Name: Renita Kurian	SRN: PES1UG20CS331	Section: N
	Date: 10/07/21	Week Number: 9

1 Write a C program to merge contents of two files into a third file.

Program:

P1.c

```c
//Merging two text files into a third
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE* fp1;
    FILE* fp2;
    FILE* fp3;
    char f1n[32];
    char f2n[32];
    char f3n[32];
    char c;
    printf("Input the full name of the first file (input) : ");
    scanf(" %[^\n]%*c", f1n);
    printf("Input the full name of the second file (input) : ");
    scanf(" %[^\n]%*c", f2n);
    printf("Input the full name of the third file (output) : ");
    scanf(" %[^\n]%*c", f3n);
    fp1 = fopen(f1n, "r");
    fp2 = fopen(f2n, "r");
    fp3 = fopen(f3n, "w");
    if (fp1 == NULL || fp2 == NULL || fp3 == NULL)
    {
        printf("An error occurred in opening the requested file(s) ...\n");
    }
    while ((c = fgetc(fp1)) != EOF)
        fputc(c, fp3);
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp3);
    printf("%s and %s were merged into %s successfully ... \n", f1n, f2n, f3n);
    fclose(fp1);
    fclose(fp2);
    fclose(fp3);
    return 0;
}
```

T1.txt

```
T1.txt
1    C Programming
2
```

T2.txt

```
T2.txt
1    Hello world
2
```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>gcc P1.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>a
Input the full name of the first file (input) : T1.txt
Input the full name of the second file (input) : T2.txt
Input the full name of the third file (output) : T3.txt
T1.txt and T2.txt were merged into T3.txt successfully ...

C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>
```

T3.txt

```
T3.txt
1    C Programming
2    Hello world
3
```

2	Write a C program to write multiple lines in a text file.

Program:

P2.c > main()

```c
//Writing multiple lines into a text file
#include <stdio.h>
int main()
{
    FILE* fp;
    char line[64];
    int i;
    int n;
    fp = fopen("f1.txt", "w");
    if (fp == NULL) printf("An error occurred in opening the requested file ...\n");
    printf("Input the number of lines to be written : ");
    scanf("%d", &n);
    printf("\n");
    printf("Input the contents of the file ... \n");
    for (i = 0; i < n; i++)
    {
        scanf(" %[^\n]%*c", line);
        fputs(line, fp);
        fputs("\n", fp);
    }
    fclose(fp);
    return 0;
}

```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>gcc P2.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>a
Input the number of lines to be written : 3

Input the contents of the file ...
Hello
World
Coding
```

F1.txt

f1.txt

```
Hello
World
Coding

```

3	Write a program to sort positive integers in the ascending order using insertion sort

Program:

P3.c > main()

```c
//Sorting an integer array using insertion sort
#include <math.h>
#include <stdio.h>
void insertionSort(int a[], int n) // insertion sorting function
{
    int i;
    int j;
    int k;
    for (i = 1; i < n; i++)
    {
        k = a[i];
        for (j = i - 1; (j >= 0) && (a[j] > k); j--)
            a[j + 1] = a[j];
        a[j + 1] = k;
    }
}

void inputArray(int a[], int n) // array input function
{
    printf("Input the integer elements with spacing : ");
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
    printf("\n");
}

```

```c
void displayArray(int a[], int n) // array printing function
{
    int i;
    for (i = 0; i < n; i++) printf("%d ", a[i]);
    printf("\n");
}

int main()
{
    int n;
    printf("Input the size of the array : ");
    scanf(" %d", &n);
    printf("\n");
    int a[100];
    inputArray(a, n);
    insertionSort(a, n);
    printf("Sorted array : ");
    displayArray(a, n);
    return 0;
}
```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>gcc P3.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>a
Input the size of the array : 5

Input the integer elements with spacing : 4 1 6 2 3

Sorted array : 1 2 3 4 6
```

4	Write a bubblesort program to sort students details based on students roll number/name in the ascending order using array of pointers, by taking input from csv file and using callback to call two functions i)sort based on roll number ii) sort based on name.
	Program:

P4.c > main()

```c
//Bubble sorting records from a .csv file
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
typedef struct student // structure to hold the .csv file's rows
{
    int roll_no;
    char name[128];
} STUDENT_T;

void swap(STUDENT_T** p, STUDENT_T** q) // array element swap function
{
    STUDENT_T* temp = *p;
    *p = *q;
    *q = temp;
}

void display(STUDENT_T* p[], int n) // display array of pointers
{
 for (int i = 0; i < n; i++)
    printf(" %d %s\n", p[i]->roll_no, p[i]->name);
}

void sort_on_roll_no(STUDENT_T* p[], int n) // bubble sort based on rollno
{
    int i, pos, j;
    for (i = 0; i < n - 1; i++)
    {
        pos = i;
        for (j = i + 1; j < n; j++)
            if (p[pos]->roll_no > p[j]->roll_no)
                pos = j;
        if (pos != i)
            swap(&p[pos], &p[i]);
    }
}
```

```c
void sort_on_name(STUDENT_T* p[], int n) // bubble sort based on name
{
    int i, pos, j;
    for (i = 0; i < n - 1; i++)
    {
        pos = i;
        for (j = i + 1; j < n; j++)
            if (strcmp(p[pos]->name, p[j]->name) > 0)
                pos = j;
        if (pos != i)
            swap(&p[pos], &p[i]);
    }
}
int main()
{
    FILE* f1 = fopen("stud.csv", "r");
    if (f1 == NULL)
        perror("The file could not be opened ..."); // file error
    else
    {
        STUDENT_T st[1024];
        STUDENT_T* p[1024];
        char line[128];
        int i;
        int n;
        int ch;
        fgets(line, 128, f1); // clearing first row
        for (i = 0; fgets(line, 128, f1) != NULL; i++)
        {
            char* r = strtok(line, ",");
            char* name = strtok(NULL, ",");
            st[i].roll_no = atoi(r);
            strcpy(st[i].name, name);
            p[i] = &st[i];
        }
        n = i; // number of elements in the arrays
        A :
        printf("Menu \n1. Sort by roll # \n2. Sort by name\n3. Exit\nInput your choice ...");
        scanf("%d", &ch);
        printf("\n");
        switch (ch)
        {
            case 1 :
                sort_on_roll_no(p, n);
                display(p, n);
                break;
            case 2 :
                sort_on_name(p, n);
```

```
                display(p, n);
                break;
            case 3 :
                exit(0);
                break;
            default :
                printf("Invalid response. Try again ... \n\n");
        }
        goto A;
    }
    return 0;
}
```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>gcc P4.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>a
Menu
1. Sort by roll #
2. Sort by name
3. Exit
Input your choice ...1

 3 x

 3 c

 4 z

 5 a

 6 b

 7 y

 12 d

 14 q

 18 t

 26 e
```

```
Menu
1. Sort by roll #
2. Sort by name
3. Exit
Input your choice ...2

 5 a

 6 b

 3 c

 12 d

 26 e

 14 q

 18 t

 3 x

 7 y

 4 z

Menu
1. Sort by roll #
2. Sort by name
3. Exit
Input your choice ...3


C:\Users\Renita Kurian\Documents\Academic\C Lab\W9>
```