

CS 104



LAB WEEK 08

1. Generate 26 text files named A.txt, B.txt, and so on up to Z.txt under a directory named 'letters'. You can use "makedirs" method from os library to generate a directory.
2. Write a Python program to create a file named "words.txt" where all letters of English alphabet are listed by specified number of letters on each line.

Example output in the file is as follows:

```
words.txt
ABC
DEF
GHI
JKL
MNO
PQR
STU
VWX
YZ
```

3. A text file named "matter.txt" contains some text, which needs to be displayed such that every next character is separated by a symbol "#". Write a function definition for hash_display() in Python that would display the entire content of the file matter.txt in the desired format.

Example :

If the file matter.txt has the following content stored in it :
THE WORLD IS ROUND

The function hash_display() should display the following content :
T#H#E# #W#O#R#L#D# #I#S# #R#O#U#N#D#

4. Create a note-taking program. When a user starts it up, it should prompt them for a filename. If they enter a file name that doesn't exist, it should prompt them to enter the text they want to write to the file. After they enter the text, it should save the file and exit.

If they enter a file name that already exists, it should ask the user if they want:

- A) Read the file
- B) Delete the file and start over
- C) Append the file

If the user wants to read the file it should simply show the contents of the file on the screen.

If the user wants to start over then the file should be deleted and another empty one made in its place.

If a user elects to append the file, then they should be able to enter more text, and that text should be added to the existing text in the file.

Extra Part:

Allow the user to select a 4th option:

D) Replace a single line

If the user wants to replace a single line in the file, they will then need to be prompted for 2 bits of information:

- 1) The line number they want to update.
- 2) The text that should replace that line.

5. Write a program which reads a 5x5 matrix from the user. This matrix will represent a grid world, and each element represents a cell. A cell is empty if it is 0, and blocked if it is 1.

Assume a player will start from the top-left, (0,0), and will try to reach the bottom-right (4,4) of the grid. Player is allowed to go up, down, left, right, and one cell each time. The top-left and bottom-right corners will always be given empty.

You should write a method which checks if there is such a possible path. If such a path exists, it should return true and it should return false otherwise. If the method returns true, print the latest status of the board to an output file.

- First read the input from the user and print it to a file to check whether you can read a 2-D array correctly from the user.
- Change the cell (0,0) to a 2.
- Repeat the following process until no new cell is changed to 2:
 - If a cell contains a 2 and it has neighbors which are equal to 0, update neighbors as 2.
- If at the end the cell(4,4) gets marked, then it means that there is a path. Print the board to a file and return true.

```
0 1 1 0 1
0 1 0 0 1
0 0 0 1 1
1 1 0 0 0
1 1 1 1 0
```

Path found

Output file:

```
21121
21221
22211
11222
11112
```