

Bài tập tại lớp: BTTL07

Các toán tử trên biến, viết hàm

Tạo một thư mục mang tên **BTTL07_ MSSV** chứa tất cả các file .sh cho từng bài tập bên dưới. Quy tắc đặt tên cho mỗi file .sh tương ứng từng bài tập:
BTTL0701_ MSSV.sh BTTL0706_ MSSV.sh.

Nội dung submit: là một file nén **BTTL07_ MSSV.zip** chứa thư mục vừa tạo ở trên.

Bài tập 1:

Viết script bash lấy đầu vào là một dãy số nguyên, cách nhau bằng một khoảng trắng.

Hãy xuất dãy số ra màn hình, mỗi số nằm trên một dòng.

Bài tập 2:

Viết script bash lấy đầu vào là một chuỗi các chuỗi ký tự, mỗi chuỗi ký tự đặt trong dấu nháy kép, cách nhau bởi một khoảng trắng.

Hãy hiển thị các chuỗi ký tự ra màn hình, mỗi chuỗi trên một dòng.

Bài tập 3:

Viết script bash lấy đầu vào là một dãy số nguyên, cách nhau bằng một khoảng trắng: Số đầu tiên trong dãy số là factor, các số còn lại là dãy số cần xử lý.

Kiểm tra số lượng tham số đầu vào: nếu số tham số nhỏ hơn hoặc bằng 1, hiển thị ra màn hình hướng dẫn sử dụng và thoát. Ngược lại, lưu phần tử đầu tiên vào biến factor, duyệt tất cả các phần tử còn lại trong danh sách và xuất ra màn hình dãy số trong đó từng phần tử của dãy số được nhân với factor.

```
./test 3 1 2 3 4 5 6
```

```
-> 3 6 9 12 15 18
```

Bài tập 4:

Viết script bash lấy 3 tham số đầu vào: hai toán hạng và một toán tử (+, -, *, /).

Script kiểm tra số lượng tham số, nếu số lượng tham số không thoả mãn, hiển thị hướng dẫn sử dụng và thoát.

Script sẽ xuất ra màn hình kết quả tương ứng với giá trị của toán tử do người dùng đưa vào. Riêng đối với phép chia, script sẽ kiểm tra chia 0.

./test 2 3 +

--> 2 + 3 = 5

Bài tập 5:

Viết script nhận tham số đầu vào một tên file đầu vào và một tên file đầu ra.

Script sẽ kiểm tra số lượng tham số, nếu không hợp lệ sẽ xuất ra màn hình hướng dẫn sử dụng và thoát.

Tiếp theo, script kiểm tra file đầu vào hợp lệ không, nếu không xuất ra thông báo lỗi và thoát.

Script kiểm tra file đầu ra tồn tại chưa, nếu chưa thì tạo file mới.

Script kiểm tra tùy chọn đầu vào và xử lý tương ứng:

-n : in n dòng đầu tiên có trong file

-e : mã hoá mỗi dòng bằng cách bằng ROT13^α bằng cách sử dụng lệnh tr.

Script sẽ lưu nội dung file đầu vào sau khi xử lý vào file đầu ra.

Bài tập 6:

1. Viết hàm **isExist** kiểm tra xem một đường dẫn có tồn tại hay không
2. Viết hàm **isDir** kiểm tra xem đầu vào có phải một đường dẫn không
3. Viết hàm **isFile** kiểm tra xem đầu vào có phải là một file hay không
4. Viết hàm **getExtension** lấy phần mở rộng của một file.
5. Viết hàm **getFileName** để lấy tên file (không có phần mở rộng)
6. Viết hàm **getFileInfo** để hiển thị thông tin về file: tên file và phần mở rộng, loại file, thông tin về số từ, số ký tự, số dòng của nội dung chứa trong file
7. Viết script đọc tham số từ dòng lệnh là một đường dẫn, sau đó thực hiện các công việc sau:
 - a. Nhập vào một đường dẫn
 - b. Kiểm tra xem đường dẫn đó có tồn tại không. Nếu không hiển thị thông báo và thoát ra khỏi script.
 - c. Nếu đầu vào là file: hiển thị ra màn hình các thông tin về file (gọi hàm **getFileInfo** đã định nghĩa ở trên).

Nếu đầu vào là thư mục: hiển thị ra màn hình thông tin từng file có trong thư mục đó.

^α <https://en.wikipedia.org/wiki/ROT13>