Continuous Assessment – 02 Report

Only for Course Teacher						
		Needs Improvement	**Developing**	**Sufficient**	**Above Average**	**Total Mark**
Allocate mark & Percentage		**25%**	**50%**	**75%**	**100%**	**15**
Problem Analysis	**03**					
Solution Design	**02**					
Code Development	**06**					
Accuracy	**04**					
Total obtained mark						
Comments						

Semester: Spring 2025

Student Name: Raktim Bain

Student ID: 242-35-560

Batch: 43 **Section:** M1

Course Code: SE 133 **Course Name:** Software Development Capstone Project

Course Teacher Name: Nazmus Sakib

Designation: Lecturer

Submission Date: 25/06/2025

Task 4: Conversion of Multiple Temperatures from Celsius to Fahrenheit

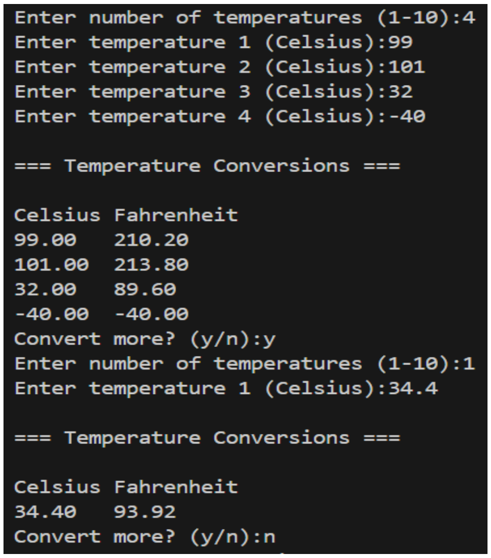
Code:

```c
#include<stdio.h>
float CelsiusToFahrenheit(float c);
int main()
{   int n;
    char cond;
    do
    {
       while(1)
       {
           printf("Enter number of temperatures (1-10):");
           scanf("%d",&n);
           if( n<1 || n>10 )
           {
            printf("Invalid Operation!\n");
            continue;
           }
           else { break; }
       }

       float temp[n];
       for( int i=0; i<n; i++ )
       {
           printf("Enter temperature %d (Celsius):",i+1);
           scanf("%f",&temp[i]);
       }

       printf("\n=== Temperature Conversions === \n");
       printf("\nCelsius\tFahrenheit\n");

       for( int i=0; i<n; i++)
       {
           printf("%.2f\t%.2f\n",temp[i],CelsiusToFahrenheit( temp[i] ));
       }

       printf("Convert more? (y/n):");
       scanf(" %c",&cond);
    } while (cond=='Y' || cond=='y');


}
float CelsiusToFahrenheit ( float c )
```

```
{
    return c*9/5 + 32;
}
```

Program Explanation: This C program converts multiple temperatures from Celsius to Fahrenheit. It asks the user how many temperatures to convert (1-10), takes those inputs, and displays the converted values using the formula F = C × 9/5 + 32. After displaying the results, it asks if the user wants to convert more. The process repeats until the user chooses not to continue.

Output:

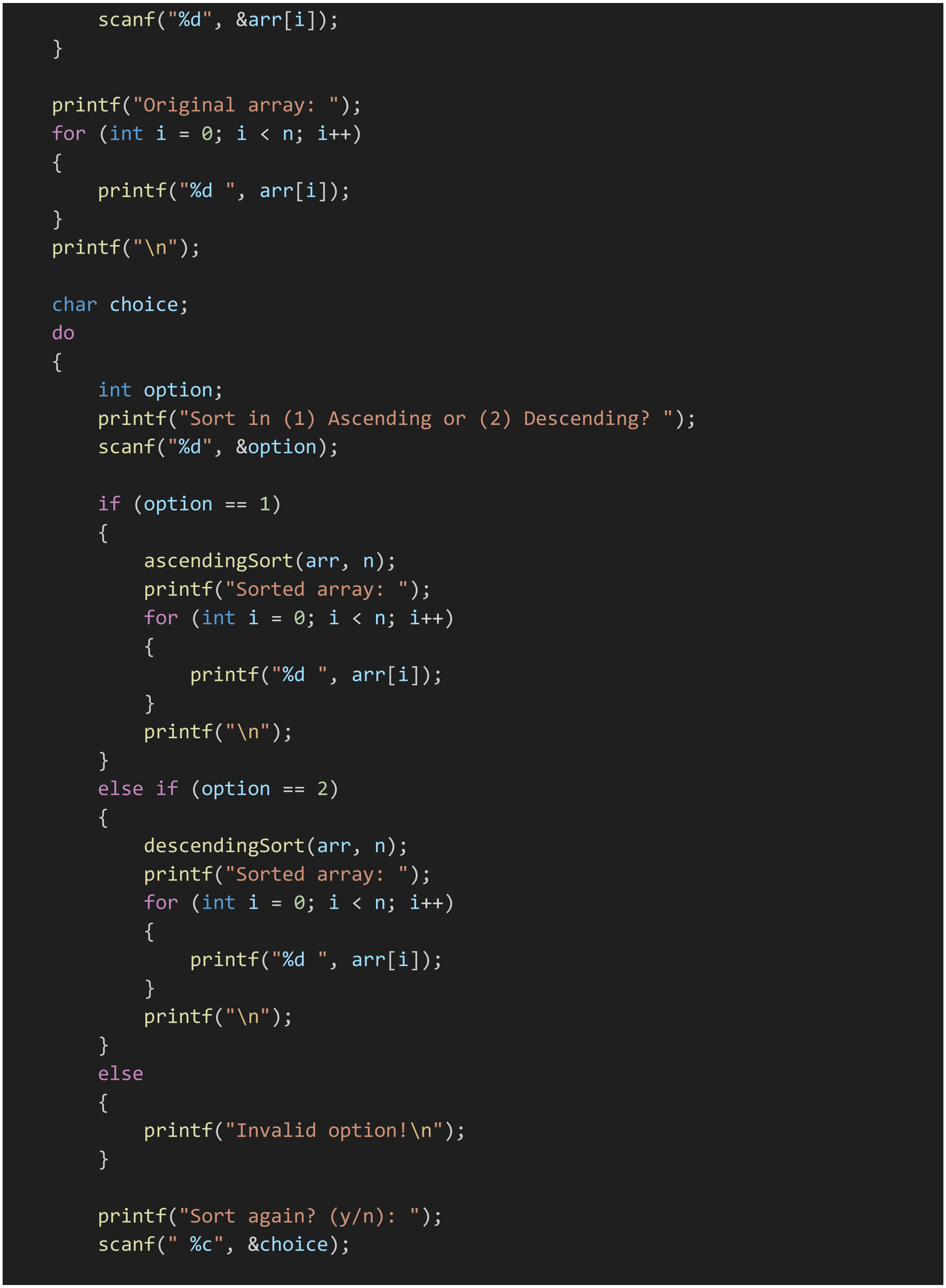


```
Enter number of temperatures (1-10):4
Enter temperature 1 (Celsius):99
Enter temperature 2 (Celsius):101
Enter temperature 3 (Celsius):32
Enter temperature 4 (Celsius):-40

=== Temperature Conversions ===

Celsius Fahrenheit
99.00   210.20
101.00  213.80
32.00   89.60
-40.00  -40.00
Convert more? (y/n):y
Enter number of temperatures (1-10):1
Enter temperature 1 (Celsius):34.4

=== Temperature Conversions ===

Celsius Fahrenheit
34.40   93.92
Convert more? (y/n):n
```

Task 5: Bubble Sorting of Numerical Values

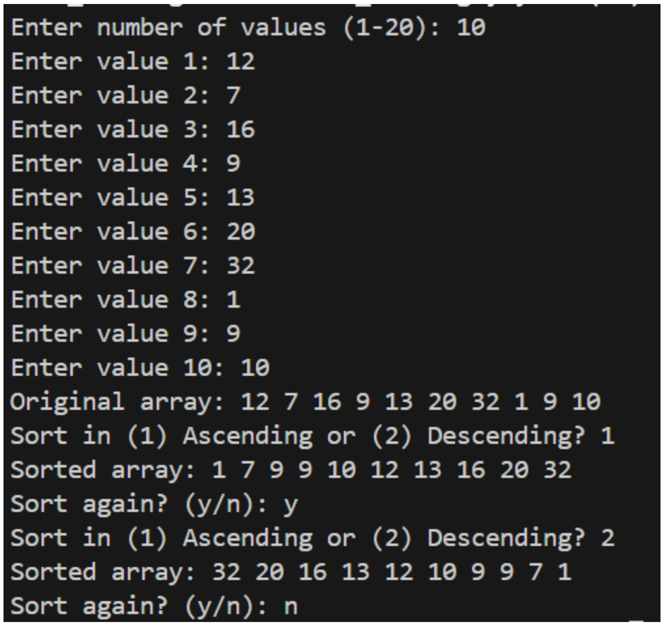
Code:

```
#include<stdio.h>

void ascendingSort(int arr[], int n);
void descendingSort(int arr[], int n);

int main()
{
    int n;
    printf("Enter number of values (1-20): ");
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter value %d: ", i + 1);
```



```c
        scanf("%d", &arr[i]);
    }

    printf("Original array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");

    char choice;
    do
    {
        int option;
        printf("Sort in (1) Ascending or (2) Descending? ");
        scanf("%d", &option);

        if (option == 1)
        {
            ascendingSort(arr, n);
            printf("Sorted array: ");
            for (int i = 0; i < n; i++)
            {
                printf("%d ", arr[i]);
            }
            printf("\n");
        }
        else if (option == 2)
        {
            descendingSort(arr, n);
            printf("Sorted array: ");
            for (int i = 0; i < n; i++)
            {
                printf("%d ", arr[i]);
            }
            printf("\n");
        }
        else
        {
            printf("Invalid option!\n");
        }

        printf("Sort again? (y/n): ");
        scanf(" %c", &choice);
    } while (choice == 'y' || choice == 'Y');

    return 0;
}

void ascendingSort(int arr[], int n)
{
    for ( int i=0; i < n-1; i++ )
    {
        for ( int j=0; j < n-1-i; j++ )
        {
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void descendingSort(int arr[], int n)
{
    for (int i=0; i < n-1; i++ )
    {
        for ( int j=0; j < n-1-i; j++ )
        {
            if (arr[j] < arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

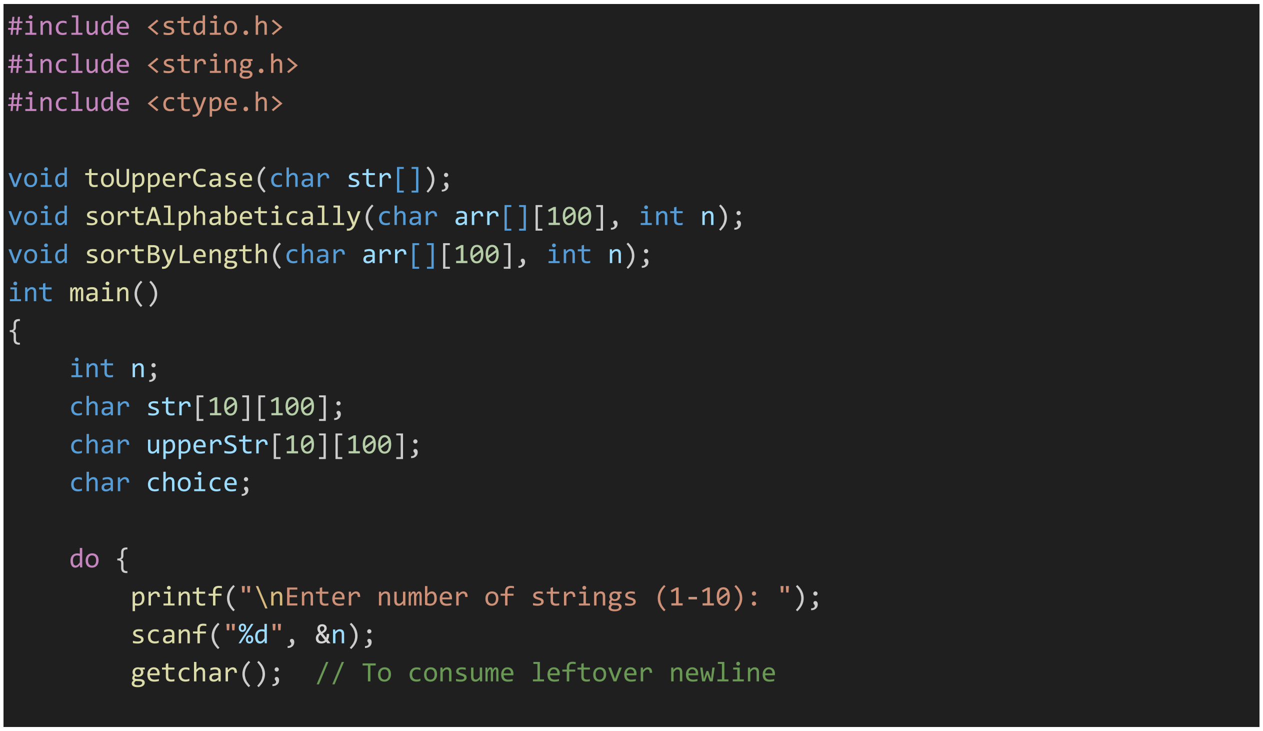
Program Explanation: This C program takes a set of integer values from the user and allows them to sort the values in either ascending or descending order as many times as they want. The program first asks the user to enter the number of values, with a limit of 1 to 20. It then reads those values into an array and displays the original array. After that, the user is given the choice to sort the array in ascending or descending order. Depending on the user's choice, the program calls either the ascendingSort() or descendingSort() function, both of which use a bubble sort algorithm to rearrange the elements. Once sorted, the array is printed. The user is then asked if they want to sort again, and the process continues until the user chooses not to. This structure allows the user to experiment with different sorting orders on the same set of numbers without having to re-enter them.



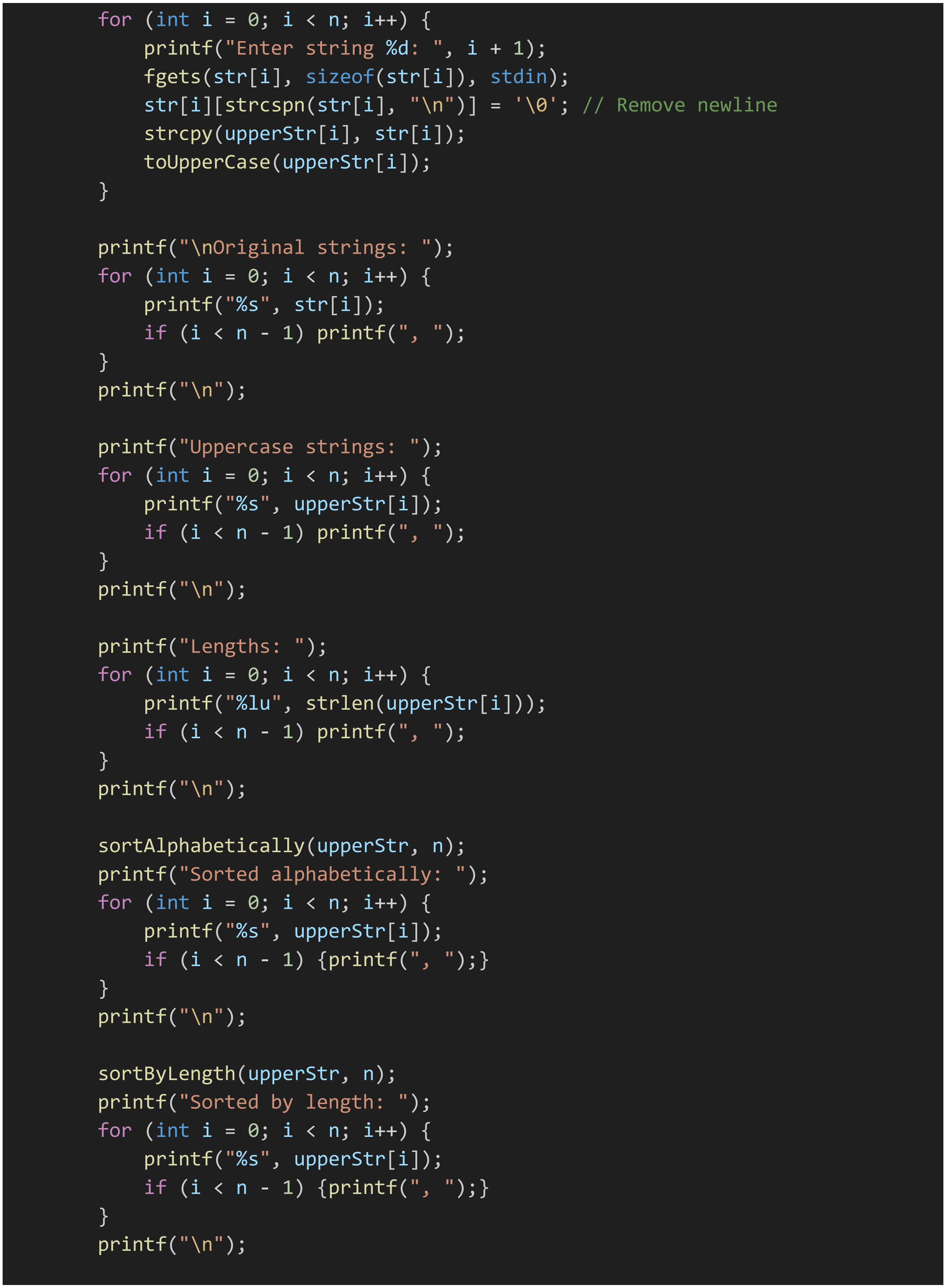
```
Enter number of values (1-20): 10
Enter value 1: 12
Enter value 2: 7
Enter value 3: 16
Enter value 4: 9
Enter value 5: 13
Enter value 6: 20
Enter value 7: 32
Enter value 8: 1
Enter value 9: 9
Enter value 10: 10
Original array: 12 7 16 9 13 20 32 1 9 10
Sort in (1) Ascending or (2) Descending? 1
Sorted array: 1 7 9 9 10 12 13 16 20 32
Sort again? (y/n): y
Sort in (1) Ascending or (2) Descending? 2
Sorted array: 32 20 16 13 12 10 9 9 7 1
Sort again? (y/n): n
```

Task 6: String Sorting in Uppercase and Length of String

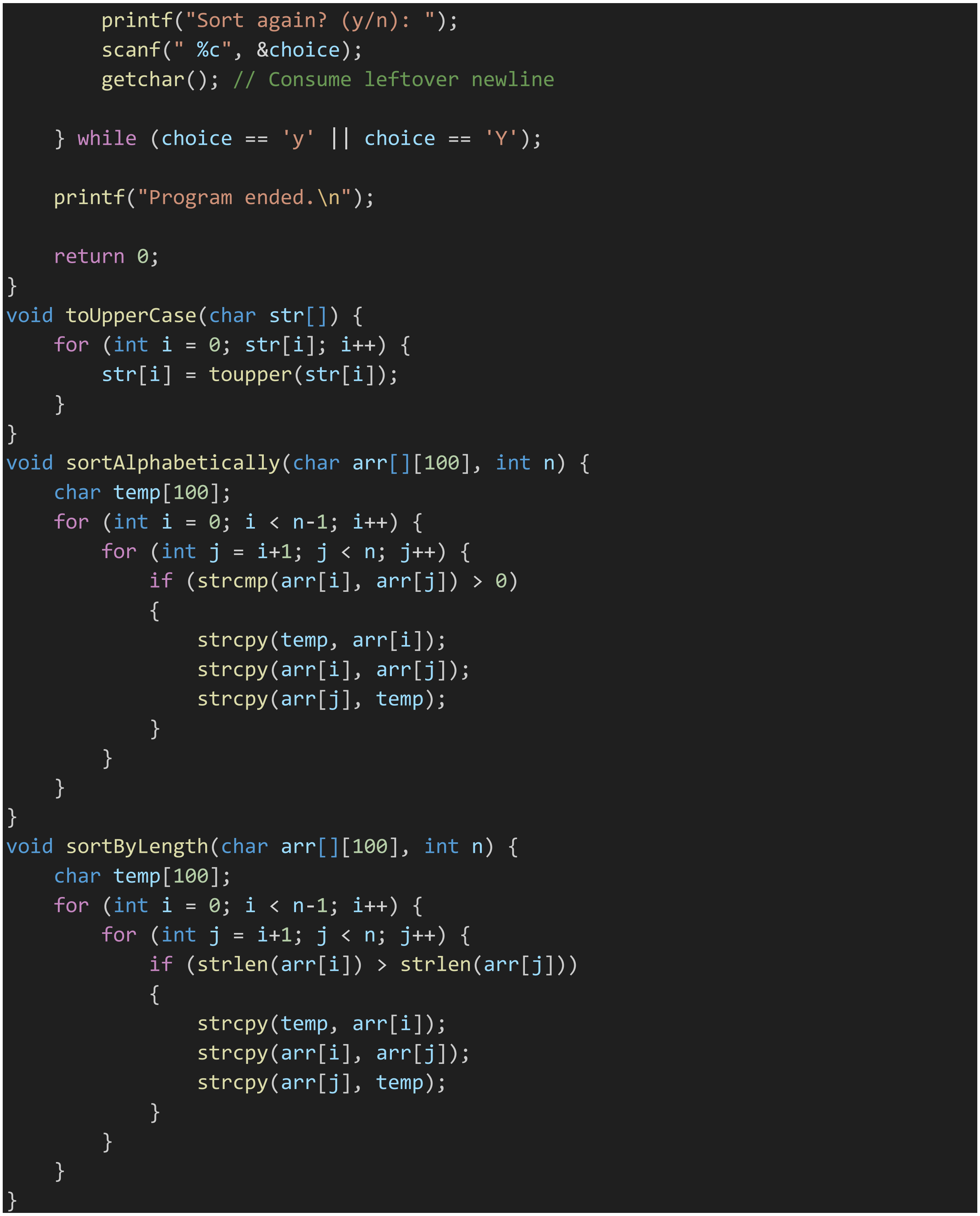
Code:



```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

void toUpperCase(char str[]);
void sortAlphabetically(char arr[][100], int n);
void sortByLength(char arr[][100], int n);
int main()
{
    int n;
    char str[10][100];
    char upperStr[10][100];
    char choice;

    do {
        printf("\nEnter number of strings (1-10): ");
        scanf("%d", &n);
        getchar();  // To consume leftover newline
```



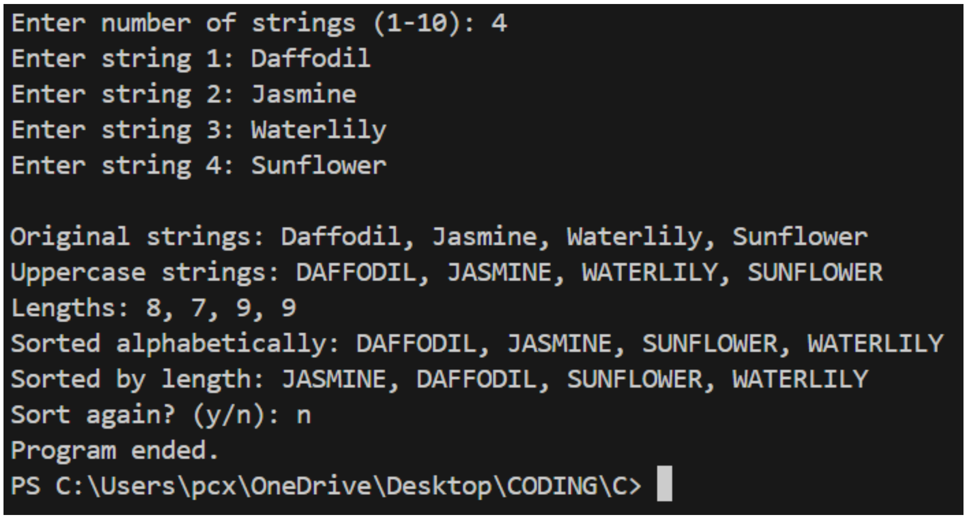
```
    for (int i = 0; i < n; i++) {
        printf("Enter string %d: ", i + 1);
        fgets(str[i], sizeof(str[i]), stdin);
        str[i][strcspn(str[i], "\n")] = '\0'; // Remove newline
        strcpy(upperStr[i], str[i]);
        toUpperCase(upperStr[i]);
    }

    printf("\nOriginal strings: ");
    for (int i = 0; i < n; i++) {
        printf("%s", str[i]);
        if (i < n - 1) printf(", ");
    }
    printf("\n");

    printf("Uppercase strings: ");
    for (int i = 0; i < n; i++) {
        printf("%s", upperStr[i]);
        if (i < n - 1) printf(", ");
    }
    printf("\n");

    printf("Lengths: ");
    for (int i = 0; i < n; i++) {
        printf("%lu", strlen(upperStr[i]));
        if (i < n - 1) printf(", ");
    }
    printf("\n");

    sortAlphabetically(upperStr, n);
    printf("Sorted alphabetically: ");
    for (int i = 0; i < n; i++) {
        printf("%s", upperStr[i]);
        if (i < n - 1) {printf(", ");}
    }
    printf("\n");

    sortByLength(upperStr, n);
    printf("Sorted by length: ");
    for (int i = 0; i < n; i++) {
        printf("%s", upperStr[i]);
        if (i < n - 1) {printf(", ");}
    }
    printf("\n");
```



```
        printf("Sort again? (y/n): ");
        scanf(" %c", &choice);
        getchar(); // Consume leftover newline

    } while (choice == 'y' || choice == 'Y');

    printf("Program ended.\n");

    return 0;
}
void toUpperCase(char str[]) {
    for (int i = 0; str[i]; i++) {
        str[i] = toupper(str[i]);
    }
}
void sortAlphabetically(char arr[][100], int n) {
    char temp[100];
    for (int i = 0; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (strcmp(arr[i], arr[j]) > 0)
            {
                strcpy(temp, arr[i]);
                strcpy(arr[i], arr[j]);
                strcpy(arr[j], temp);
            }
        }
    }
}
void sortByLength(char arr[][100], int n) {
    char temp[100];
    for (int i = 0; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (strlen(arr[i]) > strlen(arr[j]))
            {
                strcpy(temp, arr[i]);
                strcpy(arr[i], arr[j]);
                strcpy(arr[j], temp);
            }
        }
    }
}
```

Program Explanation: This C program takes multiple strings from the user, converts them to uppercase, and allows sorting them both alphabetically and by length. The user first inputs how many strings they want to enter, with a limit of 1 to 10. For each string, the program reads the input using `fgets` and removes the trailing newline character. It then copies the original string into another array and converts that copy to uppercase using the `toUpperCase` function. After all strings are entered, the program displays the original strings, the uppercase versions, and their lengths. It then sorts the uppercase strings alphabetically using the `sortAlphabetically` function and displays the sorted result. After that, the strings are sorted again by length using the `sortByLength` function, and the sorted list is displayed. At the end, the user is asked whether they want to repeat the entire process. If they choose 'y' or 'Y', the program runs again with new strings; otherwise, it terminates. This structure allows the user to repeatedly test string input, conversion, and sorting operations in a clear, interactive way.

Output:



```
Enter number of strings (1-10): 4
Enter string 1: Daffodil
Enter string 2: Jasmine
Enter string 3: Waterlily
Enter string 4: Sunflower

Original strings: Daffodil, Jasmine, Waterlily, Sunflower
Uppercase strings: DAFFODIL, JASMINE, WATERLILY, SUNFLOWER
Lengths: 8, 7, 9, 9
Sorted alphabetically: DAFFODIL, JASMINE, SUNFLOWER, WATERLILY
Sorted by length: JASMINE, DAFFODIL, SUNFLOWER, WATERLILY
Sort again? (y/n): n
Program ended.
PS C:\Users\pcx\OneDrive\Desktop\CODING\C>
```