Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

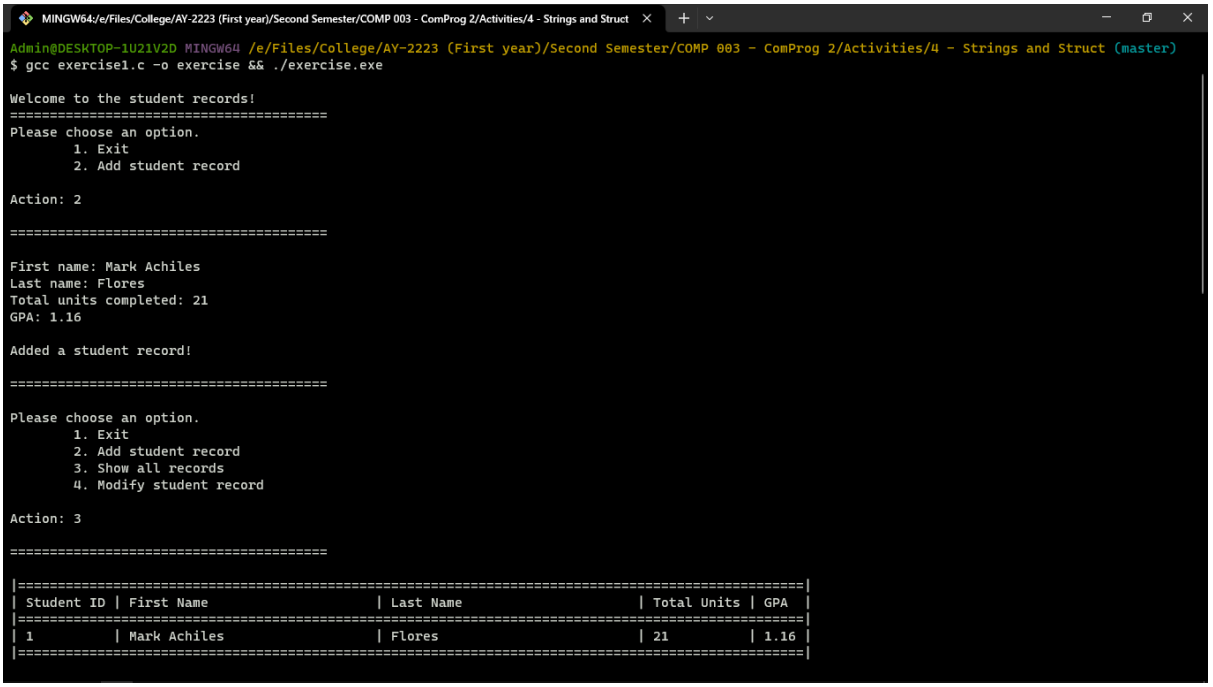
1. Create a template for a structure of student record consisting of five fields: Student ID(int), first name(string), last name(string), total units completed(int), and accumulated grade point average(float).

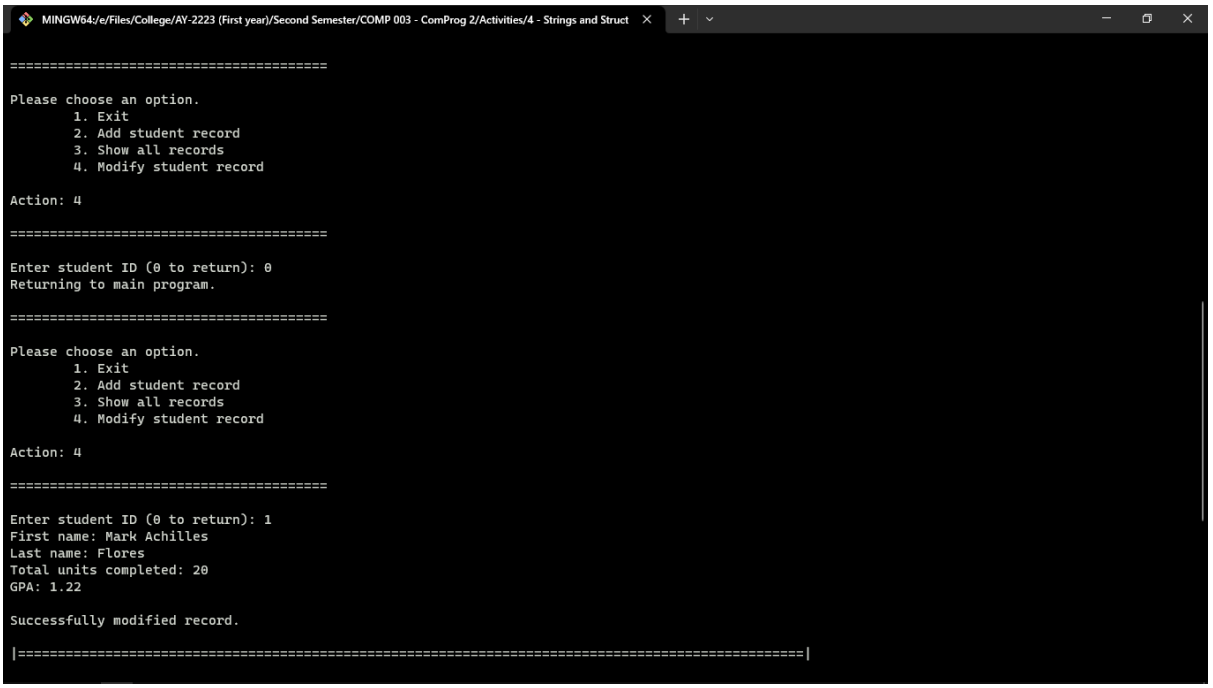
```c
#include <stdio.h>
#include <string.h>

typedef struct {
    int id;
    char first_name[30];
    char last_name[30];
    int total_units;
    float gpa;
} Student;

Student students[100];

int csc=0;

void show_records();
void add_student();
void modify_record();

int main () {
    int choice;

    printf("\nWelcome to the student records!\n");
    printf("=====================================\n");

    do {
        printf("Please choose an option.\n");
        printf("\t1. Exit\n");
        printf("\t2. Add student record\n");
        if (csc > 0) {
            printf("\t3. Show all records\n");
            printf("\t4. Modify student record\n");
        }
        printf("\n");
        printf("Action: ");
        scanf("%d", &choice);
        getchar();
        printf("\n=====================================\n");
        printf("\n");

        if (csc == 0) {
            switch (choice) {
                case 1:
                    printf("Exiting records!\n");
                    break;

                case 2:
                    if (csc == 50) {
                        printf("Maximum capacity for records have already been reached.\n");
                    }
                    add_student();

                    printf("\nAdded a student record!\n");
                    break;

                default:
                    printf("Invalid choice. Please try again.\n");
                    break;
            }
            printf("\n=====================================\n");
            printf("\n");
            continue;
        }

        switch (choice) {
            case 1:
                printf("Exiting records!\n");
                break;

            case 2:
                if (csc == 50) {
                    printf("Maximum capacity for records have already been reached.\n");
                }
                add_student();

                printf("\nAdded a student record!\n");
                break;

            case 3:
                show_records();
                break;

            case 4:
                modify_record();
                break;

            default:
                printf("Invalid choice. Please try again.\n");
                break;
        }
        printf("\n=====================================\n");
        printf("\n");

    } while (choice != 1);

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

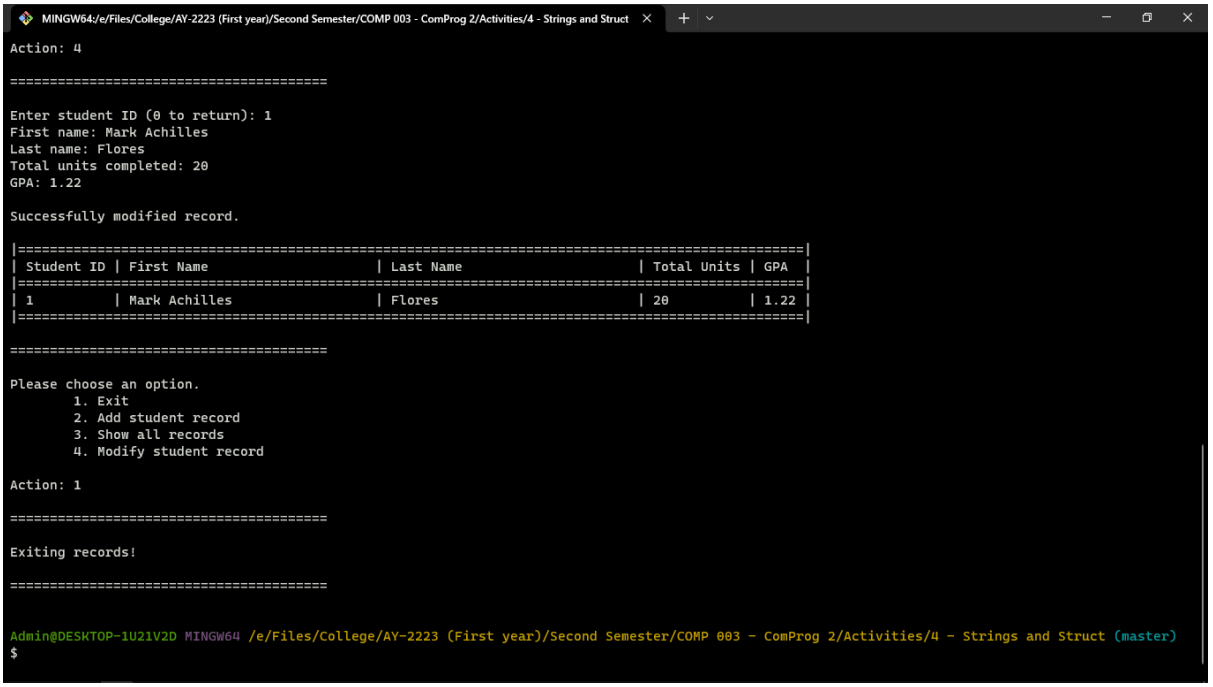
```c
void show_records () {

 printf("|=============================================================================================
=============|\n");
    printf("| Student ID | First Name                    | Last Name                     |
Total Units | GPA  |\n");

 printf("|=============================================================================================
=============|\n");

    for (int i=0; i<csc; i++) {
        printf("| %-3i        | %-30s | %-30s | %-2i          | %-.2f |\n", students[i].id,
students[i].first_name, students[i].last_name, students[i].total_units, students[i].gpa);

 printf("|=============================================================================================
=============|\n");
    }
}

void add_student () {
    Student temp;
    temp.id = csc + 1;
    printf("First name: ");
    fgets(temp.first_name, 30, stdin);
    strtok(temp.first_name, "\n");

    printf("Last name: ");
    fgets(temp.last_name, 30, stdin);
    strtok(temp.last_name, "\n");

    printf("Total units completed: ");
    scanf("%d", &temp.total_units);
    printf("GPA: ");
    scanf("%f", &temp.gpa);

    students[csc++] = temp;
}

void modify_record() {
    int id=0;

    do {
        printf("Enter student ID (0 to return): ");
        scanf("%d", &id);
        getchar();

        if (id > csc || id < 0) {
            printf("\nInvalid input. Please try again.\n\n");
        } else if (id == 0) {
            printf("Returning to main program.\n");
            return;
        }
    } while (id > csc || id < 0);

    Student temp;
    temp.id = id;

    printf("First name: ");
    fgets(temp.first_name, 100, stdin);
    strtok(temp.first_name, "\n");

    printf("Last name: ");
    fgets(temp.last_name, 100, stdin);
    strtok(temp.last_name, "\n");

    printf("Total units completed: ");
    scanf("%d", &temp.total_units);

    printf("GPA: ");
    scanf("%f", &temp.gpa);

    students[id-1] = temp;

    printf("\nSuccessfully modified record.\n\n");
    show_records();
}
```

Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4





Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4



Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

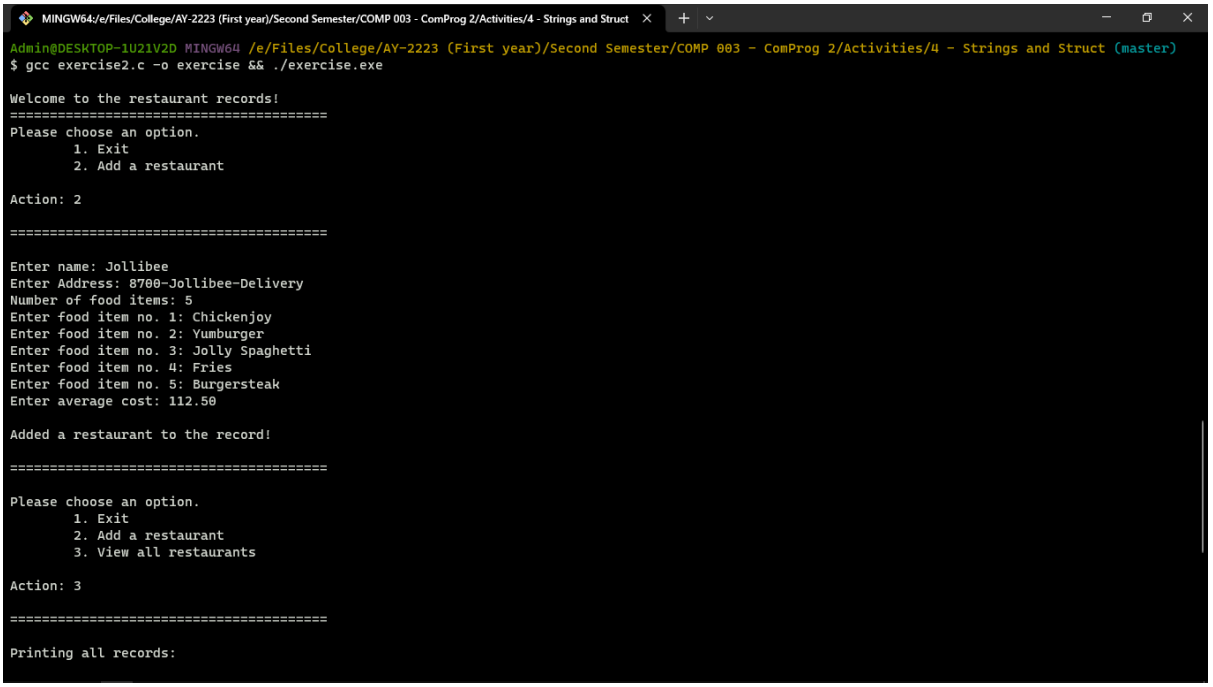
2. Create a structure that can describe a restaurant. It should have members that include the name, address, average cost, and type of food available.

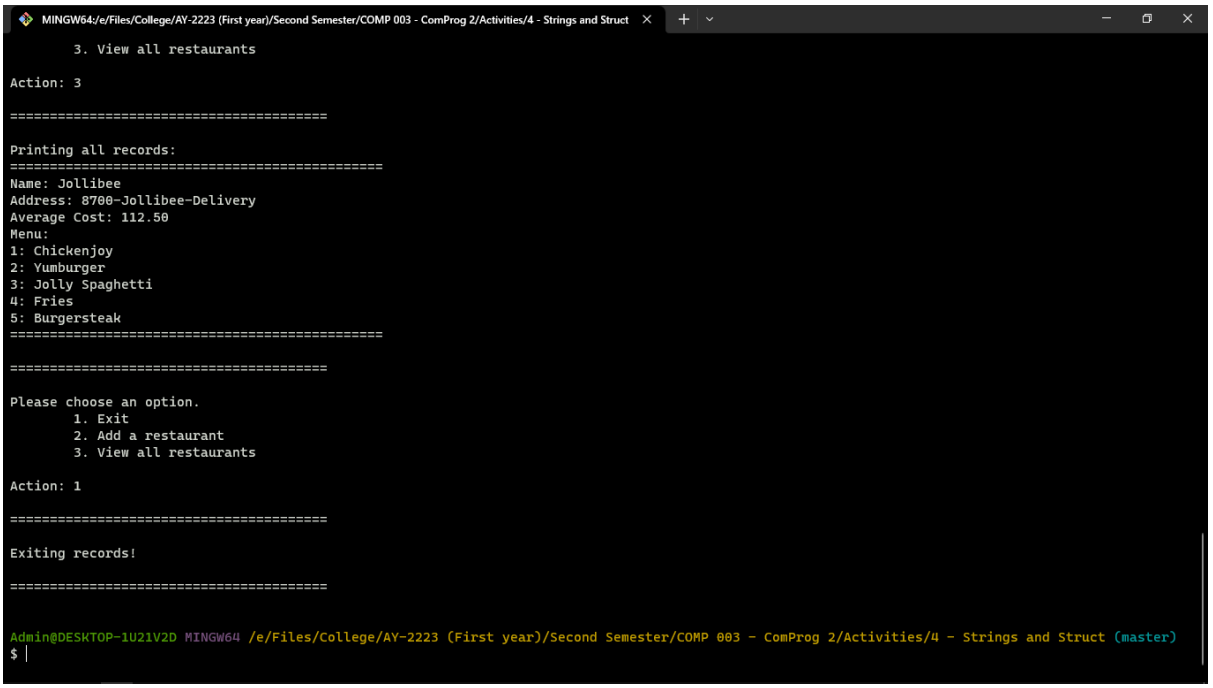
```c
#include <stdio.h>
#include <string.h>

typedef struct {
    char name[30], address[100], food_items[30][50];
    float avg_cost;
    int n_foods;
} Restaurant;

Restaurant restaurants[100];

int crc;

void add_restaurant();
void show_records();

int main (int argc, char *argv[]) {
    int choice;

    printf("\nWelcome to the restaurant records!\n");
    printf("=======================================\n");

    do {
        printf("Please choose an option.\n");
        printf("\t1. Exit\n");
        printf("\t2. Add a restaurant\n");
        if (crc > 0) {
            printf("\t3. View all restaurants\n");
        }
        printf("\n");
        printf("Action: ");
        scanf("%d", &choice);
        getchar();
        printf("\n=======================================\n");
        printf("\n");

        if (crc == 0) {
            switch (choice) {
                case 1:
                    printf("Exiting records!\n");
                    break;

                case 2:
                    if (crc == 100) {
                        printf("Maximum capacity for records have already been reached.\n");
                    }
                    add_restaurant();

                    printf("\nAdded a restaurant to the record!\n");
                    break;

                default:
                    printf("Invalid choice. Please try again.\n");
                    break;
            }
            printf("\n=======================================\n");
            printf("\n");
            continue;
        }
```

Name: Mark Achiles G. Flores Jr.
Year & Section: BSCS 1-4


```c
        switch (choice) {
            case 1:
                printf("Exiting records!\n");
                break;

            case 2:
                if (crc == 100) {
                    printf("Maximum capacity for records have already been reached.\n");
                }
                add_restaurant();

                printf("\nAdded a restaurant to the record!\n");
                break;

            case 3:
                show_records();
                break;

            default:
                printf("Invalid choice. Please try again.\n");
                break;
        }
        printf("\n=======================================\n");
        printf("\n");

    } while (choice != 1);

    return 0;
}

void add_restaurant() {
    Restaurant temp;

    printf("Enter name: ");
    fgets(temp.name, 30, stdin);
    strtok(temp.name, "\n");

    printf("Enter Address: ");
    fgets(temp.address, 100, stdin);
    strtok(temp.address, "\n");

    printf("Number of food items: ");
    scanf("%d", &temp.n_foods);
    getchar();

    for (int i=0; i<temp.n_foods; i++) {
        printf("Enter food item no. %d: ", i+1);
        fgets(temp.food_items[i], 50, stdin);
        strtok(temp.food_items[i], "\n");
    }

    printf("Enter average cost: ");
    scanf("%f", &temp.avg_cost);
    getchar();

    restaurants[crc] = temp;
    crc++;
}

void show_records () {
    printf("Printing all records:\n");
    printf("=============================================\n");

    for (int i=0; i<crc; i++) {
        printf("Name: %s\n", restaurants[i].name);
        printf("Address: %s\n", restaurants[i].address);
        printf("Average Cost: %.2f\n", restaurants[i]. avg_cost);
        printf("Menu:\n");
        for (int j=0; j<restaurants[i].n_foods; j++) {
            printf("%d: %s\n", j+1, restaurants[i].food_items[j]);
        }
        printf("=============================================\n");
    }
}
```

Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4





Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

3. Declare an array of structures with 12 elements. Each element is a structure with three fields. The first field shows the month in numeric form (1 to 12). The second field shows the name of the moth. The third field shows the number of days in the month.

```c
#include <stdio.h>

typedef struct {
    int month_num;
    char month_name[10];
    int n_days;
} Month;

int main (int argc, char *argv[]) {
    Month jan = {1, "January", 31};
    Month feb = {2, "February", 28};
    Month mar = {3, "March", 31};
    Month apr = {4, "April", 30};
    Month may = {5, "May", 31};
    Month jun = {6, "June", 30};
    Month jul = {7, "July", 31};
    Month aug = {8, "August", 31};
    Month sep = {9, "September", 30};
    Month oct = {10, "October", 31};
    Month nov = {11, "November", 30};
    Month dec = {12, "December", 31};

    Month months[] = {jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec};

    for (int i=0; i<12; i++) {
        printf("Month %02d: %s has %d days.\n", months[i].month_num, months[i].month_name, months[i].n_days);
    }

    return 0;
}
```

```
Month 01: January has 31 days.
Month 02: February has 28 days.
Month 03: March has 31 days.
Month 04: April has 30 days.
Month 05: May has 31 days.
Month 06: June has 30 days.
Month 07: July has 31 days.
Month 08: August has 31 days.
Month 09: September has 30 days.
Month 10: October has 31 days.
Month 11: November has 30 days.
Month 12: December has 31 days.
```

Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

4. Write a program in C to count total number of alphabets, digits, and special characters in a string.
Test Data
Input the string: Welcome to Programming 2.
Expected Output
Number of Alphabets in the string is: 20
Number of Digits in the string is: 1
Number of Special characters in the string is: 4

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

int main (int argc, char *argv[]) {
    char input[100];
    int alpha=0, num=0, spec=0, i=0;

    printf("Input the string: ");
    fgets(input, 100, stdin);
    strtok(input, "\n");

    for (i=0; i<strlen(input); i++) {
        if (isalpha(input[i])) {
            alpha++;
        } else if (isdigit(input[i])) {
            num++;
        } else {
            spec++;
        }
    }

    printf("Number of Alphabets in the string: %d\n", alpha);
    printf("Number of Digits in the string: %d\n", num);
    printf("Number of Special characters in the string: %d\n", spec);

    return 0;
}
```

```
Admin@DESKTOP-1U21V2D MINGW64 /e/Files/College/AY-
$ gcc exercise4.c -o exercise && ./exercise.exe
Input the string: Welcome to Programming 2.
Number of Alphabets in the string: 20
Number of Digits in the string: 1
Number of Special characters in the string: 4
```

Name: Mark Achiles G. Flores Jr. **Strings and Struct**
Year & Section: BSCS 1-4

5. Write a C program to sort a string array in ascending order.
Test Data
Input the string: programming
Expected Output
After sorting the string appears like: aggimmnoprr

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

int main (int argc, char *argv[]) {
    char input[100], clean[100];
    int counter=0, i, j;
    printf("Input the string: ");
    fgets(input, 100, stdin);

    // Remove spaces
    for (i=0; i<strlen(input); i++) {
        if (input[i] != ' ') {
            clean[counter] = input[i];
            counter++;
        }
    }

    strtok(clean, "\n");
    printf("Clean: %s\n", clean);

    for (i=0; i<strlen(clean)-1; i++) {
        for (j=0; j<strlen(clean)-i-1; j++) {
            if (clean[j] > clean[j+1]) {
                char temp = clean[j];
                clean[j] = clean[j+1];
                clean[j+1] = temp;
            }
        }
    }

    printf("Result: %s\n", clean);

    return 0;
}
```

```
Admin@DESKTOP-1U21V2D MINGW64 /e/Files/College/AY
$ gcc exercise5.c -o exercise && ./exercise.exe
Input the string: programming
Clean: programming
Result: aggimmnoprr

Admin@DESKTOP-1U21V2D MINGW64 /e/Files/College/AY
$ gcc exercise5.c -o exercise && ./exercise.exe
Input the string: Welcome to Programming 2.
Clean: WelcometoProgramming2.
Result: .2PWaceeggilmmmnooorrt
```