**Log files**

In config.py selection of log type can be made.it can be csv or log type.

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

****

In previous execution log file wasn't implements and all information was printed to output

**Sort Commands**

Run sort commands "python sort.py c:\Main\mixed"

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Sort help command

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

In attached Input Data folder there are folders with before and after data to run sort command.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Sort with no files to sort



Sort command with an empty folder



תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Sort command with optional hash value

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**Clean command**

Clean command provides an ability to maintain a folder and delete older files if needed.

In configuration file "Commands\config.py" the user can update the number of files he wants to keep and number of files he wants to delete.

For example:

the number of files he wants to keep,

'window\_size': 4,

number of files he wants to delete

'deletion\_threshold': 2,

COMMANDS\_CONFIG = {

    'window\_size': 4,

    'log\_type': 'log',

    'deletion\_threshold': 2,

    "log\_extension": ".txt"

}

Clean – delete 2 files .keep 4 , delete 2 , total files 6תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Clean – delete 6 files, keep 1 ,total files 7

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Clean – delete 2 files, keep 3, total files 7

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Clean – no log files to delete

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Clean – folder is empty, no log files to delete



**Click**

**How to install and run Click module**

**python -m venv click**

**.\click\Scripts\Activate.ps1**

**pip install click**

**".. \click\Scripts\python.exe"**

**Python click**

Python click module is used to create command-line (CLI) applications.

It is an easy-to-use alternative to the standard optparse and argparse modules. It allows arbitrary nesting of commands, automatic help page generation, and supports lazy loading of subcommands at runtime.

The click module was created as a supporting library for the Flask web framework.

The commands are basic building blocks of an application. Click defines commands through decorators. They are created with the click.command() decorator. Values are passed to the commands via options or arguments. Options are added with the click.option() decorator, arguments with the click.argument(). Values in options follow the name of the option while arguments are taken positionally.