

Week 9

ANIRUDH ROWJEE PES2UG20CS050

Task 1

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

main.c

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

int main()
{
    char first_filename[100], second_filename[100], target_filename[100];

    printf("Enter the First Filename >> ");
    scanf("%s", first_filename);
    printf("Enter the Second Filename >> ");
    scanf("%s", second_filename);
    printf("Enter the Target Filename >> ");
    scanf("%s", target_filename);

    // error checking
    if (merge("target.txt", "first.txt", "second.txt"))
    {
        printf("Could not merge!\n");
    }
    else
    {
        printf("Merged Successfully!\n");
    }

    return 0;
}
```

utils.h

```
int merge(char* target, char* first, char* second);
```

utils.c

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

// function to output the content of two files in one file
int merge(char* target, char* first, char* second)
{
    // open all three files and check for errors
    FILE* target_file = fopen(target, "w");
    FILE* first_file = fopen(first, "r");
```

```

FILE* second_file = fopen(second, "r");

char c;
while ((c = fgetc(first_file)) != EOF)
{
    fputc(c, target_file);
}
while ((c = fgetc(second_file)) != EOF)
{
    fputc(c, target_file);
}
// fputc EOF, target_file);

// close all three files
int t = fclose(target_file);
int f = fclose(first_file);
int s = fclose(second_file);

if (s == EOF || t == EOF || f == EOF)
{
    printf("Error closing the files!\n");
    return 1;
}
return 0;
}

```

Console

```

nidavellir :: UE20CS152/week7/t1 » cat first.txt
Hi! This is a test to see if this program works with multiple
lines and multiple files.

This is the first file.
nidavellir :: UE20CS152/week7/t1 » cat second.txt
Hello! This is a test to see if this program works with multiple
lines and multiple files.

This is the second file.
nidavellir :: UE20CS152/week7/t1 » cat target.txt
cat: target.txt: No such file or directory
nidavellir :: UE20CS152/week7/t1 » ./main
Enter the First Filename >> first.txt
Enter the Second Filename >> second.txt
Enter the Target Filename >> target.txt
Merged Successfully!
nidavellir :: UE20CS152/week7/t1 » cat target.txt
Hi! This is a test to see if this program works with multiple
lines and multiple files.

This is the first file.
Hello! This is a test to see if this program works with multiple
lines and multiple files.

This is the second file.
nidavellir :: UE20CS152/week7/t1 »

```

Task 2

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

main.c

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

int main()
{
    int count;
    char filename[50];

    printf("Enter Filename to write to >> ");
    scanf("%s", filename);

    printf("Enter Number of Lines to write >> ");
    scanf("%d", &count);

    // extremely weird bug where not doing this skips over the first
    // iteration of the loop to scan a new loop
    // dispose of the newline character by using a single getchar()
    getchar();

    printf("Reading lines...\n");

    int status = AppendNLines(filename, count);

    if (status == 0)
    {
        printf("%d Lines succesfully written!\n", count);
    }
    else
    {
        printf("Error!\n");
    }

    return 0;
}
```

utils.h

```
int AppendNLines(char* filename, int line_count);
```

utils.c

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

// function to append N lines to a file
int AppendNLines(char* filename, int line_count)
{
    FILE* outfile = fopen(filename, "w");
    if (outfile == NULL)
    {
        printf("File could not be opened!\n");
        return 1;
    }

    for (int i = 0; i < line_count; ++i)
    {
```

```

        printf("Line %d > \n", i+1);
        char c[100];
        fflush(stdin);
        fgets(c, 100, stdin);
        fputs(c, outfile);
    }

    fclose(outfile);
    return 0;
}

```

Console

```

nidavellir :: UE20CS152/week7/t2 » cat test1.txt
cat: test1.txt: No such file or directory
nidavellir :: UE20CS152/week7/t2 » ./main
Enter Filename to write to >> test1.txt
Enter Number of Lines to write >> 3
Reading lines...
Line 1 >
Hello, world! This
Line 2 >
is Anirudh Rowjee and
Line 3 >
this is my program.
3 Lines succesfully written!
nidavellir :: UE20CS152/week7/t2 » cat test1.txt
Hello, world! This
is Anirudh Rowjee and
this is my program.
nidavellir :: UE20CS152/week7/t2 »

```

Task 3

Write a program to sort integers in the ascending order using bubble sort.

main.c

```

#include <stdio.h>
#include "utils.h"

int main()
{

    int count;
    printf("Enter the number of items in the list >> ");
    scanf("%d", &count);

    int temp[count];
    for (int i = 0; i < count; i++)
    {
        printf("Enter the number at index %d >> ", i);
        scanf("%d", &temp[i]);
    }

    bubblesort(temp, count);

    printf("The sorted list is >> \n");
    for (int i = 0; i < count; i++)

```

```

    {
        printf("%d\t", temp[i]);
    }
    printf("\n");

    return 0;
}

```

utils.h

```
int bubblesort(int* numbers, int length);
```

utils.c

```

#include <stdio.h>
#include "utils.h"

int bubblesort(int* numbers, int length)
{
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length - i - 1; ++j)
        {
            if (numbers[j] >= numbers[j+1])
            {
                int temp = numbers[j];
                numbers[j] = numbers[j+1];
                numbers[j+1] = temp;
            }
        }
    }
    return 0;
}

```

Console

```

nidavellir :: UE20CS152/week7/t3 » ./main
Enter the number of items in the list >> 6
Enter the number at index 0 >> -23
Enter the number at index 1 >> 234
Enter the number at index 2 >> 574
Enter the number at index 3 >> 23434
Enter the number at index 4 >> 325581
Enter the number at index 5 >> -23423
The sorted list is >>
-23423  -23    234    574    23434  325581
nidavellir :: UE20CS152/week7/t3 » ./main
Enter the number of items in the list >> 7
Enter the number at index 0 >> -234
Enter the number at index 1 >> 237423
Enter the number at index 2 >> 983845
Enter the number at index 3 >> 234
Enter the number at index 4 >> 2394
Enter the number at index 5 >> 983475
Enter the number at index 6 >> -234
The sorted list is >>
-234    -234    234    2394    237423  983475  983845
nidavellir :: UE20CS152/week7/t3 »

```

Task 4

Write A Bubblesort program to sort student details based on

1. Name
2. Roll Number

in the ascending order using an array of pointers. Take input from a CSV File, and use callbacks to sort the list.

main.c

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

int main()
{

    printf("\n** Student Manager **\n");

    // Initialize pointer array of student objects
    student* students[100];

    // read students from file
    int student_count = parse_csv_file("test.csv", students);

    int choice = 1;

    while (choice != 0)
    {
        printf(
            "Enter 1 to sort by Name, enter 2 to sort by Roll Number, 0 to exit>> "
        );
        scanf("%d", &choice);

        // declare a NULL function pointer
        int (*comparator)(student* f, student* s) = NULL;

        switch (choice)
        {
            case 1:
                comparator = &compare_by_name;
                sort_student_details(student_count, students, comparator);
                print_student_details(student_count, students);
                break;
            case 2:
                comparator = &compare_by_rollno;
                sort_student_details(student_count, students, comparator);
                print_student_details(student_count, students);
                break;
            case 0:
                printf("Thank you for using Student Manager!\n");
                break;
            default:
                printf("Invalid Argument!\n");
                break;
        }
    }

    free_pointer_array(student_count, students);
```

```
    return 0;
}
```

utils.h

```
typedef struct student_details_t
{
    int roll_number;
    char* name;
} student;

// function to parse the CSV file into the array of pointers
int parse_csv_file(char* filename, student** destination);

// show all the details
void print_student_details(int count, student** student_pointer_array);

int compare_by_rollno(student* student1, student* student2);
int compare_by_name(student* student1, student* student2);

// accept a callback
void sort_student_details(
    int student_count,
    student** students,
    int (*sort_function)(student* first, student* second)
);

// memory safety
int free_pointer_array(int count, student** student_pointer_array);
```

utils.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "utils.h"

/*
 * Read from the CSV file at <filename> and tokenize
 * store each in destination
 */
int parse_csv_file(char* filename, student** destination)
{
    FILE* csv_file = fopen(filename, "r");
    if (csv_file == NULL)
    {
        printf("Couldn't open file!\n");
        return 1;
    }

    // calculate sizes
    fseek(csv_file, 0, SEEK_END);
    long fsize = ftell(csv_file);
    fseek(csv_file, 0, SEEK_SET);
    // create a buffer to read the contents of the file from
    char* file_buffer = malloc((fsize + 1)*sizeof(char));
```

```

// read everything into file_buffer
fread(file_buffer, 1, fsize, csv_file);

// fputs(file_buffer, stdout);

// begin the parsing
char* row;
int counter = 0;
row = strtok(file_buffer, "\n");
while (row != NULL)
{
    // printf("Tokenizer START: %s\n", row);
    // printf("Parsing student %d\n", counter+1);

    // since we have a guarantee of the number of columns in
    // the CSV file, we need not call strtok() in a while loop

    // this is necessary because the string variable might
    // have been garbage-collected and overwritten by the time it's read
    // at its memory location
    char* name = malloc(sizeof(char)*20);
    int roll_number = 0;

    char temp_row[100];
    strcpy(temp_row, row);

    // printf("%s\n", temp_row);
    sscanf(temp_row, "%[^,],%d", name, &roll_number);

    // convert from string to integer
    // printf("READ (%s, %d)\n", name, roll_number);

    // allocate new memory for the pointer
    student* new_student = malloc(sizeof(student));

    new_student->name = name;
    new_student->roll_number = roll_number;

    /*
    printf(
        "New student (%d, %s) at %p\n",
        new_student->roll_number, new_student->name,
        new_student
    );
    */

    destination[counter] = new_student;
    counter++;
    row = strtok(NULL, "\n");
    // printf("Tokenizer END: %s\n", row);
}

fclose(csv_file);
free(file_buffer);
printf("Successfully parsed into array\n");
return counter;
}

// function to print the array of students formatted
void print_student_details(int count, student** student_pointer_array)
{
    printf("----- STUDENT DETAILS ----- \n");
    for (int i = 0; i < count; i++)
    {
        student* temp = student_pointer_array[i];
    }
}

```



```

        printf("%5d | %10s\n",
               temp->roll_number,
               temp->name
               );
    }
}

int free_pointer_array(int count, student** student_pointer_array)
{
    for (int i = 0; i < count; i++)
    {
        free(student_pointer_array[i]->name);
        free(student_pointer_array[i]);
    }
    printf("Successfully cleaned up!\n");
    return 0;
};

// comparator function for roll numbers
int compare_by_rollno(student* student1, student* student2)
{
    if (student1->roll_number > student2->roll_number)
    {
        return 1;
    }
    else
    {
        // save swapping on equal
        return 0;
    }
}

// comparator function for names
int compare_by_name(student* student1, student* student2)
{
    if (strcmp(student1->name, student2->name) > 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

/*
 * This function accepts an array of pointers and a
 * comparator function that acts on two pointers.
 * It uses bubble sort on the array of pointers to
 * sort them on the basis of the comparator function provided.
 */
void sort_student_details(
    int student_count,
    student** students,
    int (*sort_function)(student* first, student* second)
)
{
    for (int i = 0; i < student_count; i++)
    {
        for (int j = 0; j < student_count - i - 1; j++)
        {
            if (sort_function(students[j], students[j+1]))
            {

```

```

        student* temp = students[j+1];
        students[j+1] = students[j];
        students[j] = temp;
    }
}
}
}

```

Console

```

nidavellir :: UE20CS152/week7/t4 » cat test.csv
anirudh,20
shreyas,11
ramdev,30
sarath,50
somesh,22
rahul,41
ajay,52
nidavellir :: UE20CS152/week7/t4 » ./main

** Student Manager **
Successfully parsed into array
Enter 1 to sort by Name, enter 2 to sort by Roll Number, 0 to exit>> 1
----- STUDENT DETAILS -----
  52 |      ajay
  20 |    anirudh
  41 |      rahul
  30 |    ramdev
  50 |    sarath
  11 |    shreyas
  22 |    somesh
Enter 1 to sort by Name, enter 2 to sort by Roll Number, 0 to exit>> 2
----- STUDENT DETAILS -----
  11 |    shreyas
  20 |    anirudh
  22 |    somesh
  30 |    ramdev
  41 |    rahul
  50 |    sarath
  52 |    ajay
Enter 1 to sort by Name, enter 2 to sort by Roll Number, 0 to exit>> 0
Thank you for using Student Manager!
Successfully cleaned up!
nidavellir :: UE20CS152/week7/t4 »

```