Name Priyadarshi Prabhakar SAP ID 590029237

Experiment 9: File Handling in C – Algorithms, Pseudocode & C Programs

1. Create a new file and write text into it

Algorithm:

1. Start
2. Declare a file pointer
3. Open a file in write mode
4. If the file cannot be opened, display an error
5. Write content into the file
6. Close the file
7. Stop

Pseudocode:

```
BEGIN
  DECLARE file pointer fp
  OPEN file in write mode
  IF file is not opened THEN
    DISPLAY error
  ELSE
    WRITE text to file
  ENDIF
  CLOSE file
END
```

C Program:

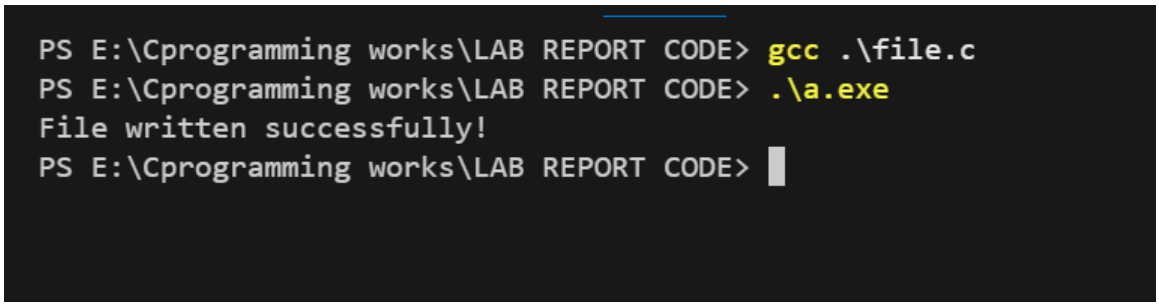
```
#include <stdio.h>


int main() {
   FILE *fp = fopen("output.txt", "w");
   if (fp == NULL) {
      printf("Error opening file!");
      return 1;
   }


   fprintf(fp, "This is a sample text written to the file.\n");
   fclose(fp);


   printf("File written successfully!\n");
   return 0;
}
```

OUTPUT



2. Read a file character by character

Algorithm:

1. Start
2. Declare a file pointer
3. Open file in read mode
4. If file isn't found, print error
5. Read each character until EOF
6. Display each character
7. Close file
8. Stop

Pseudocode:

```
BEGIN
  OPEN file in read mode
  IF file not found THEN
    DISPLAY error
  ELSE
    WHILE not EOF
      READ a character
      PRINT character
    END WHILE
  ENDIF
  CLOSE file
END
```

C Program:

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

```
int main() {
   FILE *fp = fopen("output.txt", "r");
   char ch;

   if (fp == NULL) {
      printf("File not found!");
      return 1;
   }

   while ((ch = fgetc(fp)) != EOF) {
      printf("%c", ch);
   }

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

OUTPUT

```
PS E:\Cprogramming works\LAB REPORT CODE> gcc .\character.c
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
This is a sample text written to the file.
PS E:\Cprogramming works\LAB REPORT CODE>
```

3. Read a file line by line

Algorithm:

1. Start

2. Declare a file pointer

3. Open file in read mode

4. If file is not available, display error

5. Read line by line using fgets()

6. Display each line

7. Close the file

8. Stop

Pseudocode:

```
BEGIN
   OPEN file in read mode
   IF file cannot open THEN
      PRINT error
   ELSE
      WHILE line exists
         READ line using fgets
         PRINT line
      END WHILE
   ENDIF
   CLOSE file
END
```

C Program:

```
#include <stdio.h>

int main() {
```

```
    FILE *fp = fopen("output.txt", "r");
    char line[200];

    if (fp == NULL) {
        printf("Unable to open file!");
        return 1;
    }

    while (fgets(line, sizeof(line), fp)) {
        printf("%s", line);
    }

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

OUTPUT