

## RECUPERI: Istruzioni ed Esercizi

E` possibile perdere e recuperare fino ad un massimo di 4 attivita` su 8.

Chi ha perso uno o più seminari, fino a 4, può recuperare in due passi:

1. studiando le slide dei seminari (disponibili in:  
[https://github.com/bertocco/abilita\\_info\\_units\\_2021](https://github.com/bertocco/abilita_info_units_2021) o sul moodle del corso)  
ed eseguendo gli esercizi proposti nelle slide

PIU`

2. svolgendo per ciascuno dei seminari gli esercizi proposti di seguito e mandando le soluzioni (come lista di comandi o come script), in formato testo ed eseguibili, alla docente all'e-mail:

sara 'dot' bertocco 'at' inaf 'dot' it  
entro il 28 febbraio 2021

La verifica della correttezza delle soluzioni consisterà in una revisione sommaria degli script più l'esecuzione degli script stessi da parte della docente.

### TESTI DEGLI ESERCIZI DA SVOLGERE per ciascuna lezione.

Lo studente deve eseguire e consegnare solo gli esercizi della/e lezione/i che deve recuperare.

#### ##### BASH 1

1) Visualizzare il valore delle variabili d'ambiente

PATH, PS1, USERNAME, PWD

Come vengono modificate queste variabili quando eseguite il comando "cd <nomedirectory>" ?

Modificare il prompt dei comandi in modo che visualizzi:

Nomeutente:directorycorrente>

Nomeutente:ora:homedirectory>

Facoltativo:

Nomeutente@nomemacchina:directorycorrente>

2) Scrivere i comandi per:

creare il seguente albero nel file system:

esercizio2/

esercizio2/primofile.txt

esercizio2/secondofile.txt

esercizio2/terzofile.sh

esercizio2/folder1  
esercizio2/folder2  
esercizio2/folder3/primofileinfolder3.txt

Entrare nella radice dell'albero,  
listare tutti i file  
creare un nuovo file vuoto  
creare un link simbolico al nuovo file  
controllare il valore di ritorno del comando appena eseguito  
cancellare tutto l'albero

## ##### BASH 2

1) Usare il comando `which` per verificare l'esistenza e il percorso in cui si trova l'eseguibile `find`. Usare il comando `ls` per visualizzare il contenuto della directory in cui e' collocato l'eseguibile `find`.

2) Definire i seguenti alias per la propria shell:

- `cp` come alias di `cp -i`
- `mv` come alias di `mv -i`
- `rm` come alias di `rm -i`
- `bye` che effettua l'uscita dalla shell

Fare in modo che gli alias siano disponibili ogni volta che una shell viene attivata.  
(Suggerimento: `man bash` cercare `bashrc`)

## ##### BASH 3

1) Creare il file `input.txt` utilizzando il comando `cat` con il seguente contenuto:

You're packing a suitcase for a place  
None of us has been  
A place that has to be believed to be seen  
You could have flown away  
A singing bird in an open cage  
Who will only fly, only fly for freedom

con `grep` e `sort` selezionare tutte le linee che contengono esattamente la parola `place`, ordinarle lessicograficamente e farle stampare sullo standard output.

2) Preparare uno script che, dentro ad un ciclo infinito, scrive una stringa "Sono un output standard" su standard output ed una stringa "sono un messaggio di log" in un file in modalità "append".

Fornire il comando per rendere eseguibile lo script, fornire il comando per lanciare lo script in modo da visualizzare lo standard output sia sulla console che in un file.

#### ##### BASH 4

1) Preparare uno script che:

- Visualizzi un messaggio di benvenuto contenente il suo nome;
- Visualizzi un help in cui dice che deve essere chiamato con 3 parametri
- Visualizzi i valori del primo e terzo parametro;
- Visualizzi il numero totale di argomenti;

2) Fare le seguenti cose:

creare una directory my\_work\_area

creare 3 sottodirectory ciascuna contenente uno script:

la prima contiene uno script che dice "la data di oggi è ...." e la data corrente al posto dei puntini;

la seconda contiene uno script che chiede all'utente "come ti chiami", acquisisce la risposta e scrive sull'output "Ciao .... come va?" con il nome dell'utente acquisito al posto dei puntini.

Preparare infine uno script che presenta all'utente un menù:

"Salve,  
se vuoi sapere che giorno è oggi premi 1  
se vuoi dirmi come ti chiami premi 2  
se vuoi un messaggio di benvenuto premi 3  
"

acquisisce l'input fornito dall'utente e in base alla scelta fatta esegue uno degli script precedentemente preparati.

#### ##### PYTHON 1

1) Scrivere una funzione che calcola la somma dei primi n numeri interi. Acquisire il numero n inserito da tastiera, chiamare la funzione passando il parametro n, stampare il risultato con un opportuno messaggio.

2) Contare il numero di caratteri, di interi e di altri simboli contenuti in due stringhe fornite in input dall'utente.

3) Definita in  $\mathbb{R}^2$  distanza euclidea tra 2 numeri  $A(x_A, y_A)$  e  $B(x_B, y_B)$  come

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}.$$

preparare uno script che

- legge in input da tastiera le coordinate di 2 punti nel piano cartesiano  $(x_A, y_A)$  e  $(x_B, y_B)$ ,
- chiama una funzione (opportunamente scritta) che calcoli la loro distanza Euclidea
- stampi il risultato

## ##### PYTHON 2

1) Creare due tuple che rappresentino i tre elenchi indicati sotto:

modello auto: Panda, Yaris, Captur, Corsa  
costruttore: Fiat, Toyota, Renault, Opel  
paese: Italia, Giappone, Francia, Germania

Ottenere una lista in cui ogni elemento è un dizionario {'modello': modello, 'costruttore': costruttore, 'paese':paese}, che accoppia modello, costruttore e paese in base all'ordine. Stamparla.

2) Scrivere un programma che chieda all'utente "dammi una stringa", legga la stringa fornita, ne stampi la lunghezza, ripeta l'operazione finchè l'utente non inserisce la stringa 'exit'

## ##### PYTHON 3

1) Scrivere uno script che, dati due elenchi di numeri, crei la matrice dei loro prodotti come nell'esempio:

v1: 1, 2, 3, 4, 5

v2: 8, 7, 8, 9, 10

mat: 1\*8 1\*7 1\*8 ... 2\*6 2\*7 2\*8 ... ..

Completare il programma con una stampa della matrice riga per riga:

[8, 7, 8 ...]

[16, 14, 16 ...]

2) Scrivere uno script che legge da tastiera un numero intero positivo, verifica che sia effettivamente intero e positivo altrimenti stampa un errore, controlla se è un numero primo e lo scrive sull'output

#### ##### PYTHON 4

1) Creare 2 matrici quadrate di dimensione 5000, scriverle in un file in 2 modi diversi:  
a\_ con dei cicli for

b\_ usando le funzionalità della libreria numpy

Usando opportunamente la funzione time del modulo time, calcolare il tempo impiegato dalle 2 operazioni separatamente e fare un confronto. Suggerimento: attenzione alla funzione del modulo time da usare che può essere diversa a seconda della versione di python in uso.

2) Plottare un semplice grafico della funzione "seno", includendo griglia, marcatore di tipo "diamond" con dimensione 5, con titolo e legenda del grafico.

3) Definire una funzione capace di plottare una gaussiana, che prende come parametri, oltre all'array del dominio, anche il valore medio e la deviazione standard

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{con } x \in \mathbb{R}.$$