# Exercise Week 8

**Experiment 1**
Program name: print_me.c

The program below opens the file print_me.c and prints every character in the file to the screen. Understand and run this program.

```
#include <stdio.h>


int main()
{
  FILE *p;            /* p is a file pointer */
  char c;
  int success;

  p = fopen("print_me.c", "r");
  /* Open the file test.c in read-only mode and assign its handle to p */
  if (p == 0) /* Check if the file was opened correctly */
  {
   printf("Error in opening the file\n");
   return(1);
  }


  do
  {
    success = fscanf(p, "%c", &c);    /* Read next character from file */
    printf("%c", c);                  /* Print the character to screen */
  }
  while(success == 1);                /* Repeat till fscanf reads something successfu
lly */


  fclose(p);                          /* Close the file */
  return(0);
}
```

**Experiment 2.**
Program name: statistics.c

The file data01.txt contains the name and marks of students in a class. The name and the marks are separated by a comma. Download the above file to the folder Week8. Write a program to read the data from this file and print the following results: (i) Average Mark, (ii) Highest Mark and (iii) the names of all those students with highest mark.You can take hints from the program below which reads all the names and marks into arrays and prints them in the reverse order.

```
#include <stdio.h>

int main()
{
  int success,count = 0;
  char names[200][30];
  int  marks[200];
  FILE *p;

  p = fopen("data01.txt", "r");
  /* Open the file test.c in read-only mode and assign its handle to p */
  if (p == 0)   /* Check if the file was opened correctly */
  {
   printf("Error in opening the file\n");
   return(1);
  }

  while(count < 200)
  {
    success = fscanf(p, "%s %d", names[count], &marks[count] );    /* Read next name
nd marks from file */
    if(success == -1)
        break;
    count = count+1;
  }

  fclose(p);                          /* Close the file */

  count = count-1;
  while(count >= 0)
  {
    printf("%s\t%d\n", names[count], marks[count]);
    count = count - 1;
  }
```

```
  return 0;
}
```

**Experiment 3**
Program name: chessboard_while.c

Write a C program to read a positive integer n and print an n x n chess board. Use only **while** loops. If the user inputs 4, the output should be

```
O X O X
X O X O
O X O X
X O X O
```

Note: You do not need any array or pointers to do this.

**Experiment 4**
Program name: chessboard_dowhile.c

Repeat Experiment 1, this time using only **do-while** loops.

**Experiment 5**
Program name: chessboard_for.c

Repeat Experiment 1, this time using only **for** loops.

**Homework 6.**
Program name: prime_factors.c

Write a C program to read an input n and print all the prime factors of n. If the input is 12, the output should be 2,3.

**Homework 7.**
Program name: number_palindrome.c

A palindromic number is a number that remains the same when its digits are reversed. Like 131, 2552, 5, 66, etc. Write a C program to read a single positive integer and check whether it is a palindromic number.

**Experiment 8.**
Program name: binary.c

Write a C program to read a positive integer and display it in binary. For example, if the user inputs 10, you should print 1010.

**Experiment 9.**

Write a C program to **read a file name from the user** and then read the matrix from that file and store it in a 2D array. To set the size of the array during declaration, you can assume that the maximum size of the matrix is 100 x 100. Also print the matrix in its transposed form. You can use matrix1.txt, matrix2.txt, matrix3.txt and matrix4.txt as test inputs.

Your can start building from the program below.

```
#include <stdio.h>


int main()
{
  FILE *fp;
  char filename[100];


  float matrix[100][100];
  int m, n, i, j;

  /* Ask the user to give the name of the file which contains the matrix */
  printf("Enter the file name: ");
  scanf("%s", filename);

  /* Open the file for reading and check if it was opened successfully */
  fp = fopen(filename, "r");
  if (fp == 0)
  {
   printf("Error in opening the file %s.\n", filename);
   return(1);
  }
  fscanf(fp, "%d", &m);
  fscanf(fp, "%d", &n);
  for(i = 0; i < m; i++)
  {
    for(j = 0; j < n; j++)
    {
      fscanf(fp, "%f", &matrix[i][j]); /* Read one entry */
    }
  }
  /* Insert your code to print the matrix in transpose form here */


  fclose(fp); /* Close the file */
  return(0);
}
```

**Experiment 10.**
Program name: plot_fun.c

The following program generates a data file with x and y values of one period of a sine wave. Understand the program and its output. You will need the -lm flag to compile. That is, use the command

```
gcc plot_fun.c -o plot_fun.out -lm
```

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


int main()
{
  FILE *p;
  float x, y;
  int i;

  p = fopen("data03.txt", "w"); /* Open a file to write the final data */
  if (p == 0)                   /* Check if the file was opened correctly */
  {
    printf("Error in opening the file\n");
    return(1);
  }
  for(i = 0; i <=100; i++)
  {
    x = 2.0 * 3.14159265 * i / 100.0; /*x takes values from 0 to 2pi in 101 steps */
    y = sin(x);
    fprintf(p, "%f\t%f\n", x, y);     /* Write the data to file */
  }


  fclose(p); /* Close the file */
  return(0);
}
```

Also plot the output file using the command

```
gnuplot -p -e "plot 'data03.txt'"
```

Modify the above program to plot various functions like cos(x), sin^2(x), sin(x) + sin(2x), sin(1/x) etc.