



**MARMARA UNIVERSITY**

**FACULTY OF ENGINEERING  
COMPUTER SCIENCE & ENGINEERING  
DEPARTMENT**

**CSE3033  
OPERATING SYSTEMS  
Report of Programming Assignment #1**

*A.Tunahan Cinsoy – 150117062*

*Enver Aslan – 150115851*

*Cem Güleç – 150117828*

## Explanations About Implementations of Question 1

In question 1, we simply created a loop that iterates given numbers in text file. While iteration process continues, our script counts the occurrences of each number, and prints occurrences of each number with stars.

Here is the screenshot of our sample executions:

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog1.sh
The file name is required as an argument.
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog1.sh input1.txt
0
1 **
2 ***
3
4 *
5 **
6 ***
7 *
8 *
9
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog1.sh input1.txt
The input file includes invalid values.
```

- For first execution, we wanted to show that program can't be run without argument.
- Second execution is the correct execution case.
- Third output occurs only if input file contains a value that is not between [0-9]

## Explanations About Implementations of Question 2

In question 2, we take two input arguments. Based on second numerical argument, we convert first string argument into a ciphered version. For each letter in the string, our script finds another letter in English alphabet advancing over the alphabet corresponding digit times. If there is only one digit as second input argument, our script applies that corresponding advance to all letters in first argument. For the conversion purpose, we've used ASCII values of letters.

Here is the screenshot of our sample executions:

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh apple 12345
Encrypted string: brspj
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh zoo 8
Encrypted string: hww
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh tuna 9851
Encrypted string: ccsb
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh enver 9
Encrypted string: nwna
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh cem 654
Encrypted string: ijq
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh Zuha1 85214
Encrypted string: Hzjbp
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh ALI 8
Encrypted string: ITQ
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog2.sh ALI 85
Input format is incorrect
```

- Our script contains ciphering uppercase characters as well as lowercase characters.

## Explanations About Implementations of Question 3

In question 3, our aim is to delete the oldest file under working directory unless there is an optional pathname as an argument. If there is a pathname, our script simply moves to that location and applies the same rule under that working directory.

Here are some screenshots of our sample executions:

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls -lt
total 132
drwxrwxr-x 2 enver enver 4096 Kas 21 18:49 shakespeare
-rw-rw-r-- 1 enver enver 414 Kas 21 18:48 input4.txt
-rw-rw-r-- 1 enver enver 26 Kas 21 18:45 input1.txt
-rwxrwxr-x 1 enver enver 685 Kas 21 18:37 myprog3.sh
-rwxrwxr-x 1 enver enver 1694 Kas 21 17:42 Menu.sh
-rwxrwxr-x 1 enver enver 1453 Kas 21 17:29 myprog5.sh
-rw-rw-r-- 1 enver enver 16 Kas 21 16:44 README.md
-rwxrwxr-x 1 enver enver 1441 Kas 21 02:26 myprog1.sh
-rwxrwxr-x 1 enver enver 3085 Kas 21 02:26 myprog2.sh
-rwxrwxr-x 1 enver enver 633 Kas 21 02:26 myprog4.sh
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 trees-and-other-poems.txt
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 cask-of-amontillado.txt
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 french.txt
-rw-rw-r-- 1 enver enver 90806 Kas 19 21:13 Project1.pdf
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog3.sh
rm: remove regular file 'Project1.pdf'? y

enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls -lt
total 40
drwxrwxr-x 2 enver enver 4096 Kas 21 18:49 shakespeare
-rw-rw-r-- 1 enver enver 414 Kas 21 18:48 input4.txt
-rw-rw-r-- 1 enver enver 26 Kas 21 18:45 input1.txt
-rwxrwxr-x 1 enver enver 685 Kas 21 18:37 myprog3.sh
-rwxrwxr-x 1 enver enver 1694 Kas 21 17:42 Menu.sh
-rwxrwxr-x 1 enver enver 1453 Kas 21 17:29 myprog5.sh
-rw-rw-r-- 1 enver enver 16 Kas 21 16:44 README.md
-rwxrwxr-x 1 enver enver 1441 Kas 21 02:26 myprog1.sh
-rwxrwxr-x 1 enver enver 3085 Kas 21 02:26 myprog2.sh
-rwxrwxr-x 1 enver enver 633 Kas 21 02:26 myprog4.sh
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 trees-and-other-poems.txt
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 cask-of-amontillado.txt
-rw-rw-r-- 1 enver enver 0 Kas 19 21:13 french.txt
```

· As seen screenshot above, “ls -lt” prompt shows us that “Project1.pdf” is the oldest file under current working directory. So, the script asks us to delete that file and when we type “y”, it deletes successfully.

```

enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ ls shakespeare -lt
total 0
-rw-rw-r-- 1 enver enver 0 Kas 21 17:31 barleby-scrivener.txt
-rw-rw-r-- 1 enver enver 0 Kas 21 17:03 calaveras-county.txt
enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ ./myprog3.sh shakespeare
rm: remove regular empty file './shakespeare/calaveras-county.txt'? y

enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ ls shakespeare -lt
total 0
-rw-rw-r-- 1 enver enver 0 Kas 21 17:31 barleby-scrivener.txt

```

- When we give an optional pathname, script goes to that specific folder and applies the same procedure under that working directory.

### Explanations About Implementations of Question 4

In question 4, our script iterates the content of the given input argument text file with the aim of finding numbers between 0 to 9 both inclusive. When it faces with numbers, it converts these numbers to text.

Here is the screenshot of our sample execution:

```

enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ cat input4.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. 8 Suspendisse vitae odio blandit, commodo nisl dignissim, 5 commodo est. Quisque blandit laoreet ante id tincidunt. Vivamus in vestibulum sem. Duis ac faucibus quam. Mauris posuere, sapien quis elementum porttitor, leo turpis finibus erat, vel dapibus 12 lorem mauris in elit. Curabitur quis mas4sa sit amet ligula suscipit pulvinar.
enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ ./myprog4.sh input4.txt
enver@enverUbuntu:~/Documents/Lectures/Opst/HW1/Shell-Problems$ cat input4.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. eight Suspendisse vitae odio blandit, commodo nithreesl dignissim, five commodo est. Quisque blandit laoreet ante id tincidunt. Vivamus in vestibulum sem. Duis ac faucibus quam. Mauris posuere, sapien quis elementum porttitor, leo turpis finibus erat, vel dapibus onetwo lorem mauris in elit. Curabitur quis masfoursa sit amet ligula suscipit pulvinar.

```

- When we randomly put numbers in a text file, script detects and converts these numbers to text.

### Explanations About Implementations of Question 5

In question 5, we wrote a script that takes a wildcard argument and a -R option which is for recursive iteration. Based on wildcard argument, script creates a folder named “copied” and puts files whose name obey the wildcard. If there is an -R option, script works recursively which means that it goes to every folder that is located under working directory and applies the same procedure to these folders as well.

Here are some screenshots of our sample executions:

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls -l
total 132
-rw-rw-r-- 1 enver enver 0 Kas 21 23:01 cask-of-amontillado.txt
-rw-rw-r-- 1 enver enver 0 Kas 21 20:15 french.txt
-rw-rw-r-- 1 enver enver 26 Kas 21 18:45 input1.txt
-rw-rw-r-- 1 enver enver 403 Kas 22 00:06 input4.txt
-rwxrwxr-x 1 enver enver 1441 Kas 21 02:26 myprog1.sh
-rwxrwxr-x 1 enver enver 3085 Kas 21 02:26 myprog2.sh
-rwxrwxr-x 1 enver enver 2189 Kas 22 00:01 myprog3.sh
-rwxrwxr-x 1 enver enver 633 Kas 21 02:26 myprog4.sh
-rwxrwxr-x 1 enver enver 3206 Kas 22 00:19 myprog5.sh
-rwxrwxr-x 1 enver enver 2076 Kas 22 00:01 myprog.sh
-rw-rw-r-- 1 enver enver 90806 Kas 21 18:58 Project1.pdf
-rw-rw-r-- 1 enver enver 16 Kas 21 16:44 README.md
drwxrwxr-x 2 enver enver 4096 Kas 22 00:31 shakespeare
-rw-rw-r-- 1 enver enver 0 Kas 21 23:25 trees-and-other-poems.txt
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog5.sh "c*.txt"
cask-of-amontillado.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls -l
total 136
-rw-rw-r-- 1 enver enver 0 Kas 21 23:01 cask-of-amontillado.txt
drwxrwxr-x 2 enver enver 4096 Kas 22 00:32 copied
-rw-rw-r-- 1 enver enver 0 Kas 21 20:15 french.txt
-rw-rw-r-- 1 enver enver 26 Kas 21 18:45 input1.txt
-rw-rw-r-- 1 enver enver 403 Kas 22 00:06 input4.txt
-rwxrwxr-x 1 enver enver 1441 Kas 21 02:26 myprog1.sh
-rwxrwxr-x 1 enver enver 3085 Kas 21 02:26 myprog2.sh
-rwxrwxr-x 1 enver enver 2189 Kas 22 00:01 myprog3.sh
-rwxrwxr-x 1 enver enver 633 Kas 21 02:26 myprog4.sh
-rwxrwxr-x 1 enver enver 3206 Kas 22 00:19 myprog5.sh
-rwxrwxr-x 1 enver enver 2076 Kas 22 00:01 myprog.sh
-rw-rw-r-- 1 enver enver 90806 Kas 21 18:58 Project1.pdf
-rw-rw-r-- 1 enver enver 16 Kas 21 16:44 README.md
drwxrwxr-x 2 enver enver 4096 Kas 22 00:31 shakespeare
-rw-rw-r-- 1 enver enver 0 Kas 21 23:25 trees-and-other-poems.txt
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$
```

· When we execute myprog5.sh without -R flag, it creates a copied folder under working directory and copies files which obey given wildcard argument.

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls shakespeare -l
total 0
-rw-rw-r-- 1 enver enver 0 Kas 21 23:26 barleby-scrivener.txt
-rw-rw-r-- 1 enver enver 0 Kas 22 00:07 calaveras-county.txt
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog5.sh -R "c*.txt"
cask-of-amontillado.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied
calaveras-county.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/shakespeare/copied
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls shakespeare -l
total 4
-rw-rw-r-- 1 enver enver 0 Kas 21 23:26 barleby-scrivener.txt
-rw-rw-r-- 1 enver enver 0 Kas 22 00:07 calaveras-county.txt
drwxrwxr-x 2 enver enver 4096 Kas 22 00:35 copied
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls shakespeare/copied -l
total 0
-rw-rw-r-- 1 enver enver 0 Kas 22 00:35 calaveras-county.txt
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ls copied -l
total 0
-rw-rw-r-- 1 enver enver 0 Kas 22 00:35 cask-of-amontillado.txt
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$
```

· When we execute myprog5.sh with -R flag, program understands that it will work recursively, then it creates a copied folder under working directories and copies files which obey given wildcard argument.

```
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$ ./myprog5.sh -R "*.txt"
input1.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied
input4.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied
There is no file that obeys the wildcard under /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/shakespeare
enver@enverUbuntu:~/Documents/Lectures/Opsis/HW1/Shell-Problems$
```

- If there is no file that obeys the wildcard, we simply give an error prompt to user

## Explanations About Implementations of Menu

For the bonus part of the programming assignment #1, we created a menu that includes all questions above.

Here are some screenshots of our sample executions:

```
-----
                          Main Menu
-----
1. Create histogram
2. Encryption
3. Delete oldest
4. Convert numbers
5. Organized files
6. Exit\n
Enter your menu choice [1-6]:
```

```
-----
                          1. Create Histogram
-----
Please enter the file name that includes number data: input1.txt

0
1 **
2 ***
3
4 *
5 **
6 ***
7 *
8 *
9

Press any key to back to main menu...
```

```
-----  
1. Create Histogram  
-----  
Please enter the file name that includes number data: test  
  
File does not exist  
  
Press any key to back to main menu... █
```

```
-----  
2. Encryption  
-----  
Please enter string and encryption number: apple 12345  
Encrypted string: brspj  
  
Press any key to back to main menu... █
```

```
-----  
2. Encryption  
-----  
Please enter string and encryption number: zoo 8  
Encrypted string: hww  
  
Press any key to back to main menu... █
```

```
-----  
2. Encryption  
-----  
Please enter string and encryption number: asdf 12  
Input format is incorrect  
  
Press any key to back to main menu... █
```

```
-----  
2. Encryption  
-----  
Please enter string and encryption number: asdf  
The number of parameters should be two.  
  
Press any key to back to main menu... █
```



```
-----  
3. Delete oldest  
-----  
Please enter optional path name (if it exists):  
rm: remove regular empty file 'trees-and-other-poems.txt'? y  
1 file is deleted  
Press any key to back to main menu... █
```

```
-----  
3. Delete oldest  
-----  
Please enter optional path name (if it exists):  
rm: remove regular file 'myprog4.sh'? n  
Deletion process is cancelled  
Press any key to back to main menu... █
```

```
-----  
3. Delete oldest  
-----  
Please enter optional path name (if it exists): shakespeare  
rm: remove regular empty file 'barleby-scrivener.txt'? y  
1 file is deleted  
Press any key to back to main menu... █
```

```
-----  
3. Delete oldest  
-----  
Please enter optional path name (if it exists): shakespeare  
rm: remove regular empty file 'calaveras-county.txt'? n  
Deletion process is cancelled  
Press any key to back to main menu... █
```

```
-----  
4. Convert numbers  
-----  
Please enter the file name to convert number to string: input4.txt  
Press any key to back to main menu... █
```

```
-----  
4. Convert numbers  
-----  
Please enter the file name to convert number to string:  
Please enter the file name as an argument  
Press any key to back to main menu... █
```



```
-----  
5. Copy files  
-----  
Please enter optional recursive flag -R and wildcard argument respectively: -R "fre???.txt"  
french.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied  
There is no file that obeys the wildcard under /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/shakespeare  
Press any key to go back to main menu... █
```

```
-----  
5. Copy files  
-----  
Please enter optional recursive flag -R and wildcard argument respectively: -R "c*.txt"  
cask-of-amontillado.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied  
calaveras-county.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/shakespeare/copied  
Press any key to go back to main menu... █
```

```
-----  
5. Copy files  
-----  
Please enter optional recursive flag -R and wildcard argument respectively: "i*.txt"  
input1.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied  
input4.txt is copied under to /home/enver/Documents/Lectures/Opsis/HW1/Shell-Problems/copied  
Press any key to go back to main menu... █
```

· As seen above, we've handled most of the corner cases such as file not found, deletion process and different wildcard cases.

· For every question, we've tried a lot of case to find out some minor and major bugs, the ones that we've caught are all fixed.

## Used Technologies

- GitHub
- VS Code
- UNIX