

Soluzioni prima parte Sistemi Operativi di Letizia Leonardi

07/06/2023

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**;
- 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=`expr $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):

- i primo parametro deve essere considerato un numero intero strettamente positivo (X);
- 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** 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 (G1) 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, file uguali, due for, for innestato

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

        n=`expr $n + 1`
    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 riorsivo!
```

```
    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, ...) all'interno del file system;
- 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, ...) 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, ...) 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
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

```

```
#!/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
```

La parte in Shell deve prevedere un numero variabile di parametri N+1 (con N maggiore o uguale a 2):

- il primo parametro deve essere considerato un singolo carattere C;
- gli altri N devono essere nomi assoluti di directory che identificano N gerarchie (G1, G2, ...) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi, una per ogni gerarchia. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia Gi corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente tutte le directory D che hanno le caratteristiche di seguito specificate.

Il file comandi ricorsivo FCR.sh deve cercare tutte le directory D il cui nome sia di 3 caratteri e i cui caratteri dispari, sempre nel nome, siano uguali al carattere C. Al termine di tutte le N fasi, si deve riportare sullo standard output:

- a) il numero totale di directory D trovate globalmente (che soddisfano la condizione sopra indicata);
- b) i nomi assoluti delle directory D trovate e, contestualmente, per ogni directory D si deve chiedere all'utente, chiamandolo Elena, se vuole o meno visualizzare il contenuto della directory D compresi anche gli elementi nascosti.

NOTA BENE NEL FILE COMANDI FCP.sh SI USI OBBLIGATORIAMENTE:

- una variabile di nome C per il primo parametro;
- il nome /tmp/nomiAssoluti per il file temporaneo
- una variabile di nome answer per ricevere la risposta di Elena

FCP.sh

```
#!/bin/sh
# Soluzione della prova d'esame del 3 Giugno 2020
#
# File comandi. Uso: $0 C dirass1 dirass2 ...
#

# controllo sul numero dei parametri
case $# in
0|1|2)    echo "Errore nel numero dei parametri $#";
          exit 1;;
*)        echo DEBUG-"Numero parametri corretti $#";;
esac

#controllo su primo parametro
case $1 in
?) ;;
*)        echo $1 non singolo carattere
          exit 2;;
esac

#salviamo il primo parametro: il nome della variabile da usare era specificato nel
testo
C=$1

#ora possiamo usare il comando 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 i
do
    case $i in
    /*) if test ! -d $i -o ! -x $i
```

```

        then
            echo $i non directory
            exit 3
        fi;;
    *) echo $i non nome assoluto; exit 4;;
esac

done

#controlli sui parametri finiti possiamo passare alle N fasi

PATH=`pwd`: $PATH
export PATH

#azzeriamo il file temporaneo: usiamo un solo file temporaneo: il nome del file
temporaneo era specificato nel testo
> /tmp/nomiAssoluti

for i
do
    echo fase per $i
    #invochiamo il file comandi ricorsivo con la gerarchia corrente, il
carattere e il file temporaneo
    FCR.sh $i $C /tmp/nomiAssoluti #N.B. poiche' il nome del file temporaneo
era fisso si poteva anche non passarlo, ma si e' deciso di passarlo per uniformita'
per i casi in cui bisogna per forza passarlo
done

echo Il numero di direttori trovati nella gerarchia = `wc -l < /tmp/nomiAssoluti`
#N.B. Andiamo a contare le linee del file temporaneo
#In questo caso possiamo recuperare anche i nomi (assoluti) delle directory trovate
for i in `cat /tmp/nomiAssoluti`
do
    echo Il nome assoluto di una delle directory trovate: $i
    echo -n "Elena vuoi visualizzarne il contenuto, compresi gli elementi
nascosti (Si/si/Yes/yes)? " #il nome dell'UTENTE da usare era specificato nel testo
    read answer #il nome della variabile da usare era specificato nel testo
    case $answer in
        s* | S* | Y* | y*) ls -a $i;;
        *) ;; #non facciamo nulla!
    esac
done

#cancelliamo il file temporaneo
rm /tmp/nomiAssoluti

```

FCR.sh

```

#!/bin/sh
# Soluzione della prova d'esame del 3 Giugno 2020
#
# File ricorsivo. Uso: $0 dirass C filetemp
#file comandi ricorsivo che scrive il nome dei file trovati sul file
#il cui nome e' passato come terzo argomento
cd $1

#poiche' cerchiamo delle directory NON si puo' escludere che anche la radice della

```

```

gerarchia soddisfi le specifiche indicate nel testo!
#quindi controlliamo $1 che la prima volta sara' il nome della gerarchia corrente
case $1 in
    */$2?$2) #se il nome rispetta la specifica, lo scriviamo sul file
temporaneo: in questo caso la specifica richiedeva che il nome del file fosse di 3
caratteri e che i caratteri di posizione dispari (cioe' il primo e il terzo)
fossero uguali al carattere C passato
        pwd >> $3 ;; #NON SERVE SCRIVERE echo `pwd` >> $3 dato che il comando pwd
gia' da solo riporta su standard output la dir corrente!
*) ;; #altrimenti non facciamo nulla
esac

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

```

05/04/2019

La parte in Shell deve prevedere un numero variabile di parametri $N+1$ (con N maggiore o uguale a 2):

- il primo parametro deve essere considerato un nome relativo semplice F ;
- gli altri N devono essere nomi assoluti di directory che identificano N gerarchie (G_1, G_2, \dots) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi, una per ogni gerarchia. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia G_i corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente tutti i file che hanno le caratteristiche di seguito specificate.

Il file comandi ricorsivo FCR.sh deve cercare in tutte le directory ogni file leggibile con nome relativo semplice F e con lunghezza in linee maggiore o uguale 4; si dovrà quindi salvare in modo opportuno il nome assoluto del file trovato contestualmente alla sua lunghezza in linee.

Al termine di tutte le N fasi, si deve riportare sullo standard output il numero totale di file trovati globalmente. In seguito, per ogni file trovato, si deve riportare sullo standard output il suo nome assoluto, e quindi la sua lunghezza in linee (entrambe queste informazioni precedentemente salvate):

- quindi si deve richiedere all'utente un numero intero X compreso fra 1 e il numero corrispondente alla lunghezza in linee e , e se tale numero è corretto, si devono riportare su standard output le prime X linee del file con opportune frasi che facciano capire all'utente che cosa si sta visualizzando.

FCP.sh

```

#!/bin/sh
#Soluzione della Prima Prova in itinere del 5 Aprile 2019 - testo turni 1 e 2
#versione che usa un file temporaneo e usa FCR.sh per tutte le fasi

#controllo sul numero dei parametri  $N \geq 2$  e quindi  $N+1 \geq 3$ 
case $# in
0|1|2) echo Errore: numero parametri is $# quindi pochi parametri. Usage is $0

```

```

nomedir dirass1 dirass2 ...
    exit 1;;
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
esac

#Controllo primo parametro sia dato in forma relativa semplice
case $1 in
*/*)    echo Errore: $1 non in forma relativa semplice
        exit 2;;
*)      ;;
esac

F=$1 #salviamo il primo parametro
#quindi ora possiamo usare il comando 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 i
do
    case $i in
/*)    if test ! -d $i -o ! -x $i
        then
            echo $i non directory o non traversabile
            exit 3
        fi;;
*)      echo $i non nome assoluto; exit 4;;
    esac
done

#controlli sui parametri finiti possiamo passare alle N fasi
PATH=`pwd`: $PATH
export PATH
> /tmp/conta$$ #creiamo/azzeriamo il file temporaneo. NOTA BENE: SOLO 1 file
temporaneo!

for i
do
    echo fase per $i
    #invochiamo il file comandi ricorsivo con la gerarchia, il nome relativo
del file e il file temporaneo
    FCR.sh $i $F /tmp/conta$$
done
#terminate tutte le ricerche ricorsive cioe' le N fasi
#N.B. Andiamo a contare le linee del file /tmp/conta$$
echo Il numero di file totali che sono stati trovati sono = `wc -l < /tmp/conta$$`
c=0      #variabile che serve nel for successivo per capire se l'elemento corrente
del for e' un nome assoluto di file o la sua lunghezza in linee
for i in `cat /tmp/conta$$`
do
    #Stampiamo (come richiesto dal testo) i nomi assoluti dei file trovati e la
lunghezza in linee
    c=`expr $c + 1`
    if test `expr $c % 2` -eq 1
    then
        echo elemento dispari, quindi nome assoluto del file $i
        nomefile=$i      #memorizziamo il nome del file che ci serve nella
iterazione seguente
    else

```



```

echo elemento pari, quindi lunghezza in linee del file =$i
echo -n "dammi un numero compreso fra 1 e $i estremi compresi "
read X
#controlliamo che quanto inserito dall'utente sia una stringa
numerica (con case)
case $X in
*[*0-9]*) echo stringa inserita $X non numerica
          exit 5;;
*) ;;
esac
#controlliamo che il numero sia nel range giusto
if test $X -ge 1 -a $X -le $i
then
    #poiche' alcuni testi richiedevano le prime X linee e altri
le ultime X linee, si sono riportate entrambe le versioni!
    echo le prime $X linee del file sono
    head -$X $nomefile
    echo le ultime $X linee del file sono
    tail -$X $nomefile
else echo Numero inserito $X non nel range #in questo caso,
decidiamo di non uscire con errore, ma di passare al prossimo se esiste!
fi
fi
done
#da ultimo eliminiamo il file temporaneo
rm /tmp/conta$$

```

FCR.sh

```

#!/bin/sh
#FCR.sh
#file comandi ricorsivo che scrive il nome dei file creati sul file temporaneo
#il cui nome e' passato come terzo argomento
NL=      #variabile in cui salviamo il numero di linee del file corrente

cd $1    #ci posizioniamo nella directory giusta

if test -f $2 -a -r $2          #se esiste un file leggibile con il nome
specificato
then
    NL=`wc -l < $2`
    if test $NL -ge 4 #solo se il numero di linee e' maggiore o uguale di 4 si
deve contare il file
    then
        echo `pwd`/$2 $NL >> $3 #NOTA BENE: su una stessa linea sono
inseriti nome assoluto file trovato e sua lunghezza in linee: per il funzionamento
del cat usato nel for del file principale e' assolutamente indifferente; l'unica
cosa che cambia che facendo in questo modo NON serve dividere per due il conteggio
    fi
fi

for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva cui passiamo come primo parametro il nome

```

```
assoluto della directory
FCR.sh `pwd`/$1 $2 $3
fi
done
```

11/04/2018

La parte in Shell deve prevedere un numero variabile di parametri N+1 (con N maggiore o uguale a 2):

- il primo parametro deve essere considerato un numero intero Y strettamente positivo;
- gli altri N devono essere nomi assoluti di directory che identificano N gerarchie (G1, G2, ...) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi, una per ogni gerarchia. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia Gi corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente tutti i file che saranno creati secondo quanto di seguito specificato.

Il file comandi ricorsivo FCR.sh deve cercare in tutte le directory ogni file leggibile f1 con lunghezza in linee maggiore o uguale Y; si dovrà quindi selezionare la singola linea numero 5 a partire dall'inizio del file che dovrà essere scritta in un file appositamente creato avente nome f.quinta; in caso, il file sia più corto e non sia possibile selezionare la linea sopra specificata bisogna creare un file vuoto di nome f.NOquinta.

Al termine di tutte le N fasi, si deve riportare sullo standard output il numero totale di file creati globalmente. In seguito, per ogni file creato, si deve riportare sullo standard output il suo nome assoluto, e quindi il suo contenuto con opportune frasi che facciano capire all'utente che cosa si sta visualizzando.

FCP.sh

```
#!/bin/sh
#Soluzione della Prima Prova in itinere del 11 Aprile 2018: turni 1 e 2
#versione che usa un file temporaneo e usa FCR.sh per tutte le fasi

#controllo sul numero dei parametri N >= 2 e quindi N+1 >=3
case $# in
0|1|2) echo Errore: numero parametri is $# quindi pochi parametri. Usage is $0 Y
dirass1 dirass2 ...
exit 1;;
*) echo OK: da qui in poi proseguiamo con $# parametri ;;
esac

#Controllo primo parametro sia un numero (con case)
case $1 in
*[^0-9]*) echo Errore: $1 non numero
exit 2;;
*) if test $1 -eq 0
then
echo $1 non diverso da 0
exit 3
fi;;
esac

Y=$1 #salviamo il primo parametro
#quindi ora possiamo usare il comando 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 i
do
    case $i in
        /*) if test ! -d $i -o ! -x $i
            then
                echo $i non directory o non traversabile
                exit 4
            fi;;
        *) echo $i non nome assoluto; exit 5;;
    esac
done
#controlli sui parametri finiti possiamo passare alle N fasi
PATH=`pwd`: $PATH
export PATH
> /tmp/conta$$ #creiamo/azzeriamo il file temporaneo. NOTA BENE: SOLO 1 file
temporaneo!

for i
do
    echo fase per $i
    #invochiamo il file comandi ricorsivo con la gerarchia, il numero e il file
temporaneo
    FCR12.sh $i $Y /tmp/conta$$
done

#terminate tutte le ricerche ricorsive cioe' le N fasi
#N.B. Andiamo a contare le linee del file /tmp/conta$$
echo Il numero di file totali che sono stati creati sono = `wc -l < /tmp/conta$$`
for i in `cat /tmp/conta$$`
do
    #Stampiamo (come richiesto dal testo) i nomi assoluti dei file creati
    echo Creato il file $i
    case $i in
        *N0*) echo il file originale non conteneva almeno 5 linee e quindi il file
$i e\' vuoto ;;
        *) echo il contenuto del file $i e\' il seguente\:
            cat < $i ;;
    esac
done

#da ultimo eliminiamo il file temporaneo
rm /tmp/conta$$

```

FCR.sh

```

#!/bin/sh
#FCR.sh
#file comandi ricorsivo che scrive il nome dei file creati sul file temporaneo
#il cui nome e' passato come terzo argomento
NL=      #variabile in cui salviamo il numero di linee del file corrente

cd $1    #ci posizioniamo nella directory giusta

for i in *

```

```

do
    if test -f $i -a -r $i #per ogni file leggibile
    then
        NL=`wc -l < $i`
        if test $NL -ge $2 #controlliamo la lunghezza in linee del file
        rispetto a Y
        then
            if test $NL -ge 5 #controlliamo la lunghezza in linee del
            file ora rispetto a 5!
            then
                head -5 $i | tail -1 > $i.quinta #creiamo un
                file con il nome specificato
                echo `pwd`/$i.quinta >> $3 #salviamo
                il suo nome nel file temporaneo
            else
                > $i.N0quinta #il file non ha abbastanza linee!
                #creiamo un file con il nome specificato
                echo `pwd`/$i.N0quinta >> $3
                #salviamo il suo nome nel file temporaneo
            fi
        fi
    fi
done

for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva cui passiamo come primo parametro il nome
        assoluto della directory
        FCR12.sh `pwd`/$i $2 $3
    fi
done

```

07/04/2017

La parte in Shell deve prevedere un numero variabile di parametri N+1 (con N maggiore o uguale a 2):

- il primo parametro deve essere il nome relativo semplice F di un file;
- gli altri N devono essere nomi assoluti di direttori che identificano N gerarchie (G1, G2, ...) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi, una per ogni gerarchia. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia Gi corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente tutti i file con nome relativo semplice sorted che saranno creati secondo quanto di seguito specificato:

- il file comandi ricorsivo FCR.sh deve cercare tutti i file leggibili il cui nome sia F;
- per ognuno di tali file deve produrre un file ordinato (di nome sorted) secondo l'ordine alfabetico, ma senza tenere conto di minuscole/maiuscole.

Al termine di tutte le N fasi, si deve riportare sullo standard output il numero totale di file creati globalmente. In seguito, per ogni file creato, si deve riportare sullo standard output il suo nome assoluto, e quindi la prima e l'ultima linea del file (esattamente in questo ordine, con opportune frasi che facciano capire all'utente che cosa si sta visualizzando).

FCP.sh

```
#!/bin/sh
```

```
#Soluzione della Prima Prova in itinere del 7 Aprile 2017
```

```
#controllo sul numero dei parametri N >= 2 e quindi N+1 >=3
```

```
#N.B. se la specifica avesse indicato N > 1 comunque il controllo sarebbe analogo  
dato che N+1 > 1+1 > 2 e quindi N+1 deve essere alla fine o 3 o maggiore!
```

```
case $# in
```

```
0|1|2) echo Errore: numero parametri is $# quindi pochi parametri. Usage is $0  
nomefile dirass1 dirass2 ...
```

```
    exit 1;;
```

```
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
```

```
esac
```

```
#Controllo primo parametro sia dato in forma relativa semplice (quello che nel  
testo viene indicato come F)
```

```
case $1 in
```

```
*/) echo Errore: $1 non in forma relativa semplice
```

```
    exit 2;;
```

```
*) ;;
```

```
esac
```

```
F=$1 #salviamo il primo parametro appunto in una variabile che chiamiamo F
```

```
#quindi ora possiamo usare il comando 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 i
```

```
do
```

```
    case $i in
```

```
    /*) if test ! -d $i -o ! -x $i  
        then
```

```
            echo $i non directory o non traversabile
```

```
            exit 3
```

```
        fi;;
```

```
    *) echo $i non nome assoluto; exit 4;;
```

```
    esac
```

```
done
```

```
#controlli sui parametri finiti possiamo passare alle N fasi
```

```
PATH=`pwd`: $PATH
```

```
export PATH
```

```
> /tmp/conta$$ #creiamo/azzeriamo il file temporaneo. NOTA BENE: SOLO 1 file  
temporaneo!
```

```
for i
```

```
do
```

```
    echo fase per $i
```

```
    #invochiamo il file comandi ricorsivo con la gerarchia, il nome relativo  
del file (F) e il file temporaneo
```

```
    FCR.sh $i $F /tmp/conta$$
```

```
done
```

```
#terminate tutte le ricerche ricorsive cioe' le N fasi
```

```
#N.B. Andiamo a contare le linee del file /tmp/conta$$
```

```
echo Il numero di file totali che soddisfano la specifica = `wc -l < /tmp/conta$$`
```

```
for i in `cat /tmp/conta$$`
```

```
do
```

```
    #Stampiamo (come richiesto dal testo) i nomi assoluti dei file creati
```

```
echo Creato il file $i
echo La sua prima linea
head -1 $i
echo La sua ultima linea
tail -1 $i
```

done

```
#da ultimo eliminiamo il file temporaneo
rm /tmp/conta$$
```

FCR.sh

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

cd $1 #ci posizioniamo nella directory giusta

if test -f $2 -a -r $2 #verifichiamo se nella dir corrente esiste un file
leggibile con il nome richiesto
then
    sort -f $2 > sorted #usiamo l'opzione -f per fare l'ordinamento senza
tenere in conto maiuscole/minuscole (per fare l'ordinamento senza linee doppie si
doveva usare l'opzione -u)
    #NOTA BENE: la ridirezione in uscita e' necessaria per salvare il risultato
del comando sort!
    echo `pwd`/sorted >> $3 #salviamo il nome assoluto del file creato nel
file temporaneo
fi

#ricorsione in tutta la gerarchia
for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva cui passiamo come primo parametro il nome
assoluto della directory
        FCR.sh `pwd`/$i $2 $3
    fi
done
```

15/04/2016

La parte in Shell deve prevedere un numero variabile di parametri N+1 (N maggiore o uguale a 2):

- i primi N parametri devono essere nomi assoluti di directory che identificano N gerarchie (G1, G2, ...) all'interno del file system;
- l'ultimo parametro deve essere considerato un numero intero X strettamente positivo.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi, una per ogni gerarchia. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia Gi corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e devono contare globalmente tutte le directory che soddisfano le seguenti due specifiche:

1. devono contenere solo file (e non sotto-directory)
2. devono contenere solo file la cui lunghezza in linee sia strettamente maggiore di X.

Al termine di tutte le N fasi, si deve riportare sullo standard output il numero totale delle directory trovate globalmente che soddisfano la specifica precedente (directory trovate).

Inoltre, per ogni directory trovata, si deve riportare sullo standard output il suo nome assoluto, e quindi per ognuno dei file (file corrente) in esso contenuto deve essere riportato sullo standard output il suo nome assoluto e contestualmente la linea X-esima del file corrente a partire dalla fine del file.

FCP.sh

```
#!/bin/sh
#Soluzione della Prima Prova in itinere del 15 Aprile 2016
#versione che usa un file temporaneo 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 is $# quindi pochi parametri. Usage is $0
numero dirass1 dirass2 ...
        exit 1;;
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
esac
#dobbiamo isolare l'ultimo parametro e intanto facciamo i controlli
num=1 #la variabile num ci serve per capire quando abbiamo trovato l'ultimo
parametro
params= #la variabile params ci serve per accumulare i parametri a parte l'ultimo
#in $* abbiamo i nomi delle gerarchie e il numero intero
for i
do
    if test $num -ne $# #ci serve per non considerare l'ultimo parametro che e'
il numero
    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 l'ultimo parametro e quindi facciamo il solito
controllo su numerico e strettamente positivo
        #Controllo ultimo parametro (con expr)
        expr $i + 0 > /dev/null 2>&1
        N1=$?
        if test $N1 -ne 2 -a $N1 -ne 3
        then #echo numerico $i siamo sicuri che numerico
            if test $i -le 0
            then echo $i non strettamente positivo
                exit 4
            fi
        else
```

```

                                echo $i non numerico
                                exit 5
        fi
        X=$i #se i controlli sono andati bene salviamo l'ultimo parametro
    fi
    num=`expr $num + 1` #incrementiamo il contatore del ciclo sui parametri
done
#controlli sui parametri finiti possiamo passare alle N fasi
PATH=`pwd`: $PATH
export PATH
> /tmp/conta$$ #creiamo/azzeriamo il file temporaneo. NOTA BENE: SOLO 1 file
temporaneo!

for i in $params
do
    echo fase per $i
    #invochiamo il file comandi ricorsivo con la gerarchia, il numero e il file
temporaneo
    FCR.sh $i $X /tmp/conta$$
done

#terminate tutte le ricerche ricorsive cioe' le N fasi
#N.B. Andiamo a contare le linee del file /tmp/conta$$
echo Il numero di directory totali che soddisfano la specifica = `wc -l <
/tmp/conta$$`
for i in `cat /tmp/conta$$`      #nel file temporaneo abbiamo i nomi assoluti delle
directory trovate
do
    #Stampiamo (come richiesto dal testo) i nomi assoluti delle directory
trovate
    echo "Trovato la directory $i; contiene i seguenti file:"
    cd $i #ci spostiamo nella directory corrente del for
    for file in * #siamo sicuri che sono solo file (controllo fatto da FCR.sh)
    do
        echo "file: `pwd`/$file"
        echo "la cui linea $X-esima a partire dalla fine e' la seguente:"
        #selezioniamo direttamente la $X-esima linea del file corrente a
partire dalla fine
        tail -$X $file | head -1
    done
done
done
#da ultimo eliminiamo il file temporaneo
rm /tmp/conta$$

```

FCR.sh

```

#!/bin/sh
#FCR.sh
#file comandi ricorsivo che scrive il nome delle directory trovate sul file
temporaneo
#il cui nome e' passato come terzo argomento

cd $1
#la variabile NR ci serve per il numero di linee dei file; in questo caso la
inizializziamo per fare un ulteriore controllo come spiegato in seguito
NR=0

```



```

#la variabile DIR ci serve per capire se ci sono solo file
DIR=false
#la variabile trovato ci serve per capire se tutti i file rispettano la specifica
trovato=true

for i in *
do
    #controlliamo solo i nomi dei file (se inseriamo anche il controllo se
    leggibili per cautelarci dato che dopo usiamo il comando wc bisogna commentarlo!)
    if test -f $i
    then
        #calcoliamo il numero di linee
        NR=`wc -l < $i`
        #controlliamo se il numero delle linee NON e' strettamente maggiore
        di X
        if test $NR -le $2
        then
            #abbiamo trovato un file che NON soddisfa le specifiche e
            quindi mettiamo a false trovato
            trovato=false
        fi
    else
        if test -d $i
        then
            #abbiamo trovato una directory e quindi dobbiamo mettere a true DIR
            DIR=true
        fi
    fi
done
#se i due booleani sono rimasti ai valori iniziali allora abbiamo trovato una
directory giusta. Nota bene: l'ultimo controllo garantisce che si sia trovato
almeno un file dato che una directory vuota non rispetta le specifiche!
if test $DIR = false -a $trovato = true -a $NR -ne 0
then
    pwd >> $3 #salviamo il nome della directory corrente (che soddisfa le
    specifiche) nel file temporaneo
    #NOTA BENE: in questo caso basta usare il comando pwd e quindi e' ritenuto
    sbagliato scrivere echo `pwd`
fi

#ricorsione in tutta la gerarchia
for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva cui passiamo come primo parametro il nome
        assoluto della directory
        FCR.sh `pwd`/$i $2 $3
    fi
done

```

11/04/2014

La parte in Shell deve prevedere N parametri: (con N maggiore o uguale a 2):

- devono essere nomi assoluti di direttori che identificano N gerarchie (G1, G2, ... GN) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi. Il programma, per ognuna delle N fasi, deve esplorare la gerarchia Gi specificata - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente per ogni gerarchia Gi tutti i file leggibili che soddisfano la seguente specifica:

- il contenuto del file deve essere tale per cui tutte le sue linee iniziano con il carattere 'a'.

Al termine di tutte le N fasi, per ognuna delle gerarchie Gi, si deve riportare sullo standard output il nome della gerarchia Gi e il numero totale di file di quella gerarchia che soddisfano la specifica precedente (file trovati);

inoltre, si deve riportare sullo standard output il nome assoluto di ogni file trovato chiedendo contestualmente all'utente un numero X: tale numero deve essere usato per riportare sullo standard output le prime X linee del file trovato corrente.

FCP.sh

```
#!/bin/sh
#Soluzione della Prima Prova in itinere del 11 Aprile 2014
#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 2
case $# in
0|1)    echo Errore: numero parametri $# quindi pochi parametri. Usage is $0
dirass1 dirass2 ...
        exit 1;;
*)      echo DEBUG-OK: da qui in poi proseguiamo con $# parametri ;;
esac
#in $* abbiamo solo i nomi delle gerarchie e quindi possiamo fare i controlli sui
nomi assoluti e sulle directory in un for
for i
do
    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
done

#controlli sui parametri finiti possiamo passare a settare ed esportare la
variabile PATH
PATH=`pwd`: $PATH
export PATH

#ora possiamo passare alle N fasi
n=1 #serve per dare un nome diverso ai file temporanei e volendo per dire in che
fase siamo
for i
do
    > /tmp/conta$$-$n #creiamo/azzeriamo il file temporaneo
    echo fase $n per la gerarchia $i
    #invochiamo il file comandi ricorsivo con la gerarchia e il file temporaneo
    FCR.sh $i /tmp/conta$$-$n
    n=`expr $n + 1`
done

#terminate tutte le ricerche ricorsive cioe' le N fasi
```

```

n=1 #idem come prima
for i
do
    #N.B. Andiamo a contare le linee del file /tmp/conta$$-$n
    echo Il numero di file totali della gerarchia $i che soddisfano la
specifica = `wc -l < /tmp/conta$$-$n`
    for j in `cat /tmp/conta$$-$n`
    do
        #Stampiamo (come richiesto dal testo) i nomi assoluti dei file
trovati

        echo Trovato il file $j
        #chiediamo all'utente il numero X
        echo -n "Dammi il numero X "      #l'opzione -n serve per NON fare
andare a capo il cursore!
        read X
        #Controllo X (con case): questo controllo non e' richiesto, ma e'
opportuno inserirlo!
        case $X in
        *[^0-9]*) echo non numerico o non positivo
                    continue;; #usiamo continue e non exit in modo da saltare
al prossimo file
        *) ;;
        esac
        echo Ora mostriamo le sue prime $X linee o tutto il contenuto se
piu\ corti:

        #selezioniamo direttamente le prime $X linee del file corrente
        head -$X < $j

    done
    n=`expr $n + 1`
done

#cancelliamo tutti i file temporanei
n=1 #idem come prima
for i
do
    rm /tmp/conta$$-$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 secondo argomento

cd $1
#la variabile NR ci serve per il numero di linee
NR=
#la variabile NG ci serve per il numero di linee trovate dal grep
NG=

for i in *
do
    #controlliamo solo i nomi dei file leggibili come indicato nella specifica!

```

```

        if test -f $i -a -r $i
        then
            NR=`wc -l < $i`
            if test $NR -ne 0           #e' opportuno inserire il controllo che il
file non sia vuoto!
            then
                NG=`grep '^a' $i | wc -l` #con il grep cerchiamo le linee
che iniziano con il carattere richiesto (cioe' 'a'); contiamo quindi quante linee
soddisfano questa specifica
                if test $NR -eq $NG #se il numero di linee del file e'
uguale al numero di linee ricavate dal grep vuole dire che tutte le linee
cominciano con il carattere richiesto!
                then
                    #abbiamo trovato un file che soddisfa le specifiche
e quindi inseriamo il suo nome assoluto nel file temporaneo
                    echo `pwd`/$i >> $2
                fi
            fi
        fi
done

#ricorsione in tutta la gerarchia
for i in *
do
    if test -d $i -a -x $i
    then
        #chiamata ricorsiva
        FCR.sh `pwd`/$i $2
    fi
done

```

12/04/2013

La parte in Shell deve prevedere tre 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 una stringa (S);
- il terzo parametro deve essere considerato un numero intero strettamente positivo (N).

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in due fasi (Fasi A e B).

Il programma nella fase **A** deve esplorare la gerarchia G specificata - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente tutti i direttori (inclusa la radice) che contengono almeno un file leggibile il cui nome abbia terminazione .S.

Al termine dell'intera esplorazione ricorsiva di G, si deve riportare sullo standard output il numero totale di direttori della gerarchia G che soddisfano la condizione precedente insieme con il loro nome assoluto. Quindi il programma passa nella fase B solo se è il numero totale di direttori totali è strettamente maggiore di N.

In questa fase **B**, il programma deve chiedere all'utente un numero X compreso fra 1 e N: quindi si deve selezionare il direttorio corrispondente al numero X specificato dall'utente e di tale direttorio si deve riportare su standard output il nome assoluto (*) e la prima linea di tutti i file leggibili il cui nome abbia terminazione .S.

NOTA BENE: Per svolgere la fase B si può sia reinvocare il file comandi ricorsivo FCR sia prevedere una soluzione alternativa.

LO STUDENTE DEVE RIPORTARE IN MODO CHIARO IN UN COMMENTO QUALE DELLE DUE ALTERNATIVE HA SEGUITO.

(*) Precisazione detta durante lo svolgimento della prova!

FCP.sh

```
#!/bin/sh
#Soluzione della Prima Prova in itinere del 12 Aprile 2013
#versione 1 che usa un file temporaneo e non usa FCR.sh per la seconda fase

case $# in
3)    case $1 in
        /*) if test ! -d $1 -o ! -x $1
            then
                echo $1 non direttorio
                exit 1
            fi;;
        *)  echo $1 non nome assoluto; exit 2;;
    esac;;
*)    echo Errore: Usage is $0 dirass stringa numero
    exit 3;;
esac

#puo' avere senso controllare che il secondo parametro non contenga un carattere /
case $2 in
/*)    echo Errore: $f non in forma relativa semplice
    exit 3;;
*)    ;;
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

#Iniziamo la prima fase
#creiamo un file conta$$ il cui nome viene passato come argomento
> /tmp/conta$$

echo fase A
#invochiamo il file comandi ricorsivo con solo i primi due parametri (il terzo non
serve) e con il nome del file temporaneo
FCR.sh $1 $2 /tmp/conta$$
```

```

#N.B. Andiamo a contare le linee del file /tmp/conta$$
D=`wc -l < /tmp/conta$$`
echo Il numero di direttori totali della gerarchia che soddisfano la specifica = $D
#Stampiamo (come richiesto dal testo) i loro nomi assoluti
cat /tmp/conta$$
#Controlliamo se dobbiamo passare alla seconda fase
if test $D -gt $3
then
#Passiamo ora alla seconda fase che non viene risolta in questo caso con il file
ricorsivo FCR.sh, ma direttamente in questo file comandi
#chiediamo all'utente il numero X
echo -n "Dammi il numero X che deve essere compreso fra 1 e $3: "
read X
#Controllo X
expr $X + 0 > /dev/null 2>&1
N1=$?
if test $N1 -ne 2 -a $N1 -ne 3
then echo numerico $X #siamo sicuri che numerico
    if test $X -lt 1 -o $X -gt $3
    then echo $X non compreso
        rm /tmp/conta$$ #poiche' stiamo uscendo a causa di un errore, cancelliamo
il file temporaneo!
        exit 6
    fi
else
    echo $X non numerico
    rm /tmp/conta$$ #poiche' stiamo uscendo a causa di un errore, cancelliamo il file
temporaneo!
    exit 7
fi

echo fase B
#selezioniamo direttamente il nome del direttorio richiesto
d=`head -$X < /tmp/conta$$ | tail -1`
echo direttorio selezionato $d
cd $d
for i in *
do
if test -f $i -a -r $i
then
case $i in
*.$2) #stampiamo nome assoluto
    echo Nome assoluto file `pwd`/$i
    #stampiamo prima linea del file
    head -1 < $i;;
*) ;;
esac
fi
done
fi

#da ultimo eliminiamo il file temporaneo
rm /tmp/conta$$

```

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

cd $1
#la variabile trovato ci serve per capire se abbiamo trovato almeno un file
leggibile con la terminazione richiesta
trovato=0

for i in *
do
    #controlliamo solo i nomi dei file leggibili!
    if test -f $i -a -r $i
    then
        case $i in
            *.$2) trovato=1;;
            esac
        fi
    done
    if test $trovato -eq 1
    then
        #abbiamo trovato un direttorio che soddisfa le specifiche e quindi inseriamo il suo
        nome assoluto nel file temporaneo
        pwd >> $3
        fi
        for i in *
        do
            if test -d $i -a -x $i
            then
                #chiamata ricorsiava
                FCR.sh `pwd`/$i $2 $3
            fi
        done
    done
done
```

13/04/2012

La parte in Shell deve prevedere un solo parametro:

- deve essere il nome assoluto di un direttorio che identifica una gerarchia (G) all'interno del file system.

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in due fasi (Fasi A e B).

Il programma nella fase **A** deve esplorare la gerarchia G specificata - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare globalmente i livelli di cui è costituita la gerarchia G (inclusa la radice).

Al termine dell'intera esplorazione ricorsiva di G, si deve riportare sullo standard output il numero di livelli totali della gerarchia G. Quindi il programma passa nella fase B.

In questa fase **B**, il programma deve chiedere all'utente un numero pari compreso fra 1 e il numero di livelli totali calcolati nella faseA: quindi si deve reinvocare FCR in modo da cercare nella gerarchia G specificata (inclusa la radice) tutti i direttori che si trovano al livello corrispondente al numero pari specificato dall'utente e per ognuno di essi si devono visualizzare TUTTE le informazioni associate agli elementi presenti esclusi i file nascosti.

NOTA BENE: FCR dovrà prevedere dei parametri ulteriori per differenziare l'esecuzione delle diverse Fasi.

```
#!/bin/sh
```

```
#controllo sul numero di parametri: deve essere passato un solo parametro
#facciamo anche contestualmente il controllo se e' una directory traversabile
```

```
case $# in
```

```
1)    case $1 in
```

```
    /*) if test ! -d $1 -o ! -x $1
```

```
        then
```

```
            echo $1 non directory o non attraversabile
```

```
            exit 1
```

```
        fi;;
```

```
    *) echo $1 non nome assoluto; exit 2;;
```

```
esac;;
```

```
*)    echo Errore: Usage is $0 dirass
```

```
exit 3;;
```

```
esac
```

```
PATH=`pwd`: $PATH
```

```
export PATH
```

```
conta=0 #valore che andremo a scrivere nel file temporaneo
```

```
fase=A
```

```
echo $conta > /tmp/tmpContaLivelli #scriviamo il valore di conta (cioe' 0) nel file
temporaneo
```

```
#invochiamo per la fase A: bastano solo 3 parametri
```

```
echo Stiamo iniziando la fase A
```

```
FCR.sh $1 $conta $fase
```

```
read tot < /tmp/tmpContaLivelli
```

```
echo Il numero di livelli totali della gerarchia = $tot
```

```
#passiamo alla seconda fase
```

```
echo Stiamo iniziando la fase B
```

```
livello= #variabile per leggere il valore inserito dall'utente
```

```
#adesso chiediamo all'utente quale livello vuole visualizzare
```

```
echo -n "dammi il numero di livello pari che vuoi visualizzare (il numero deve
essere compreso fra 1 e $tot " > /dev/tty
```

```
read livello #chiediamo all'utente il numero di livello
```

```
case $livello in
```

```
*[!0-9]*) echo NON hai fornito un numero o non positivo
```

```
    #rimuoviamo il file temporaneo
```

```
    rm /tmp/tmpContaLivelli
```

```
    exit 4;;
```

```
*) #numero giusto;;
```

```
esac
```

```
if test $livello -ge 1 -a $livello -le $tot
```

```
then
```

```
    echo $livello compreso OK
```

```
    if test `expr $livello % 2` -eq 0
```

```
    then
```

```
        echo $livello pari
```

```
    else
```

```
        #questo sarebbe un errore per il testo considerato, ma lo teniamo
```

```
buono per fare un numero maggiore di prove
```

```
        echo $livello dispari
```

```
    fi
```



```

else
    echo $livello NON compreso
    #rimuoviamo il file temporaneo
    rm /tmp/tmpContaLivelli
    exit 5
fi
fase=B
#passiamo alla seconda chiamata
echo $conta > /tmp/tmpContaLivelli #riscriviamo il valore di conta (cioe' 0) nel
file temporaneo

FCR.sh $1 $conta $fase $livello #NOTA BENE: passiamo un parametro in piu'!
#rimuoviamo il file temporaneo
rm /tmp/tmpContaLivelli
echo FINITO

```

FCR.sh

```

#!/bin/sh
#file comandi ricorsivo che torna il numero totale di livelli
#contati fino a quel punto

cd $1
#il primo livello verra' contato come livello 1
conta=`expr $2 + 1` #nota bene questa variabile conta NON e' quella dello script
principale, ma e' una variabile conta LOCALE a questo script
for i in *
do
    if test -d $i -a -x $i
    then
        FCR.sh `pwd`/$i $conta $3 $4
        ret=$?
        read prec < /tmp/tmpContaLivelli #leggiamo dal file temporaneo il
valore del numero di livello
        if test $ret -gt $prec           #se il valore calcolato in
ricorsione e' maggiore, allora
        then
            echo $ret > /tmp/tmpContaLivelli #aggiorniamo il numero di livello
raggiunto
        fi
    fi
done
if test $3 = B #solo se siamo nella fase B bisogna fare questa parte di codice,
mentre quella precedente e' comune ad entrambe le fasi!
then
    if test $4 -eq $conta #se nella seconda fase sono in una directory che
corrisponde al livello cercato, dobbiamo stampare le info richieste
    then
        echo Adesso visualizzo il contenuto della directory `pwd` che fa
parte del livello $4
        ls -l #se dovessimo visualizzare tutte le info inclusi i file
nascosti sarebbe ls -la oppure inclusi i file nascosti a parte . e .. sarebbe ls -
la
    fi
fi
exit $conta #N.B. codice comune a tutte le fasi!

```

29/04/2011

La parte in Shell deve prevedere tre parametri:

- il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (G) all'interno del file system;
- il secondo parametro deve essere il nome relativo semplice di un file (F);
- il terzo parametro deve essere considerato un numero intero positivo (N).

Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in generale in tre fasi (Fasi A, B e C). Il programma deve cercare nella gerarchia G specificata (inclusa la radice) - tramite un file comandi ricorsivo, **FCR.sh** - tutti i direttori che contengono un file che ha nome relativo F: si riporta il nome assoluto di tali direttori sullo standard output e si continua globalmente i file che soddisfano la condizione indicata (Fase A).

Al termine dell'intera esplorazione ricorsiva di G, si deve riportare sullo standard output il numero totale di file (NF) che soddisfano la condizione indicata. In caso NF sia minore di N, il programma passa alla Fase B: si deve reinvocare FCR in modo da cercare nella gerarchia G specificata (inclusa la radice) tutti i direttori che NON contengono un file con nome relativo F, riportando sullo standard output il nome assoluto dei direttori trovati e creando un file vuoto con nome relativo F all'interno di ognuno.

Quindi, se è stato necessario eseguire la Fase B, si deve passare alla Fase C in cui si deve verificare la situazione risultante invocando nuovamente FCR come per la Fase A.

NOTA BENE: FCR dovrà prevedere un parametro ulteriore rispetto a G, F e N per differenziare l'esecuzione delle diverse Fasi.

FCP.sh

```
#!/bin/sh
#Controllo che ci siano esattamente 3 parametri
case $# in
3) ;;
*) echo "Numero sbagliato di parametri: ci vogliono 3 parametri (nome ass dir, nome
relativo file da cercare e numero intero positivo)"
exit 1 ;;
esac

G=$1
F=$2
N=$3

#Controllo che sia il nome assoluto di direttorio
case $G in
/*) ;;
*) echo "$G non e' in forma assoluta"
exit 2 ;;
esac
#Controllo che sia un direttorio esplorabile
if test ! -d $G -o ! -x $G
then
    echo "$G non e' un direttorio o non e' esplorabile"
    exit 3
fi
```

```

#Controllo che F sia un nome relativo
case $F in
    /*) echo "$F non e' in forma relativa"
        exit 4;;
    *);;
esac

#Controllo che sia un numero > 0
test $N -gt 0 1>/dev/null 2>/dev/null
case $? in
    0) ;;
    *) echo "$N non e' un numero o non e' strettamente > 0"
        exit 5;;
esac

PATH=`pwd`: $PATH
export PATH

#creiamo un file temporaneo che serve solo per riportare le cose poi alla
situazione precedente: non richiesto dalla soluzione
> /tmp/perPulisci

#Fase A
#usiamo un file temporaneo che passeremo come ultimo parametro
> /tmp/esame$$
FCR.sh $G $F A /tmp/esame$$
#conto il numero di righe sul file temporaneo=numero di file trovati
NF=`wc -l < /tmp/esame$$`
echo "File trovati fase A: $NF"

#Se il numero file < N eseguo la fase B e C
if test $NF -lt $N
then
    FCR.sh $G $F B #nella fase B il file temporaneo non serve e quindi possiamo
    non passarlo!
    > /tmp/esame$$ #azzeriamo il file temporaneo
    FCR.sh $G $F C /tmp/esame$$
    #Riconto il numero di righe per controllare il numero dei file
    NF=`wc -l < /tmp/esame$$`
    echo "File trovati fase C: $NF"
fi
#da ultimo cancelliamo il file temporaneo
rm /tmp/esame$$

```

FCR.sh

```

#!/bin/sh
cd $1

#se il terzo parametro: flag=A o C
#cerco i file che si chiamano F e se ne trovo uno scrivo il nome assoluto del
direttorio
#nome assoluto del file trovato su un file temporaneo
if test "$3" = A -o "$3" = C
then
    #echo fase $3

```

```

if test -f $2
then
    echo "Trovato $2 in `pwd`"
    echo "`pwd`/$1" >> $4
fi
fi

#se il terzo parametro e' uguale a B invoco la seconda fase
#dove cerco i direttori in cui non ci sono file che si chiamano F
if test "$3" = B
then
    #echo fase $3
    trovato=0
    if test -f $2
    then
        trovato=1
    fi
    #se non trovo un file scrivo il nome del direttorio sullo stdout
    if test $trovato -eq 0
    then
        echo "In `pwd` non c'e' un file che si chiama $2 e quindi lo creo"
        > $2
        echo "`pwd`/$2" >> /tmp/perPulisci
    fi
fi

#se trovo un direttorio esplorabile
#richiamo la ricorsione con il terzo parametro come flag
for i in *
do
    if test -d $i -a -x $i
    then
        #echo ricorsione
        FCR.sh `pwd`/$i $2 $3 $4 #nella fase B $4 sara' nullo
    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);
- 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
```