

link testo

- Esercizio 0
- Esercizio 1
- Esercizio 2
- Esercizio 3

Esercizio 0

dalla tua directory e home personale `chmod 700 ../../../../public/alice.benatti4/`

Esercizio 1

Scrivere il programma `abyssmlink` che ha come parametro il pathname di una directory. Il programma deve cercare tutti i link simbolici presenti nella directory e trasformare ogni link simbolico in uno equivalente al pathname assoluto e non relativo.

```
#include <sys/stat.h>

// abyssmlink
int main (int argc, char *argv[]) {
    if(argc != 2){ //controllo che il numero di parametri sia corretto
        printf("ERR! Usage: %s PATH_TO_FOLDER\n", argv[0]);
        exit(EXIT_FAILURE);
    }
    struct stat s;
}
```

Esercizio 2

Esercizio 3

Scrivere uno script bash o un programma python che preso come parametro `pattern` [stringa ASCII] fornisca in output l'elenco dei file che contengono il `pattern` presenti nel sottoalbero della directory corrente. La lista di output deve essere ordinata dal file più recentemente modificato al file con ultima modifica più remota.

```
#!/bin/bash

# Get the pattern from command line argument
pattern=$1
```

```
# Find all files containing the pattern in the current directory and its subdirectories
# and sort the results by modification time in descending order
find . -type f -exec grep -H "$pattern" {} \; | sort -k3,3 -r

#!/usr/bin/env python3
import os
import sys

pattern = sys.argv[1] # get pattern from command line

findfiles = {}
for root, dirs, files in os.walk('.'):
    for file in files:
        path = os.path.join(root, file)
        with open(path, 'rb') as f:
            try:
                if pattern in f.read():
                    findfiles[os.path.getmtime(path)] = path
                    # findfiles.append((os.stat(path).st_mtime, path))
            except:
                pass

findfiles.sort(reverse=True)
for file in findfiles:
    print(file[1])
```