CSE 101 GROUP – 09

Assignment on FILE operations MODE – r+b

Group members: 1805069

1805081

1805093

1805105

1805117

1. If we open a random .dat file in r+b mode it does not open and shows an error that the file does not exist.

```
FILE *fp;
if((fp = fopen("sample.dat", "r+b")) == NULL){
        printf("ERROR- file does not exist");
        exit(1);
}
Else{
        fread(str, sizeof(str), 1, fp);
        printf("%s", str);
}
```

   OUTPUT:
   ERROR- file does not exist

2. If we don't close the file (fclose), the program crashes (not always).

3. We created a file with a random extension (.ananto) using "wb" mode. And we can just read what's written in it using rb or rb+. So, we think when we use an arbitrary extension, the file gets saved in binary form in it (not .txt).

4. We can use two different pointers for the same file and it does not create an error.

5. We can use two different file operation modes in the same code for the same file.

6. So we created a file and then ran the following operations
   - First we printed whatever was in the file using fread.

```
char str[1000];
FILE *fp;

fread(str, sizeof(str), 1, fp);
```

- We can also write something in the file at first without reading it. But then fread does not always work after that

- We use fseek and read the file from a random position and print the characters upto eof
  We can use ftell in this matter and count how many characters to be printed.
  For example,
  fseek(fp, 0, SEEK_END);
  long l = ftell(fp);
  fseek(fp, 19, SEEK_SET);
  l = l-19;
  fread(str, sizeof(char), l, fp);

  If we try to print the whole string altogether it creates some problems sometimes and prints gibberish character
  printf("%s", str);
  But if we print character by character it does not create a problem

- We use puts to write at the end of the file in a different string and print it. It does not work and gives gibberish characters. But fputc works fine.

- But when we come to a random point inside the file and try to use fwrite to write something, it overwrites whatever is written in the file
  As for example,
  We at first took the whole data of the file in str1[1000]
  Then we used fseek and came to the 50th character of the file from the beginning of the file and took input from console in str2[50]. We wrote this str2 in the file and printed the whole data of the file again.
  This time it was noticed that 50 characters from the the 51st to 100th position of the file were replaced by the string we gave as input as well as the remaining of str2 was filled up with gibberish characters.

- If we declare a string of 100 characters and assign only 50 characters in it, the function strlen returns 50 and sizeof returns 100 as usual. Now, if we put that string in a file in binary mode (rb+) with this piece of code,

  fwrite(str, sizeof(char), strlen(str), fp);
  what we see is 50 characters are written as it should be but there's 50 more of just SPACES following that.

- If we fseek to the end and set the 2nd parameter (offset) to something negative (-30 for instance), nothing gets read.

- The rewind function works just fine for the file.