1 Esercizio 1

1.1 Esplorate il vostro file system. Qual è il pathname della vostra home directory?

Da terminale:

```
> cd
> pwd
/Users/Aldo
```

1.2 Visualizzate i file della vostra home directory ordinati in base alla data di ultima modifica.

Da terminale:

```
> ls -t
```

1.3 Che differenza cè tra i comandi cat, more, tail?

- **cat**: concatena i file, stampando il risultato dell'operazione.
- **more**: filtro che serve a visualizzare sullo schermo un flusso di testo, una pagina per volta.
- **tail**: stampa su standard output le ultime 10 righe dei file, forniti come argomenti.

1.4 Trovate un modo per ottenere lelenco delle subdirectory contenute ricorsivamente nella vostra home.

Da terminale:

```
> ls -R
```

1.5 I seguenti comandi che effetto producono? Perchè?

```
> cd
> mkdir d1
> chmod 444 d1
> cd d1
```

- **cd**: la directory corrente diventa quella della home
- **mkdir d1**: crea la directory di nome "dl" nella directory corrente
- **chmode 444 d1**: assegna permessi `r--r--r--` a d1
- **cd d1**: la directory corrente dovrebbe diventare d1 (se si hanno i permessi)

2 Esercizio 2

2.1 Scrivete un unico comando (pipeline) per: 1) copiare il contenuto della directory dir1 nella directory dir2; 2) fornire il numero di file (e directory) a cui avete accesso, contenuti ricorsivamente nella directory studenti (si può utilizzare "ls -R" e il comando "find"); 3) fornire la lista dei file della home directory il cui nome è una stringa di 3 caratteri seguita da un numero.

1) `> cp ~r dir1 dir2`
2) `> find ~ 2>/dev/null | wc -l`
3) `> ls ~/???[0-9]`

2.2 Qual è la differenza tra i seguenti comandi?

```
> ls
> ls | cat
> ls | more
```

- `ls`: stampa la lista dei file contenuti nella directory corrente, disposti in righe e colonne in ordine alfabetico.
- `ls | cat`: stampa la lista dei file contenuti nella directory corrente disposti su un unica colonna in ordine alfabetico.
- `ls | more`: stampa la lista dei file contenuti nella directory corrente disposti in ordine alfabetico, in colonna e organizzando l'output su schermo in pagine.

2.3 Quale effetto producono i seguenti comandi?

```
> uniq < file (dove file ‘e il nome di un file)
> who | wc -l
> ps -e | wc -l
```

- `uniq < file`: stampa il contenuto del file, sostituentdo le linee adiacenti uguali con un'unica occorrenza.
- `who | wc -l`: Stampa il numero di utenti collegati al sistema.
- `ps -e | wc -l`: Stampa il numero di processi + 1; il +1 la linea intestazione prodotta da ls.

2.4 Ridefinire il comando "rm" in modo tale che non sia chiesta conferma prima della cancellazione dei file

```
> alias rm='rm -f'
```

2.5 Definire il comando "rmi" (rm interattivo) che chiede conferma prima di rimuovere un file.

```
> alias rmi='rm -i'
```

2.6 Sapendo che il comando "ps" serve ad elencare i processi del sistema, scrivere una pipeline che fornisca in output il numero di tutti i processi in esecuzione.

```
> ps -e --no-headers | wc -l
```

2.7 Salvare in un file di testo loutput dellultimo evento contenente il comando "ls".

```
> ls | more > Scrivania/relazioni/file_out_ls.txt
```

2.8 Scrivere un comando che fornisce il numero dei comandi contenuti nella history list

```
> history | wc -l
```

2.9 Scrivere un comando che fornisce i primi 15 comandi della history list

```
> history | head -15
```

2.10 Quali sono i comandi Unix disponibili nel sistema che iniziano con "lo"?

```
> lo + il tasto 'tab' 2 volte
```

2.11 Fornire almeno due modi diversi per ottenere la lista dei file della vostra home directory il cui nome inizia con "al".

```
> ls ~/al il tasto 'tab' + il tasto 'tab'
> ls al*
```

2.12 Qual è leffetto dei seguenti comandi?

```
> ls -R || (echo file non accessibili > tmp)
> (who | grep rossi) && cd ~rossi
> (cd / ; pwd ; ls | wc -l )
```

- `ls -R || (echo file non accessibili > tmp)`: stampa la lista ricorsiva dei file contenuti nella directory corrente; nel caso in cui ls -R fallisca, stampa il messaggio 'file non accessibili' nel file tmp nella directory corrente.
- `(who | grep rossi) && cd ~rossi`: la directory corrente cambia in quella dell'utente rossi (se questo collegato al sistema)
- `(cd / ; pwd ; ls | wc -l )`: cambia la directory corrente in /, la stampa (con 'pwd') e stampa il numero di elementi contenuti nella dir corrente

3 Esercizio 3

3.1 Qual 'e leffetto del seguente comando, dove file è il nome di un file?

- `sort file >file`: Questo comando cancella il contenuto del file, perch la ridirezione dell'output cancella il contenuto del file prima che venga eseguito il 'sort'.

3.2 Fare alcuni esperimenti per scoprire qual è leffetto del comando tr str1 str2 se le stringhe str1 e str2 hanno lunghezze diverse.

I caratteri di str1 sono tradotti (uno alla volta) nei caratteri di str2, nell'ordine degli array di stringa.

3.3 Scrivere un comando per sostituire tutti i caratteri alfanumerici nellinput con un carattere Tab, in modo che non compaiano più Tab consecutivi.

```
> tr -s A-Za-z0-9 '\t'
```

3.4 Scrivere una pipeline che permetta di scoprire se ci sono linee ripetute in un file.

```
> cat file | sort | uniq -c | sort -k1,2 -t' ' |tail -1 |
| tr -s ' ' | cut
```

3.5 Visualizzare su standard output, senza ripetizioni, lo user ID di tutti gli utenti che hanno almeno un processo attivo nel sistema.

```
io:~$ ps -el --no-headers | tr -s ' ' | cut -d' ' -f3 | sort -n | uniq
```

3.6 Scrivere un comando awk per stampare il numero massimo di campi di una linea in un dato file.

```
> awk '{if(NF>max) max=NF} END {printf
"Numero massimo di campi in una linea: %d\n", max}' Scrivania/test.txt
```

4 Esercizio 4

4.1 Progettare uno script drawsquare che prende in input un parametro intero con valore da 2 a 15 e disegna sullo standard output un quadrato.

Esempio:

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

Codice:

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

if test $1 -le 2 -o $1 -ge 51
   then
   echo 'Il parametro deve essere un numero in [3;50]'
   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
```

4.2 Progettare uno script che prende in input come parametro il nome di una directory e cancella tutti i file con nome core dallalbero di directory con radice la directory parametro.

Codice:

```
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
```

4.3 Progettare uno utility-script processi per la gestione user-friendly dei processi in memoria. Le operazioni di base sono gestite tramite un semplice menu testuale con input da tastiera.
- Visualizzazione processi di un utente selezionato (PID, CPU e riga comando che lo ha generato)
- Implementare una versione di top minimale, con un ordinamento in base alluso CPU, e con la fotografia del momento, senza aggiornamenti.
- Possibilità di kill -9 su un processo utente, su tutti i processi (sempre con un comodo menu)
- Ogni funzione attivata ritorna nel menu principale di scelta

Codice:

```
echo "MENU: ";
echo "1) Visualizza i processi di un utente selezionato";
echo "2) Top minimale (con ordinamento in base alla CPU)";
echo "3) Kill -9 su un processo utente, su tutti i processi
    ";
echo " ";
echo "Scegli una delle opzioni: ";

read opt;
if [ $opt -lt 1 -o $opt -gt 4 ]
then
        echo "Scelta non valida!";
fi

case $opt in
        1) echo "inserisci il nome di un utente:"
                read username; ps -u $username ;;
        2) top ;;
        3) echo "inserisci il PID del processo da eliminare
            :";
                read pid;kill -9 $pid ;;
esac
```