il nome dei file trovati sul file temporaneo
#il cui nome e' passato come terzo argomento

cd $1
#la variabile NR ci serve per il numero di linee
NR=

for F in *
do
    #controlliamo solo i nomi dei file leggibili come indicato nella specifica!
    if test -f $F -a -r $F
    then
        NR=`wc -l < $F`
        if test $NR -eq $2
        then
            #abbiamo trovato un file che soddisfa le specifiche e
            quindi inseriamo il suo nome assoluto nel file temporaneo
            echo `pwd`/$F >> $3
        fi
    fi
done

for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva: ATTENZIONE A PASSARE IL NOME ASSOLUTO DELLA
        dir INDIVIDUATA!
        FCR.sh `pwd`/$i $2 $3
    fi
done
```

22/04/2021

La parte in Shell deve prevedere un numero variabile di parametri **Q+2** (con Q maggiore o uguale a 2): il primo parametro deve essere considerato un numero intero strettamente positivo (W), il secondo parametro deve essere una stringa (S) mentre gli altri Q devono essere nomi assoluti di directory che identificano Q gerarchie (G1, G2, ...) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in Q fasi, una per ogni gerarchia. Il programma, per ognuna delle Q fasi, deve esplorare la gerarchia G corrispondente - tramite un file comandi ricorsivo, FCR.sh – e deve contare **globalmente** tutte le directory (inclusa la radice della gerarchia) che contengono almeno un file il cui nome abbia terminazione .S. **Al termine di tutte le Q fasi**, si deve riportare sullo standard output il numero totale di directory trovate **globalmente** (che soddisfano la condizione sopra indicata). Se tale conteggio risulta maggiore o uguale a W, si deve chiedere all'utente, chiamandolo con il **proprio nome di battesimo** (in caso di nome multipli, se ne scelga uno) un numero intero X compreso fra 1 e W; quindi, si deve selezionare la directory corrispondente

al numero X e di tale directory si deve riportare su standard output il nome assoluto. NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome W per contenere il primo parametro di FCP.sh;
- una variabile di nome S per contenere il secondo parametro di FCP.sh;
- il nome **/tmp/nomiAssoluti** per il file temporaneo
- una variabile di nome G per le singole gerarchie di ognuna delle Q fasi;
- una variabile di nome F per identificare, via via, i singoli file delle directory esplorate e CANCELLATO(della directory selezionata;)
- una variabile di nome X per contenere il numero letto.

Tag: numero positivo, stringa, nomi assoluti, nome file, termina per lettera, tutte le cartelle, stringa senza backslash, nome fissato, più parametri, for con lista, input limitato, richiesta numero, selezione n-esima riga, while con condizione su stringa, ultimo carattere, bool

FCP.sh

```
#!/bin/sh
#Soluzione della Prova del 22 Aprile 2021
# ATTENZIONE IL FILE FCP.sh DEVE ESSERE UN FILE IN CUI IL DIRITTO DI ESECUZIONE
ALMENO PER L'UTENTE RISULTA SETTATO!

case $# in
0|1|2|3) echo Errore: numero parametri $# quindi pochi parametri. Usage is $0 W S
dirass1 dirass2 ...
        exit 1;;
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
esac

#Controllo $1
case $1 in
*[^0-9]*) echo $1 non numerico o non positivo
        exit 2;;
*) if test $1 -eq 0
    then echo ERRORE: primo parametro $1 uguale a zero
        exit 3
    fi ;;
esac
W=$1      #nome specificato nel testo

#Controllo $2: ha senso che controlliamo che non contenga il carattere /
case $2 in
*/*) echo $2 non deve contenere il carattere \
        exit 4;;
esac
S=$2      #nome specificato nel testo

#quindi ora possiamo usare il comando shift due volte per eliminare i primi due
parametri salvati nelle variabili indicate dal testo
shift
shift

#ora in $* abbiamo solo i nomi delle gerarchie e quindi possiamo fare i controlli
sui nomi assoluti e sulle directory in un for
for G
do
```

```

case $G in
/*) if test ! -d $G -o ! -x $G
then
echo $G non directory
exit 5
fi;;
*) echo $G non nome assoluto; exit 6;;
esac

```

done

#controlli sui parametri finiti possiamo passare alle Q fasi, dopo aver settato il path

```

PATH=`pwd`: $PATH
export PATH

```

#azzeriamo il file temporaneo: usiamo un solo file temporaneo: il nome del file temporaneo era specificato nel testo e andava usato cosi' come indicato SENZA AGGIUNGERE \$\$

```
> /tmp/nomiAssoluti
```

```

for G
do

```

```
    echo fase per $G
```

#invochiamo il file comandi ricorsivo con la gerarchia, il secondo parametro (cioe' la stringa) e il file temporaneo; il numero passato come primo parametro NON serve al file ricorsivo!

```
    FCR.sh $G $S /tmp/nomiAssoluti #N.B. in questo caso non sarebbe stato indispensabile passare il nome del file
```

done

#N.B. Andiamo a contare le linee del file temporaneo

#dato che il numero di directory trovate lo dobbiamo stampare e poi lo dobbiamo controllare lo salviamo in una variabile

```
TOT=`wc -l < /tmp/nomiAssoluti`
```

```
echo Il numero di directory trovate nelle gerarchie = $TOT
```

#dobbiamo verificare se tale numero e' maggiore o uguale a W

```
if test $TOT -ge $W
then

```

#dobbiamo chiedere all'utente, chiamandolo con il nome di battesimo, un numero

```
echo -n "Letizia dammi per favore un numero che deve essere compreso fra 1 e $W "
```

#ATTENZIONE BISOGNA USARE \$W per fare capire all'utente il range e non scrivere solo W

#poiche' in questo caso si e' usata l'opzione -n bisogna inserire uno spazio bianco per fare in modo che l'echo del numero inserito NON venga attaccato al valore di \$W rendendo il tutto poco leggibile

```
read X #il nome della variabile da usare era specificato nel testo
```

#Controllo \$X: i controlli su X sono molto semplici da inserire dato che basta adattare quelli che sono stati fatti per il primo parametro; cambia solo che va verificato anche che sia minore o uguale a \$W

```
case $X in
```

```

*[^0-9]*) echo $X non numerico o non positivo
#cancelliamo il file temporaneo
rm /tmp/nomiAssoluti
exit 7;;

```

*) if test \$X -lt 1 -o \$X -gt \$W #ATTENZIONE NELLA VERIFICA SE IL NUMERO NON E' COMPRESO BISOGNA USARE -o e NON -a!

```
then echo ERRORE: $X non e' compreso fra 1 e $W
```



```

        #cancelliamo il file temporaneo
        rm /tmp/nomiAssoluti
        exit 8
    fi ;;
esac
echo -n il nome assoluto della directory corrispondente a $X =
#dobbiamo selezionare la linea $X del file temporaneo
head -$X < /tmp/nomiAssoluti | tail -1
else
    echo $TOT non maggiore o uguale a $W #ATTENZIONE: questo caso NON deve
essere considerato un errore!
fi

#cancelliamo il file temporaneo
rm /tmp/nomiAssoluti

```

FCR.sh

```

#!/bin/sh
# Soluzione della prova d'esame del 22 Aprile 2021
#
# File ricorsivo. Uso: $0 dirass stringa filetemp
# ATTENZIONE IL FILE FCR.sh DEVE ESSERE UN FILE IN CUI IL DIRITTO DI ESECUZIONE
ALMENO PER L'UTENTE RISULTA SETTATO!
#file comandi ricorsivo che scrive il nome delle dir trovate sul file
#il cui nome e' passato come terzo argomento

trovato=false #variabile che ci servira' per capire se abbiamo trovato almeno un
file che soddisfa la specifica (termina per .S) e quindi la directory e' giusta!

cd $1

for F in *
do
    if test -f $F #se è un file
    then
        #controlliamo se termina per la stringa cercata: ATTENZIONE USARE
        UN if test e' SBAGLIATO, dato che puo' funzionare se e solo se in ogni dir c'e'
        sempre e solo un file che rispetta la specifica!
        case $F in
            *.$2) #se il nome rispetta la specifica, settiamo trovato a true
                    trovato=true;; #dato che il testo richiedeva almeno uno
potremmo anche inserire un break!
            *) ;; #altrimenti non facciamo nulla. N.B. Questa alternativa
puo' anche essere omessa!
        esac
    fi
done

if test $trovato = true #se abbiamo trovato almeno un file dobbiamo salvare il nome
della dir corrente nel file temporaneo
then
    #dobbiamo scrivere il nome della directory nel file temporaneo
    pwd >> $3 #NON SERVE SCRIVERE echo `pwd` >> $3 dato che il comando
pwd gia' da solo riporta su standard output la dir corrente!
    #in alternativa si poteva scrivere: echo $1 >> $3

```

```

fi

for i in *
do
    if test -d $i -a -x $i
    then
        #invocazione ricorsiva passando sempre il nome assoluto della dir
        $0 `pwd`/$i $2 $3
    fi
done

```

14/06/2021

La parte in Shell deve prevedere un numero variabile di parametri $Q+2$ (con Q maggiore o uguale a 2): i primi Q devono essere nomi assoluti di directory che identificano Q gerarchie ($G1, G2, \dots$) all'interno del file system, mentre gli ultimi due parametri devono essere considerati numeri interi strettamente positivi (H e M). Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in Q fasi, una per ogni gerarchia. Il programma, per ognuna delle Q fasi, deve esplorare la gerarchia G corrispondente - tramite un file comandi ricorsivo, `FCR.sh` - e deve cercare tutte le directory che contengono un numero strettamente minore di H ma maggiore o uguale a 2 di file ($F1, F2, \dots$) il cui nome sia costituito da esattamente 2 caratteri e con lunghezza in linee esattamente uguale a M . Si riporti il nome assoluto di tali directory sullo standard output. In ognuna di tali directory trovate, si deve invocare la parte in C, passando come parametri i nomi dei file trovati ($F1, F2, \dots$) che soddisfano la condizione precedente e il numero M .

Tag: Numeri positivi, numero caratteri, lunghezza linee

FCP.sh

```

#!/bin/sh
#Soluzione della Prova del 14 Luglio 2021

case $# in
0|1|2|3) echo Errore: numero parametri $# quindi pochi parametri. Usage is $0 H K
dirass1 dirass2 ...
        exit 1;;
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
esac

#dobbiamo isolare gli ultimi due parametri e intanto facciamo i controlli
num=1 #la variabile num ci serve per capire quando abbiamo trovato il penultimo e
l'ultimo parametro
params= #la variabile params ci serve per accumulare i parametri a parte gli ultimi
due
#in $* abbiamo i nomi delle gerarchie e i due numeri interi
for i
do
    if test $num -lt `expr $# - 1` #ci serve per non considerare gli ultimi due
parametri che sono i numeri
    then
        #soliti controlli su nome assoluto e directory traversabile
        case $i in
/*)      if test ! -d $i -o ! -x $i
            then
                echo $i non directory o non attraversabile

```

```

        exit 2
    fi;;
*)      echo $i non nome assoluto; exit 3;;
esac
params="$params $i" #se i controlli sono andati bene memorizziamo
il nome nella lista params
else
    #abbiamo individuato gli ultimi due parametri e quindi facciamo il solito
    controllo su numerico e strettamente positivo
    #Controllo penultimo e ultimo parametro
    case $i in
        *[^0-9]*) echo $i non numerico o non positivo
                exit 4;;
        *) if test $i -eq 0
            then echo ERRORE: parametro $i uguale a zero
                exit 5
            fi ;;
    esac
    #se i controlli sono andati bene salviamo il penultimo e ultimo
    parametro

    if test $num -eq `expr $# - 1`
    then
        H=$i      #H nome indicato nel testo
    else
        M=$i      #M nome indicato nel testo
    fi

    fi
    num=`expr $num + 1` #incrementiamo il contatore del ciclo sui parametri
done

#controlli sui parametri finiti possiamo passare alle Q fasi, dopo aver settato il
path
PATH=`pwd`: $PATH
export PATH

#ora in $params abbiamo solo i nomi delle gerarchie
for G in $params      #G nome indicato nel testo
do
    echo fase per $G
    FCR.sh $G $H $M      #invochiamo il file comandi ricorsivo con la
    gerarchia e i due numeri passati come ultimi due parametri
done

```

FCR.sh

```

#!/bin/sh
#FCR.sh
#file comandi ricorsivo della prova del 14 Luglio 2021

cd $1
#la variabile cont ci serve per contare il numero di file, nome indicato nel testo
cont=0
#la variabile files ci serve per memorizzare i file
files=

for F in *      #F nome indicato nel testo

```

```

do
    #controlliamo solo i nomi dei file
    if test -f $F
    then
        #controlliamo la lunghezza del nome del file
        case $F in
            ??)    #caso giusto
                    #calcoliamo la lunghezza in linee del file per vedere se
soddisfa le specifiche
                    L=`wc -l < $F`
                    if test $L -eq $3
                    then
                        #aggiorniamo la variabile del conteggio
                        cont=`expr $cont + 1`
                        #salviamo il nome del file nella variabile apposita
                        files="$files $F"
                    fi
                ;;
            *) #non si deve fare nulla ;;
        esac
    fi
done

if test $cont -lt $2 -a $cont -ge 2
then
    echo TROVATO direttorio `pwd`
    #dobbiamo invocare la parte C passando i nomi dei file precedentemente
salvati insieme con il valore M (ultimo parametro parte principale)
    echo invochiamo 14Lug21 $files $3
    14Lug21 $files $3
fi

for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva cui passiamo il nome assoluto della dir
        $0 `pwd`/$i $2 $3
    fi
done

```

16/07/2010

La parte in Shell deve prevedere 3 parametri: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (G) all'interno del file system, il secondo deve essere il nome relativo semplice di un direttorio (D), mentre il terzo parametro deve essere considerato un numero intero strettamente positivo (H). Il programma deve cercare nella gerarchia G specificata tutti i direttori di nome D che contengono almeno un file che abbia un numero di linee uguale a H. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (F0, F1, ... FN-1) che soddisfano la condizione precedente e il numero intero H.

Tag: Nome assoluto, nome relativo semplice, numero intero positivo

FCP.sh

```
#!/bin/sh
#princ dirass D H

case $# in
0|1|2)  echo Errore: Usage is $0 dirass D H
        exit 1;;
3) ;;
*) echo TROPPI PARAMETRI
   exit 1;;
esac

#Controllo primo parametro
case $1 in
/*) if test ! -d $1 -o ! -x $1
    then
        echo $1 non direttorio
        exit 2
    fi;;
*)  echo $1 non nome assoluto
    exit 3;;
esac

#Controllo secondo parametro
case $2 in
/*) echo $2 NON relativo
    exit 4;;
esac

#Controllo terzo parametro
expr $3 + 0 > /dev/null 2>&1
N1=$?
if test $N1 -ne 2 -a $N1 -ne 3
then echo numerico $3 #siamo sicuri che numerico
    if test $3 -le 0
    then echo $3 non positivo
        exit 4
    fi
else
    echo $3 non numerico
    exit 5
fi

PATH=`pwd`: $PATH
export PATH
FCR.sh $*
```

FCR.sh

```
#!/bin/sh
#esplora dirass D H

cd $1
#definiamo una variabile per memorizzare i nomi dei file
```

```

file=

if test -d $2
then
    cd $2
    for i in *
    do
        if test -f $i -a -r $i
        then
            nl=`wc -l < $i`
            if test $nl -eq $3
            then
                file="$file $i"
            fi
        fi
    done
    cd ..
fi

if test $file
then
    echo TROVATO DIRETTORIO `pwd`/$2
    echo CHIAMO PARTE C
    partec $file $3
fi

for i in *
do
    if test -d $i -a -x $i
    then
        echo RICORSIONE in `pwd`/$i
        FCR.sh `pwd`/$i $2 $3
    fi
done

```

Convertito con:

```

npm install -g pretty-markdown-pdf
pretty-md-pdf -i soluzioni_prima_parte.md

```