Department of Statistics & Computer Science, University of Kelaniya

ACADEMIC YEAR – 2022/2023

BECS 11223 – Fundamentals of Programming

Lab Session 8

Throughout this lab session, you will learn about file processing in C.

1. Develop a C program that read an integer value, float value and character value from the external file called *input.txt*. Then, display the values read from the external file.

   Here is the content of the *input.txt* file (Create your own *input.txt* file in your computer):

   ```
   100
   1.25
   A
   ```

Upload your completed C program into the Lab 8– Program 01 submission folder.

2. Develop a C program that reads five numbers from the file *infile.txt* and store in an array called *values*. Then, find and display how many odd numbers are in the *values* array.

   Here is the content of the *infile.txt* file (Create your own *infile.txt* file in your computer):

   ```
   2
   5
   8
   13
   14
   ```

   The output of the program for the above input file should be "*2 odd numbers*".

Upload your completed C program into the Lab 8– Program 02 submission folder.

3. Develop a C program that reads ten numbers from the file *numbers.txt* and store in an array called *values*. Then, find and display the sums, minimum value and maximum value of the numbers stores in the values array.

Upload your completed C program into the Lab 8– Program 03 submission folder.

4. Steve Jobs following quote is stored in a text file called *quote.txt*. Read the content of the file and find how many times character *'i'* is appearing in the quote.

   Here is the content of the *quote.txt* file (Create your own *quote.txt* file in your computer):

   ```
   Innovation distinguishes between a leader and follower.
   ```

Upload your completed C program into the Lab 8– Program 04 submission folder.

5. Develop a C program that will read a set of numbers of type *int* from a text file called *set.txt*. Then,

output the frequency of each number (count of each number) in the file. The file contains only whole numbers, positive and negative, separated by spaces, tabs, or line breaks.

Note: Assume that you don’t know how many numbers are in the data file and your program should be able to read any numbers in the file.

Here is an example *set.txt* file and expected output from the program:

```
1 2 4 5 6 7 4 6 7 4 3
2 1 5 5 6 3 2 4
2
```

Output of the Program:

```
   Frequency of number 1: 2
   Frequency of number 2: 4
   Frequency of number 3: 2
   Frequency of number 4: 4
   Frequency of number 5: 3
   Frequency of number 6: 3
   Frequency of number 7: 2
```

Upload your completed C program into the Lab 8– Program 05 submission folder.

File Handling in C

General File Input / Output Steps

I. Create the stream via a pointer variable using the FILE structure:

```
FILE * fptr;
```

II. Open the file, associating the stream name with the file name.

```
fptr = fopen(“file-name”, “file –type”);
```

File-Type	Meaning
“r”	Open an existing file for reading only.
“w”	Open a new file for writing only. If a file with the specified filename currently exists, it will be destroyed, and a new file created in its place.
“a”	Open an existing file for appending (i.e., for adding new information at the end of the file). A new file will be created if the file with the specified filename does not exist.
“r+”	Open an existing file for both reading and writing.
“w+”	Open a new file for both reading and writing. If a file with the specified filename currently exists, it will be destroyed, and a new file created in its place.
“a+”	Open an existing file for both reading and appending. A new file will be created if the file with the specified filename does not exist.

```
fptr = fopen ("file.dat","r");
fptr = fopen ("file.dat","w");
fptr = fopen ("file.dat","a");
```

III. Read or write the data.

Read

```
int num;
char ch;
fscanf(fptr, ”%c”, &ch); → read Characters
fscanf(fptr, ”%d”, &num); → Read integer
```

Write

```
int a = 10, b = 20;
fprintf(fptr, "Hello World!\n”);
fprintf(fptr, “%d %d”, a, b);
```

IV. Close the file.

```
fclose (fptr);
```

Example 1 - Read from a text file

```
#include <stdio.h>

int main()
{
   int num;
   FILE *fptr;

   fptr = fopen("input.txt","r");

   if(fptr == NULL)
   {
      printf("Error!");
      return;
   }
      fscanf(fptr,"%d", &num);

      printf("Value of n=%d", num);
      fclose(fptr);

      return 0;
}
```

Example 2 - Write data to a text file

```
#include <stdio.h>

int main()
{
   int num;
   FILE *fptr;

   fptr = fopen("input.txt","w");

   if(fptr == NULL)
   {
      printf("Error!");
      return;
   }

   printf("Enter num: ");
   scanf("%d",&num);

   fprintf(fptr,"%d",num);
   fclose(fptr);

   return 0;
}
```