**Study year: 2020/2021**

# Midterm Exam 1

## Computer Practicum 1

**Instructions:**

- The exam consists of three tasks worth a total of 100 points.
- You have two hours  to solve the tasks.
- Use of the literature, internet or other media is NOT permitted.
- Save the solution to each task in its own file.
- Submit the solutions (3 bash script files) via e-učilnica when the instructor tells you to.

# Task 1 ( 30 points )

Write a script that performs following operations, on the first two arguments given.
Script should ask the operation from the user, he/she would like to perform.

- Summation
- Subtraction
- Product

You should check whether the both arguments are integers before entering the routine

- print the error message if they are not and exit the script with status 1.

**Example 1**

```
user@bash: ./task1.sh 5 3

What operation would you like to perform: *

Prod of 5 and 3 is  : 15.

----------------------------------------
```

For the rest of the operations, the output should like below

```
Sum of 5 and 3 is   : 8.
```

```
Sub of 5 and 3 is   : 2.
```

**Example 2**

```
user@bash: ./task1.sh a 8

Error: The provided arguments are not
integers. Please provide two integers to
multiply them up.
```

## Task 2 ( 40 points )

Write a bash script which accepts 2 command line arguments, to perform following operations. The first argument is a name for a file and the second argument is a name for a directory.

IMPORTANT (following operations should perform by the script itself)
First check if the folder already exists in your current working directory.
If it exists delete the folder and all the sub folders and files in it.
If not create the folder in your current working directory.

Get a snapshot of the processes running on your computer for all users, and save the output to a file called processes.log , inside your newly created folder.

Create a new file, inside the directory and name the file as the first argument. Append following details into the file
username
Your current working directory
Information of your operating system
Files and the folders with size and permission list inside your newly created folder

**Example 1**

```
user@bash: ./task2.sh info.txt details
```

**details/Processes.log**

```
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
_windowserver      223   8.4  0.4  8600372  67464   ??  Ss    8:40PM  41:18.39
/System/Library/P
rt               72801   1.9  3.4  9683548 570644   ??  S     8:52AM  13:41.60
/Applications/Goo
rt                4813   1.8  3.5 13408500 592940   ??  S    10:11AM   0:55.89
/Applications/Goo
rt               94382   1.4  0.5  5038408  75572   ??  S     9:50AM   0:06.43
/System/Applicati
_hidd              132   1.3  0.0  4381732   8016   ??  Ss    8:40PM   7:14.54
```

**details/info.txt**

```
89193007(your username goes here)
/home/studenti/famnit/89193007/Documents
Linux student 4.4.0-161-generic #189-Ubuntu SMP Tue Aug 27 08:10:53 UTC
2019 i686 i686 i686 GNU/Linux
drwxrwxr-x  2 89193004 97111 4096 Nov  8 11:04 .
drwxr-xr-x 32 89193004 97111 4096 Nov  8 11:04 ..
-rw-rw-r--  1 89193004 97111    0 Nov  8 11:04 info.txt
-rw-rw-r--  1 89193004 97111    0 Nov  8 11:04 processes.log
```

# Task 3 ( 30 points )

Write a script that asks the user for a number N (using read command) and prints all the even numbers from 1 to N. Count how many even numbers are there and calculate the sum of the even numbers

Also check whether the user has provided an integer. If the input isn't an integer, print out the error message.

**Example 1:**

```
user@bash: ./task3.sh
Please provide a number (integer): 6
Even numbers are from 1-6:
2
4
6
Number of total even numbers: 3
Sum of all the even numbers from 1-6 : 12
```

**Example 2:**

```
user@bash: ./task3.sh
Please provide a number (integer): N
Error: N is not an integer. Please provide an integer.
```

HINT:

You can use **seq** command as the iterator of the loop

- **seq FIRST LAST** : When two arguments are given then it produces numbers from *FIRST* till *LAST* is step increment of 1. If *LAST* is less than *FIRST*, then it produces no output.