Lab 4

1. **(1p) AWK:** Print only the first 4 fields from each even-numbered line from a file, considering that the fields are separated by whitespaces. If a line has fewer than 4 fields, print all of them.

```
awk '{print $1" "$2" "$3" "$4}' test_input_1
```

Comment: if a line does not have at least 4 fields, the $X variables will just be empty

2. **(1p) GREP:** Print all the lines that contain only non-alphanumeric characters from a file. (any character that isn't a letter or a digit).

```
grep -E "^[^a-zA-Z0-9]*$" test_input_1
```

Comment: find all lines that contain 0 or more characters that are not letters or numbers

or

```
grep -E -v "[a-zA-Z0-9]" test_input_1
```

Comment: ignore all lines that contain at least one alphanumeric character and print all the other ones

3. **(1p) SED:** Duplicate each occurrence of an integer number from a file. We will consider that an integer number is a sequence of neighboring base 10 digits.
   - Ex: line "This 1234 is a number" will become "This 12341234 is a number"
   - Ex: line "56.34" will become "5656.3434"

```
sed -E "s/([0-9]+)/\1\1/g" test_input_1
```

4. **(2p) SED:** Delete all characters after the last whitespace from each line from a file.
   - Ex: line *"A regular, boring line"* will become "*A regular, boring "*
   - Ex: line *"A less regular line"* will become *"A less regular "*

```
sed -E "s/(\s)[^[:space:]]*$/\1/" test_input_2
```

Comment: if a "match any character except" is needed for some characters represented by a backslash expression (like \s in this case) [^\s] will not yield the desired result. [^\s] will mean "any character except \ and s". So we need to use a character class -> [:space:].

See **man grep**, section "REGULAR EXPRESSION", subsection "Character Classes and Bracket Expressions".

5. (2p) **AWK:** Print the line number and the field from the middle of the line from each line that contains an odd number of fields from a file. Consider that the fields are separated by whitespaces. Note: division in awk is by default float division. If you need the integer part of a division use the int function. Ex: int(5/2) = 2.

```
awk 'NF % 2 == 1{i=int((NF+1)/2); print NR" "$i}' test_input_1
```

6. (2p) **SED:** Swap field number 2 with field number 3 from a file where the fields are separated by the "**:**" character (Ex. /etc/passwd if available, but any file where fields are separated by : should do)

```
sed -E "s/^([^:]*):([^:]*):([^:]*)/\1:\3:\2/" /etc/passwd
```

7. (2p) **GREP:** Print all lines that contain at most 5 vowels, not necessarily consecutive, situated between 2 **^** signs from a file.
   - Ex: line *"aei^, still works^"* satisfies the condition
   - Ex: line *"abc^, way too many vowels here ^"* has too many vowels between the two **^**
   - Ex: line *"^here there are too many vowels^but not here^"* satisfies the condition because there are 4 vowels between the second and third occurrences of the **^** character

```
grep -E "\^([^aeiouAEIOU]*[aeiouAEIOU][^aeiouAEIOU]*){0,5}\^" test_input_4
```

Comment: `[^aeiouAEIOU]*[aeiouAEIOU][^aeiouAEIOU]*` would be interpreted as "a vowel surrounded by any amount of non-vowels on either side". The whole thing repeated up to 5 times ({0, 5}) will match at most 5 vowels, regardless if they are consecutive or not.

8. (3p) **SED:** Remove the first word containing only lowercase letters from each line of a file

```
sed -E "s/\<[a-z]+\>//" test_input_1
sed -E "s/\b[a-z]+\b//" test_input_1
sed -E "s/[[:<:]][a-z]+[[:>:]]\b//" test_input_1 - !! this maybe works on macOS, it will not work on the exam server !!
```

9. **(3p) AWK + GREP:** Print the number of processes run by each user in the system (in the format *nr_processes user*). You can obtain a list of all processes in the system and the user that is running each process using **ps -ef**. Check the manual for **sort** and **uniq**.

```
ps -ef | grep -E -v "^UID" | awk '{print $1}' | sort | uniq -c
```

10. **(3p) GREP + SED + AWK:** For each file from the current directory, display only the name of the file and the permissions for the user. (not the permissions for the *group* or for *other*; you can use **ls -l** to get information about files and folders from the current directory)

```
ls -l | grep -E "^-" | awk '{print $1" "$NF}' | sed -E "s/^.(...).{7}/\1/"
```