1. W.A.P to print the contents of a supplied textfile.

I/P: filename

O/P: File doesn't exist

File Data should be displayed.

Objective: fopen, fgetc, fclose

```
FILE *fopen(const char *, const char *);
```

- FILE: typedefined user-defined struct
- const char *: filename
  - if path not supplied then current folder is used
- const char *: mode
  - r, w, a
  - rt, wt, at
  - rb, wb, ab

Step1: try to open the file. if non existing then exit the app.

Step2: Read byte by byte from the file stream and print.

Step3: close the file stream

```
ch = getchar();
```

//getchar() function collects/reads one char from stdin-filestream and returns the character. ch variable collects that.

// if stdin-filestream is Empty, then getchar waits.

```
ch = fgetc(stdin);
```

//SAME explanation as above.

fgetc can work as getchar, but getchar can't do fgetc work.

2. W.A.P to Implement cp command.

Your app a.out/fileCpy should behave like the cp command.

$ ./a.out src_file dest_file

(src_file: r, dest_file: w)

content of src_file should be copied into dest_file

step1: open the src file, open the dest file

step2: Read char by char from src-file-stream and write those chars into dest-file-stream

step3: close src & dest streams

putchar(97); //this stdout function sends 97/'a' to stdout buffer.

fputc(97, stdout); //Exact Equivalent of the above putchar

TASK MUST COMPLETE Before monday

W.A.P to search a string in a file and perform the following

(i) Delete all occurences

or (ii) Hide " "

or (iii) replace with reverse equivalent.

$ ./a.out vector datafile.txt

step1 : open the file fopen(argv[2], "r");

step2 : Bring the content into dyn_mem_block.

loop { buf = realloc(-, -);
buf[i] = ch;
}

step3 : close file

step4 : String manipulation

step5 : open the same file in w mode
and write the buf into file.