

# Esempi di esercizi di esame passati

Suggerimento: vi ricordo l'esistenza dei seguenti operatori per le variabili bash

`${#VAR}` stringa che contiene la lunghezza del contenuto della variabile

`${VAR:offset}` sottostringa che parte dal offset-esimo carattere del contenuto della variabile di nome VAR

`${VAR:offset:length}` sottostringa lunga length che parte dal offset-esimo carattere del contenuto della variabile di nome VAR

Esempi:

```
VAR="ciao"; echo ${VAR:0:1};
```

*viene messo in output c, cioè il primo carattere*

```
for ((INDICE=0; $INDICE<${#VAR}; INDICE=$((INDICE+1))) ; do echo  
${VAR:$INDICE:1}; done;
```

*mette in output tutti i caratteri uno per volta*

## Esercizio Esame Pratica - 146 - **lancia.sh**

Implementare due script bash **lancia.sh** e **figlio.sh**. Lo script **lancia.sh** imposta tutto quello che serve e poi lancia in esecuzione lo script **figlio.sh** (lo penso come se avesse indice  $i=1$ ) **SENZA passargli argomenti a riga di comando**, e poi termina.

Il primo script **figlio.sh** ( $i=1$ ) a sua volta lancia lo script **figlio.sh** ( $i=2$ ) **SENZA** passargli argomenti **a riga di comando** e poi termina.

Ciascuno script figlio ( $i$ -esimo) lancia in esecuzione lo script **figlio.sh** ( $(i+1)$ -esimo) **SENZA** passargli argomenti **a riga di comando**, e poi termina.

La sequenza di script uno figlio dell'altro termina quando in totale sono stati messi in esecuzione 10 script **figlio.sh** più lo script iniziale **lancia.sh**.

Ciascun **i-esimo** script **figlio.sh** deve mettere in output il proprio **pid**.

Dopo di questo, se **i** è maggiore di 10 allora lo script termina; se invece **i** è minore o uguale a 10 allora lo script lancia lo script **figlio.sh** e poi termina.

# SOLUZIONE Esercizio Esame Pratica - 146 - lancia.sh

## **lancia.sh**

```
#!/bin/bash
export VARGLOBALEINDICEATTUALE="1"
export VARGLOBALEINDICEMASSIMO="10"
./figlio.sh
```

## **figlio.sh**

```
#!/bin/bash
echo "$$"
if (( ${VARGLOBALEINDICEATTUALE} < ${VARGLOBALEINDICEMASSIMO} )) ; then
    ((VARGLOBALEINDICEATTUALE=${VARGLOBALEINDICEATTUALE}+1))
    ./figlio.sh
fi
```

## Esercizio Esame Pratica - 152 - 2di3.sh

Implementare uno script 2di3.sh che legge una dopo l'altra le righe di testo del file /usr/include/stdio.h .  
Per ciascuna riga lo script mette sullo standard output una riga che contiene il solo secondo carattere della terza parola di quella riga.  
Se in una riga del file letto ci sono meno di 3 parole allora lo script non stampa nulla relativo a quella riga.  
Se nella terza parola di una riga del file ci sono meno di 2 caratteri allora lo script non stampa nulla relativo a quella riga.

# SOLUZIONE Esercizio Esame Pratica - 152 - 2di3.sh

## script 2di3.sh

```
#!/bin/bash

while read uno due tre quattro ; do
    if [[ $tre != "" ]] ; then
        car=${tre:1:1}
        if [[ $car != "" ]] ; then
            echo $car
        fi
    fi
done < /usr/include/stdio.h
```

**NB: questa soluzione presenta un errore che va corretto mettendo il comando**  
**echo "\$car" al posto di echo \$car**  
**Perché occorre mettere quel particolare quoting?**

## Esercizio Esame Pratica - 164 - estrai.sh

Scrivere un file di testo **input1.txt** in cui in ciascuna riga c'è una parola, seguita da una virgola, seguita da un numero intero maggiore o uguale a zero, seguito da una virgola, seguito da un'altra parola.

Ad esempio, il file **input1.txt** passato assieme al testo dell'esercizio contiene:

```
gatto,52,vacca  
alfa,31,triangolo  
orto,9,bici  
zappa,69,guzzi  
disco,12,celentano
```

Realizzare uno script **estrai.sh** che stampa a video le righe lette dal file **input1.txt** ma senza i numeri in mezzo a ciascuna riga e senza la virgola che segue il numero. Infine, stampa a video la somma dei numeri che non ha visualizzato.

Qui sotto ecco cosa deve apparire a video usando il file **input1.txt** di esempio.

```
gatto, vacca  
alfa, triangolo  
orto, bici  
zappa, guzzi  
disco, celentano  
173
```

# SOLUZIONE Esercizio Esame Pratica - 164 - estrai.sh

```
#!/bin/bash
```

```
IFS=$',\n'
```

```
SOMMA=0
```

```
while read PRIMA SECONDA TERZA ; do
```

```
    ( (SOMMA=${SOMMA}+${SECONDA}) )
```

```
    echo "${PRIMA},${TERZA}"
```

```
done < input1.txt
```

```
echo ${SOMMA}
```

## Esercizio Esame Pratica - **158** - **asterischi**

Realizzare uno script `asterischi.sh` che conta quante sono le righe che contengono almeno un asterisco `*` in tutti i file il cui nome termina con `.h` nella parte di filesystem a partire dalla directory `/usr/include/`

Lo script deve stampare sul proprio standard error il numero delle righe contate.



# SOLUZIONE Esercizio Esame Pratica - 158 - asterischi

```
#!/bin/bash
```

```
echo ` find /usr/include/ -type f -name "*.h" -exec grep -H '*' '{*}' \; | wc -l `
1>&2
```

Una soluzione un po' più leggibile è questa:

```
for name in ` find /usr/include/ -type f -name "*.h" ` ; do
    grep -H '*' ${name} | wc -l    1>&2
done
```

## Esercizio Esame Pratica - 160 - DIFFICILE - nAnB

Realizzare uno script `nAnB.sh` che prende come argomento una stringa di testo e verifica che la stringa sia formata esattamente da N caratteri A seguiti da N caratteri B, con un N qualunque maggiore di zero.

Se la stringa corrisponde ai requisiti lo script scrive sullo standard output la parola "vero" seguita dal Numero N individuato.

Altrimenti scrive "falso".

Ad esempio, passando come argomento:

AAABBB vedo in output "vero N=3"

ADABBB vedo in output "falso" perché c'è carattere diverso da A e da B

AAB vedo in output "falso"

AabB vedo in output "falso"

**Realizzare lo script richiesto completando lo script base allegato.**

Suggerimento: vi ricordo l'esistenza dei seguenti operatori per le variabili bash

`${VAR:offset}` sottostringa che parte dal offset-esimo carattere del contenuto della variabile di nome VAR

`${VAR:offset:length}` sottostringa lunga length che parte dal offset-esimo carattere del contenuto della variabile di nome VAR

Esempio:

`VAR="ciao"; echo ${VAR:0:1}` viene messo in output c

# SOLUZIONE Esercizio Esame Pratica - 160 - DIFFICILE - nAnB

```
#!/bin/bash
if (( $# != 1 )) ; then echo "manca 1' argomento a riga di comando";      exit 1; fi
NA=0;  NB=0; RESULT=0;
STRINGA=$1
LUNGHEZZASTRINGA=${#STRINGA}
NLETTI=0
# cerco le A e la prima B dopo qualche A
while (( ${NLETTI} < ${LUNGHEZZASTRINGA} )) ; do # ci sono altri caratteri da leggere
    # prendo il carattere in posizione NLETTI (all'inizio NLETTI 0)
    CAR=${STRINGA:${NLETTI}:1}
    (( NLETTI=${NLETTI}+1 ))
    if [[ ${CAR} == "A" ]] ; then
        ((NA=$NA+1)) ;
    else
        if [[ ${CAR} == "B" && ${NA} -gt 0 ]] ; then
            NB=1
            break;
        fi
        # arrivo qui solo se ho letto un carattere diverso da A e B oppure se ho letto una B senza prima avere letto delle A
        # quindi in ogni caso termino senza avere verificato la simmetria
        echo falso caso 1
        exit 1
    fi
done

# cerco il resto delle B
while (( ${NLETTI} < ${LUNGHEZZASTRINGA} )) ; do # ci sono altri caratteri da leggere
    # prendo il carattere in posizione NLETTI (all'inizio NLETTI 0)
    CAR=${STRINGA:${NLETTI}:1}
    (( NLETTI=${NLETTI}+1 ))
    if [[ ${CAR} == "B" ]] ; then
        ((NB=$NB+1)) ;
    else
        # arrivo qui solo se ho letto un carattere diverso da B e quindi termino senza avere verificato la simmetria
        echo falso caso 2
        exit 2
    fi
done

# verifico di avere letto tante B quante A
if [[ $NA -gt 0 && $NA == $NB ]] ; then echo vero  N=$NA; exit 0
Else      echo falso caso 3:      exit 3; fi
```

# Esercizio Esame Pratica - 148 - concatena.sh

Implementare uno script bash chiamandolo concatena.sh

Lo script concatena.sh scrive il proprio output nel file output.txt

Se il file output.txt esiste già lo script lo svuota eliminando il contenuto.

Lo script concatena.sh cerca tutti i files che stanno nella directory /usr/include/ e il cui nome inizia con la stringa std, ma non cerca nelle sottodirectory.

Per ciascuno dei file trovati, lo script concatena.sh aggiunge al file output.txt la seconda, terza e quarta riga di quel file trovato.

Suggerimento: vi ricordo l'esistenza dei seguenti operatori per le variabili bash

<code>\${#VAR}</code>	stringa che contiene la lunghezza del contenuto della variabile
<code>\${VAR:offset:length}</code>	sottostringa lunga length che parte dal offset-esimo carattere del contenuto della variabile di nome VAR

Il primo carattere della stringa ha indice 0

# SOLUZIONE Esercizio Esame Pratica - 148 - concatena.sh

```
#!/bin/bash
```

```
echo -n "" > output.txt
for name in /usr/include/std* ; do
    if [[ -e ${name} ]] ; then
        if exec {FD}<${name} ; then
            NUM=0
            while read -u ${FD} RIGA ; do
                ((NUM=${NUM}+1));
                if (( $NUM > 1 )) ; then
                    echo "$RIGA" >> output.txt
                fi
                if (( $NUM >= 4 )) ; then
                    break;
                fi
            done
            exec {FD}>&-
        fi
    fi
done
```