

Lezione 6

- Progettare uno script che prende in input come parametri i nomi di due directory e copia tutti i file della prima nella seconda, trasformando tutte le occorrenze della stringa *SP* in *SU* in ogni file.

```
if test $# -ne 2
then
    echo 'Utilizzo dello script: SP_to_SU <dir1> <dir2>'
    exit 1
fi

if ! test -d $1 -a -d $2 # Gestione degli errori.
then
    echo "$1 e $2 devono essere delle directory"
    exit 2
fi

for i in $1/*
do
    sed "s/SP/SU/g" "$i" > $2/'basename $i'
done

exit 0
```

Alternativamente, per ricercare ricorsivamente tutti i file regolari nella prima directory, si può utilizzare il comando `find` nel ciclo `for` come segue:

```
if test $# -ne 2
then
    echo 'Utilizzo dello script: SP_to_SU <dir1> <dir2>'
    exit 1
fi

if ! test -d $1 -a -d $2 # Gestione degli errori.
then
    echo "$1 e $2 devono essere delle directory"
    exit 2
fi

for i in `find $1 -type f`
do
    sed "s/SP/SU/g" "$i" > $2/'basename $i'
done

exit 0
```

- Progettare uno script *drawsquare* che prende in input un parametro intero con valore da 2 a 15 e disegna sullo standard output un quadrato (utilizzando i caratteri `+`, `-` e `|`) come nel seguente esempio:

```

> drawsquare 4
+---+
|   |
|   |
+---+

if test $# -ne 1
then
    echo 'Utilizzo dello script: drawsquare <n>'
    exit 1
fi

if test $1 -lt 2 -o $1 -gt 15
then
    echo 'Il parametro deve essere un numero >1 e <16'
    exit 2
fi

x=$1
y=$1

while test $y -gt 0
do
    while test $x -gt 0
    do
        if test $x -eq 1 -o $x -eq $1
        then
            if test $y -eq 1 -o $y -eq $1
            then
                echo -n "+"
            else
                echo -n "|"
            fi
        else
            if test $y -eq 1 -o $y -eq $1
            then
                echo -n "-"
            else
                echo -n " "
            fi
        fi
        x=$((x-1))
    done
    x=$1
    y=$((y-1))
    echo
done

exit 0

```

- *Progettare uno script che prende in input come parametro il nome di una directory e cancella tutti i file con nome **core** dall'albero di directory con radice la directory parametro.*

```
if test $# -ne 1
then
    echo 'Utilizzo dello script: rmcore <dir>'
    exit 1
fi

if ! test -d $1 # Gestione degli errori.
then
    echo "$1 deve essere una directory"
    exit 2
fi

find $1 -name core -exec rm {} \; 2>/dev/null

exit 0
```