Saivya Singh
CSE D 44
220905370

Lab 1 : Basic File Handling Operations

Q1.To count the number of lines and characters in a file.

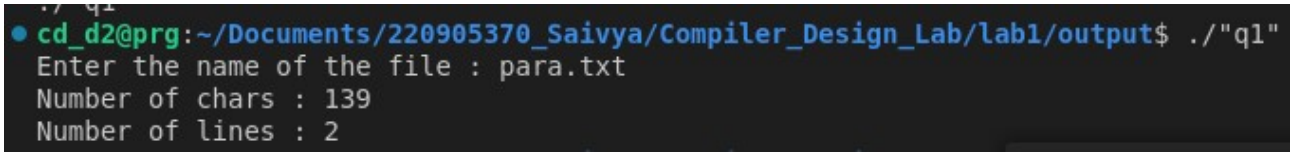
Code:

```
#include <stdio.h>
#include <stdlib.h>

int main(){
   FILE *fp1;
   char filename[100],c;

   printf("Enter the name of the file : ");
   scanf("%s",filename);

   fp1 = fopen(filename,"r");

   if(fp1==NULL){
      perror("Invalid File Name");
      exit(0);
   }

   int ctr=0;
   int sctr=1;
   c = fgetc(fp1);

   while(c != EOF){
      ctr++;
      c = fgetc(fp1);
      if(c == '\n'){
         sctr++;
      }
   }

   fclose(fp1);

   printf("Number of chars : %d \n",ctr);
   printf("Number of lines : %d \n",sctr);
}
```

Input:

John quickly ran to the store, bought some groceries, and returned home. He then started cooking dinner and called his friend to join him.

Output:



Q2.To reverse the file contents and store in another file. Also display the size of file using file handling function.

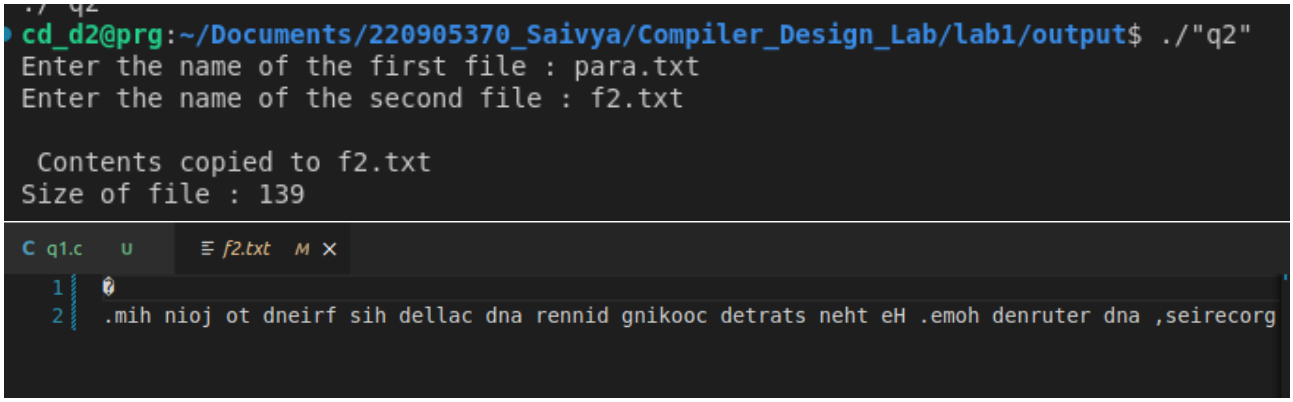
Code:

```
#include <stdio.h>
#include <stdlib.h>

int main(){
FILE *fp1,*fp2;
char filename1[100],filename2[100],c;

printf("Enter the name of the first file : ");
scanf("%s",filename1);

printf("Enter the name of the second file : ");
scanf("%s",filename2);

fp1 = fopen(filename1,"r");
fp2 = fopen(filename2,"w+");

if(!fp1){
perror("Invalid File Name");
exit(0);
}

fseek(fp1,0,SEEK_END);
long size = ftell(fp1);


for(long i = size;i>=0;i--){
fseek(fp1,i,SEEK_SET);
c = fgetc(fp1);
fputc(c,fp2);
}

printf("\n Contents copied to %s \n",filename2);
printf("Size of file : %ld \n \n",size);

fclose(fp1);
fclose(fp2);
}
```

Input:

John quickly ran to the store, bought some groceries, and returned home. He then started cooking dinner and called his friend to join him.

Output:



Q3.That merges lines alternatively from 2 files and stores it in a resultant file.

Code:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
FILE *fp1, *fp2, *fp3;
char filename1[100], filename2[100], filename3[100];
char c1, c2;

printf("Enter the name of file 1: ");
scanf("%s", filename1);

printf("Enter the name of file 2: ");
scanf("%s", filename2);

printf("Enter the name of file 3 (output file): ");
scanf("%s", filename3);

fp1 = fopen(filename1, "r");
fp2 = fopen(filename2, "r");
fp3 = fopen(filename3, "w+");

if (!fp1 || !fp2) {
perror("Invalid File Name");
exit(1);
}

while (1) {
while ((c1 = fgetc(fp1)) != EOF && c1 != '\n') {
fputc(c1, fp3);
}
if (c1 != EOF) {
fputc('\n', fp3);
}
```

```
while ((c2 = fgetc(fp2)) != EOF && c2 != '\n') {
fputc(c2, fp3);
}
if (c2 != EOF) {
fputc('\n', fp3);
}

if (c1 == EOF && c2 == EOF) {
break;
}
}

fclose(fp1);
fclose(fp2);
fclose(fp3);

return 0;
}
```

Input:

f1.txt
style
saivya

f2.txt
while
singh

Output:

```
cd_d2@prg:~/Documents/220905370_Saivya/Compiler_Design_Lab/lab1/output$ ./"q3"
Enter the name of file 1: f1.txt
Enter the name of file 2: f2.txt
Enter the name of file 3 (output file): f3.txt
```

```
1  style
2  while
3  saivya
4  singh
```

Q4.That accepts an input statement, identifies the verbs present in them and performs the following
functions
a. INSERT: Used to insert a verb into the hash table.
Syntax: insert (char *str)
b. SEARCH: Used to search for a key(verb) in the hash table. This function is called by INSERT
function. If the symbol table already contains an entry for the verb to be inserted, then it returns
the hash value of the respective verb. If a verb is not found, the function returns -1.
Syntax: int search (key)

Code:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_WORD_LENGTH 50
#define MAX_WORDS 10
#define MAX_PARAGRAPH_LENGTH 1000

int main() {
char file1[100], file2[100];

printf("Enter the name of the paragraph file (File 1): ");
scanf("%s", file1);
printf("Enter the name of the words file (File 2): ");
scanf("%s", file2);

FILE *f1 = fopen(file1, "r");
if (!f1) {
printf("Error opening file %s\n", file1);
return 1;
}

char paragraph[MAX_PARAGRAPH_LENGTH];
int i = 0;
char c;
while ((c = fgetc(f1)) != EOF && i < MAX_PARAGRAPH_LENGTH - 1) {
paragraph[i++] = c;
}
paragraph[i] = '\0';
fclose(f1);

FILE *f2 = fopen(file2, "r");
if (!f2) {
printf("Error opening file %s\n", file2);
return 1;
}

char word[MAX_WORD_LENGTH];
for (int w = 0; w < MAX_WORDS; w++) {
i = 0;
while ((c = fgetc(f2)) != EOF && !isspace(c)) {
word[i++] = c;
}
word[i] = '\0';

int found = 0;
char *ptr = paragraph;
int index = 0;
while ((ptr = strstr(ptr, word)) != NULL) {
```

```
index = ptr - paragraph + 1;
found = 1;
printf("Word '%s' found at position %d in the paragraph.\n", word, index);
ptr++;
}

if (!found) {
printf("Word '%s' not found in the paragraph.\n", word);
}

if (c == EOF) break;
while (isspace(c)) {
c = fgetc(f2);
}
}

fclose(f2);
return 0;
}
```

Input:

para.txt
John quickly ran to the store, bought some groceries, and returned home. He then started cooking dinner and called his friend to join him.

verbs.txt
ran
bought
returned
started
called

Output: