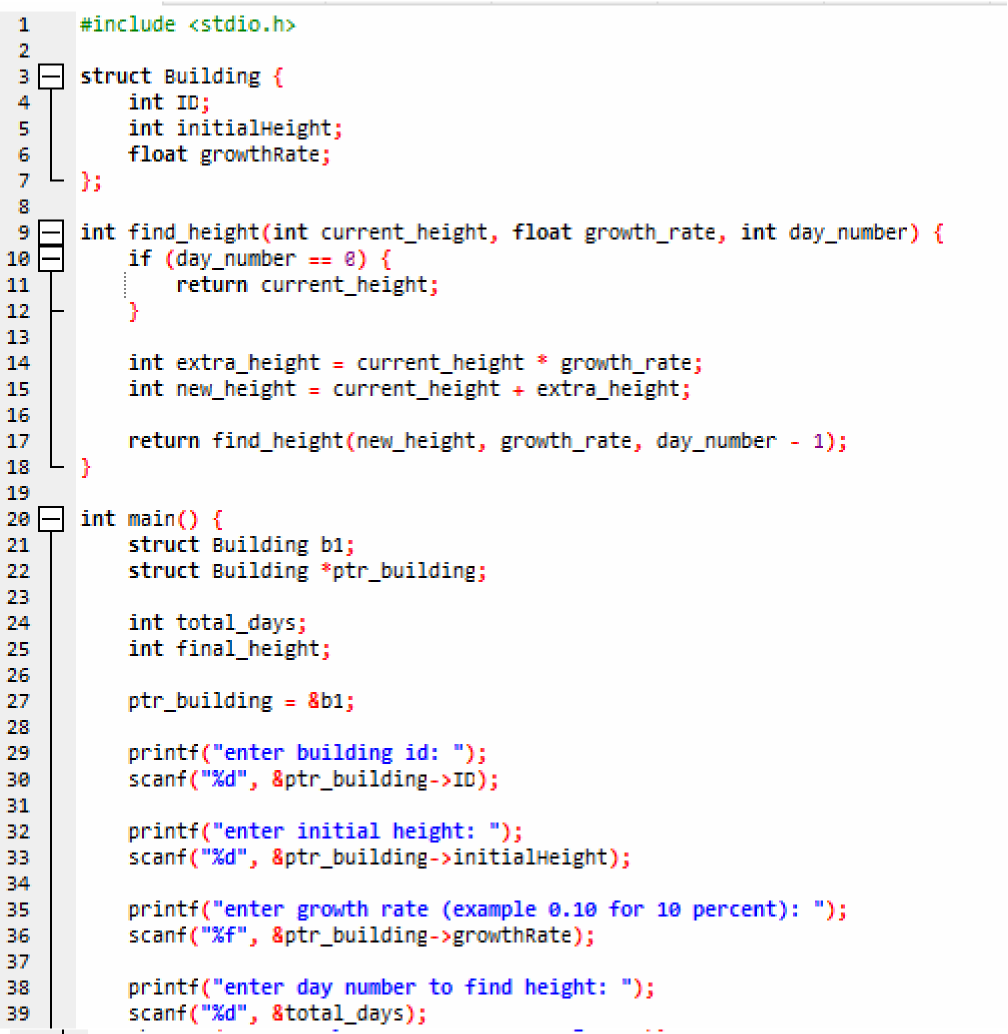
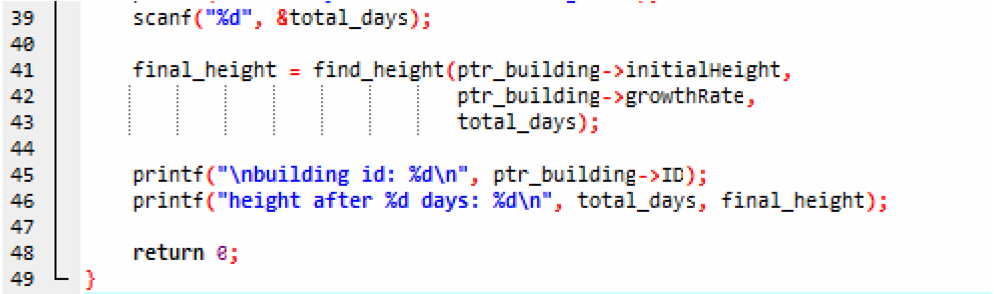
25K-2038 (LAB 11 TASKS)

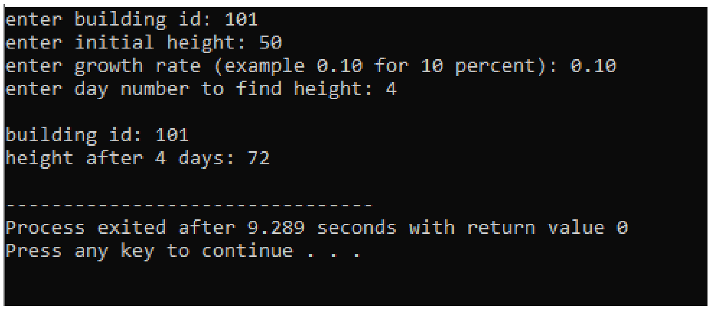
Question #1



```c
#include <stdio.h>

struct Building {
    int ID;
    int initialHeight;
    float growthRate;
};

int find_height(int current_height, float growth_rate, int day_number) {
    if (day_number == 0) {
        return current_height;
    }

    int extra_height = current_height * growth_rate;
    int new_height = current_height + extra_height;

    return find_height(new_height, growth_rate, day_number - 1);
}

int main() {
    struct Building b1;
    struct Building *ptr_building;

    int total_days;
    int final_height;

    ptr_building = &b1;

    printf("enter building id: ");
    scanf("%d", &ptr_building->ID);

    printf("enter initial height: ");
    scanf("%d", &ptr_building->initialHeight);

    printf("enter growth rate (example 0.10 for 10 percent): ");
    scanf("%f", &ptr_building->growthRate);

    printf("enter day number to find height: ");
    scanf("%d", &total_days);
```

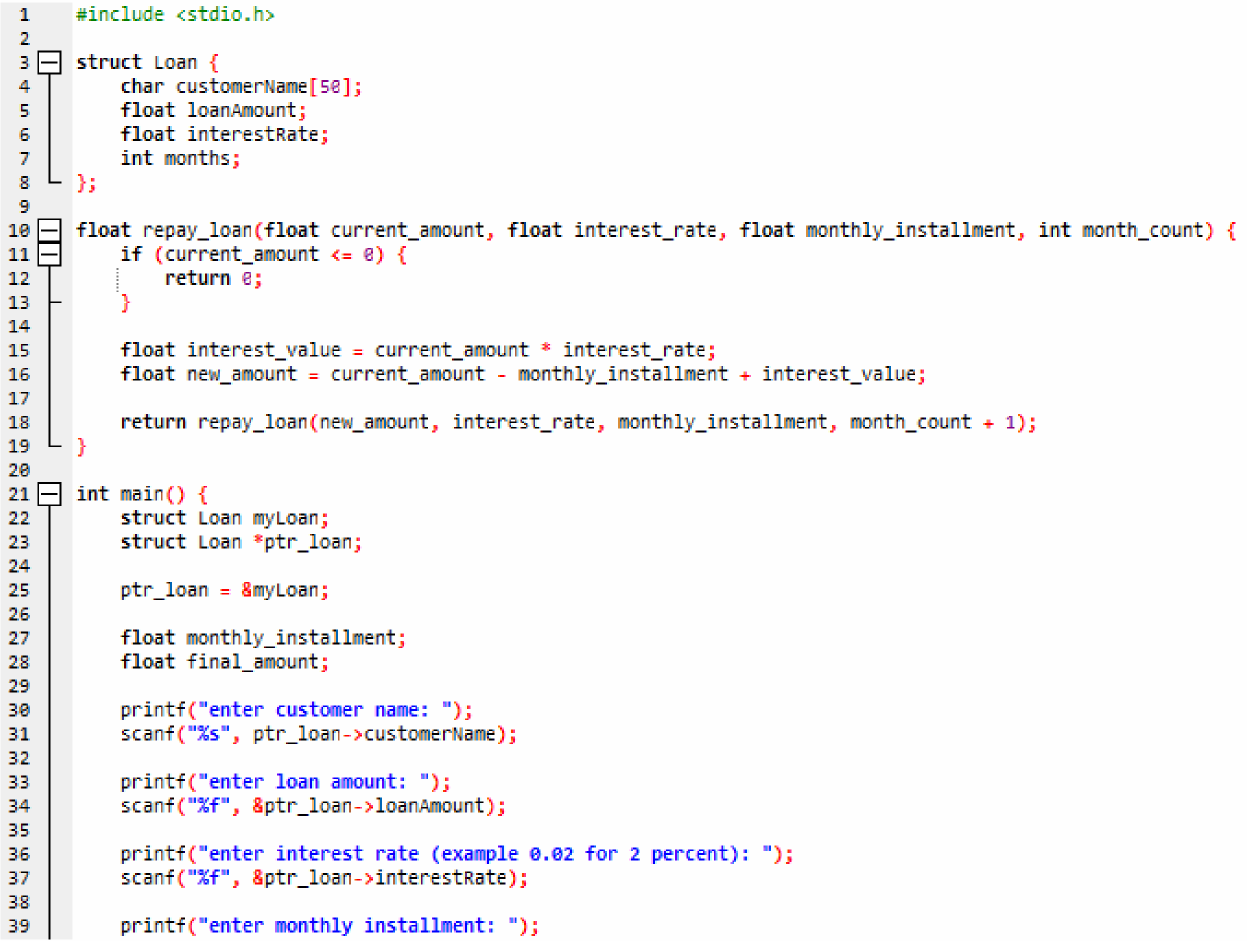


```c
    scanf("%d", &total_days);

    final_height = find_height(ptr_building->initialHeight,
                               ptr_building->growthRate,
                               total_days);

    printf("\nbuilding id: %d\n", ptr_building->ID);
    printf("height after %d days: %d\n", total_days, final_height);

    return 0;
}
```

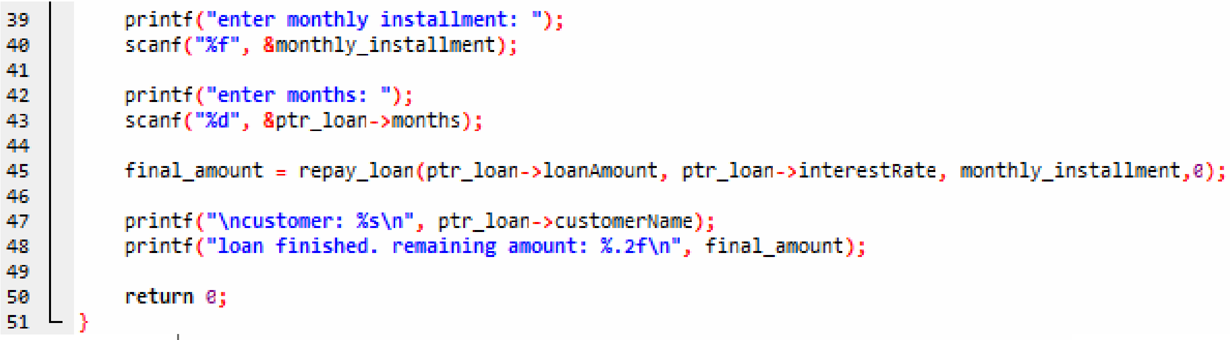


```
enter building id: 101
enter initial height: 50
enter growth rate (example 0.10 for 10 percent): 0.10
enter day number to find height: 4

building id: 101
height after 4 days: 72

--------------------------------
Process exited after 9.289 seconds with return value 0
Press any key to continue . . .
```

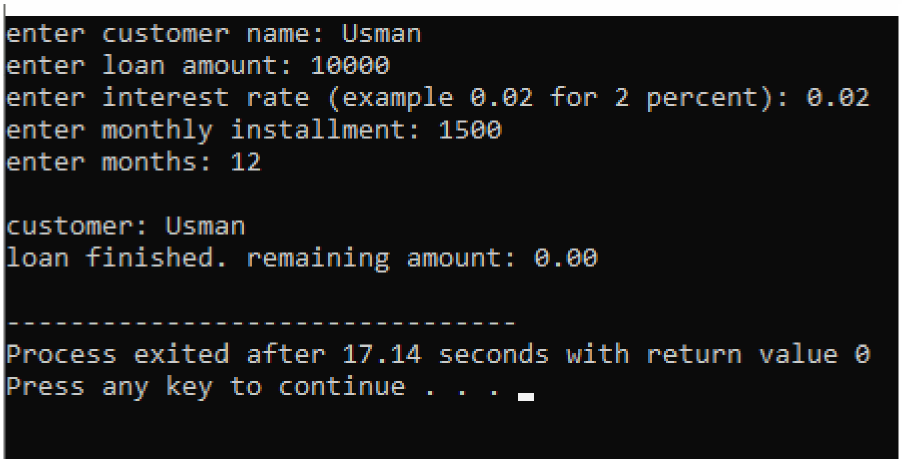
Question #2



```c
#include <stdio.h>

struct Loan {
    char customerName[50];
    float loanAmount;
    float interestRate;
    int months;
};

float repay_loan(float current_amount, float interest_rate, float monthly_installment, int month_count) {
    if (current_amount <= 0) {
        return 0;
    }

    float interest_value = current_amount * interest_rate;
    float new_amount = current_amount - monthly_installment + interest_value;

    return repay_loan(new_amount, interest_rate, monthly_installment, month_count + 1);
}

int main() {
    struct Loan myLoan;
    struct Loan *ptr_loan;

    ptr_loan = &myLoan;

    float monthly_installment;
    float final_amount;

    printf("enter customer name: ");
    scanf("%s", ptr_loan->customerName);

    printf("enter loan amount: ");
    scanf("%f", &ptr_loan->loanAmount);

    printf("enter interest rate (example 0.02 for 2 percent): ");
    scanf("%f", &ptr_loan->interestRate);

    printf("enter monthly installment: ");
```

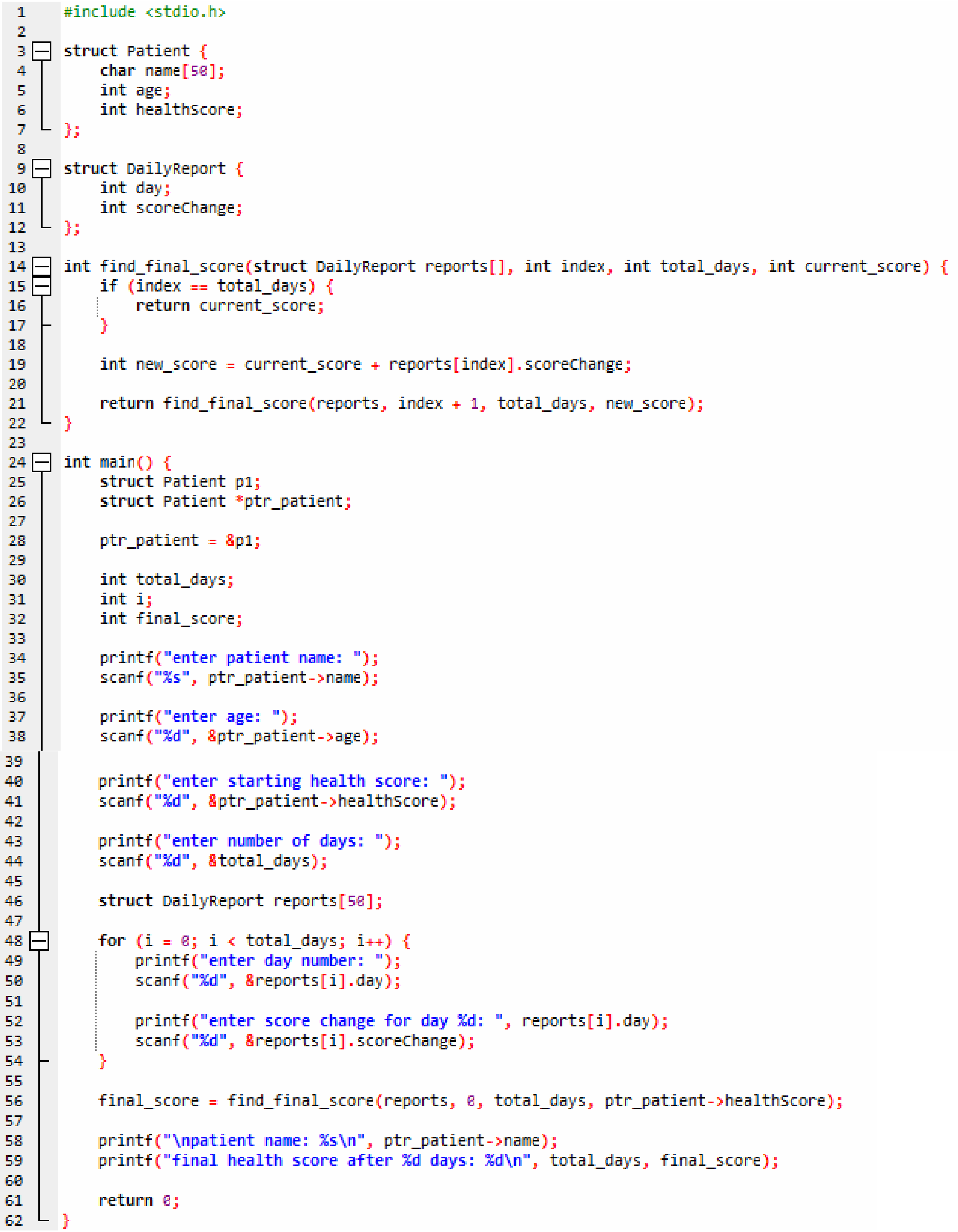


```c
    printf("enter monthly installment: ");
    scanf("%f", &monthly_installment);

    printf("enter months: ");
    scanf("%d", &ptr_loan->months);

    final_amount = repay_loan(ptr_loan->loanAmount, ptr_loan->interestRate, monthly_installment,0);

    printf("\ncustomer: %s\n", ptr_loan->customerName);
    printf("loan finished. remaining amount: %.2f\n", final_amount);

    return 0;
}
```

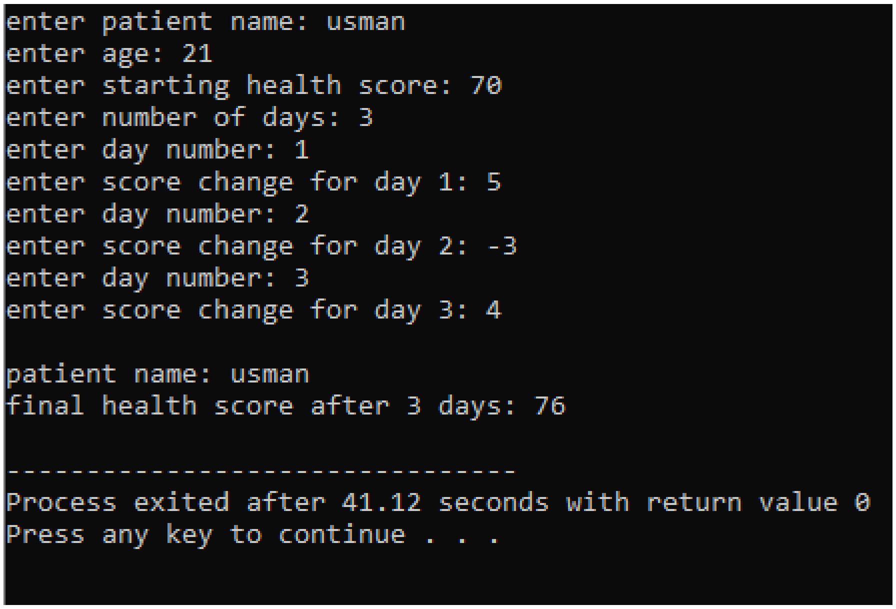


```
enter customer name: Usman
enter loan amount: 10000
enter interest rate (example 0.02 for 2 percent): 0.02
enter monthly installment: 1500
enter months: 12

customer: Usman
loan finished. remaining amount: 0.00

--------------------------------
Process exited after 17.14 seconds with return value 0
Press any key to continue . . .
```

Question #3

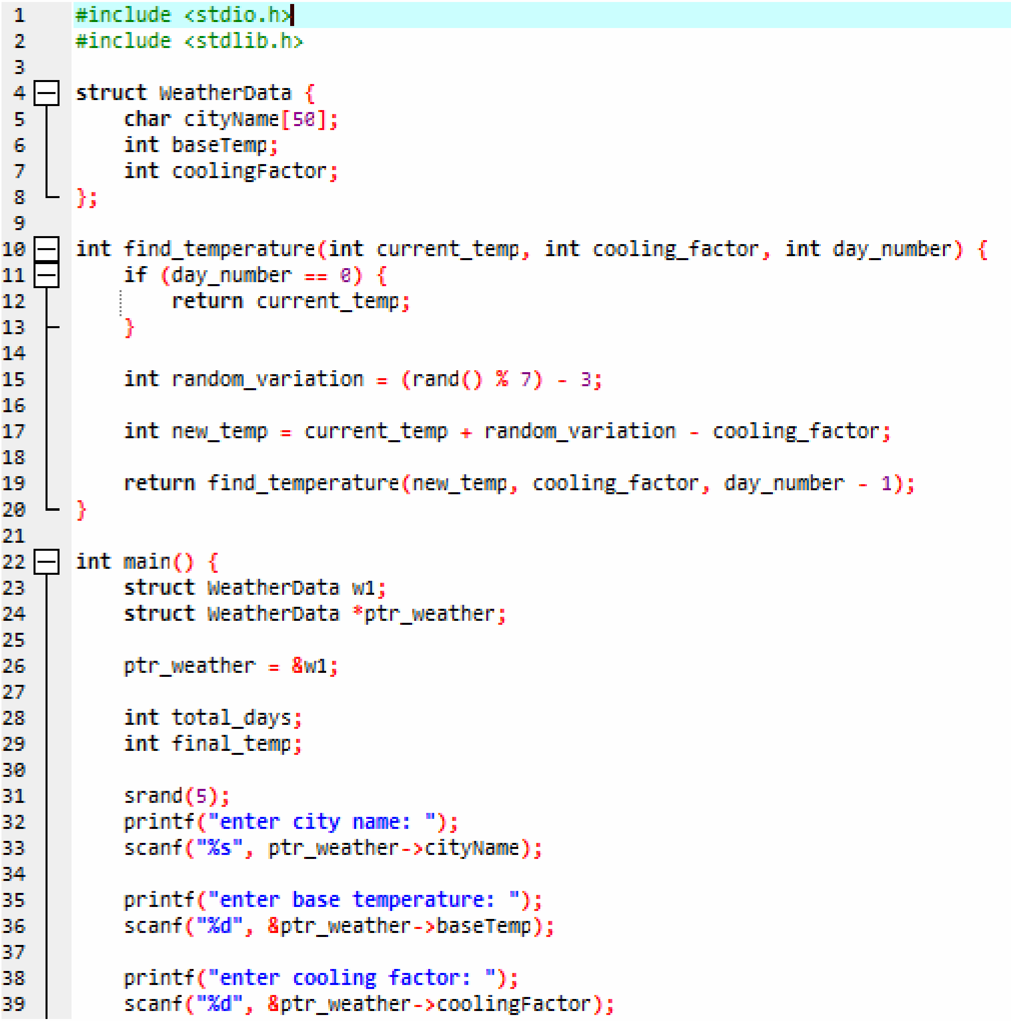


```c
#include <stdio.h>

struct Patient {
    char name[50];
    int age;
    int healthScore;
};

struct DailyReport {
    int day;
    int scoreChange;
};

int find_final_score(struct DailyReport reports[], int index, int total_days, int current_score) {
    if (index == total_days) {
        return current_score;
    }

    int new_score = current_score + reports[index].scoreChange;

    return find_final_score(reports, index + 1, total_days, new_score);
}

int main() {
    struct Patient p1;
    struct Patient *ptr_patient;

    ptr_patient = &p1;

    int total_days;
    int i;
    int final_score;

    printf("enter patient name: ");
    scanf("%s", ptr_patient->name);

    printf("enter age: ");
    scanf("%d", &ptr_patient->age);

    printf("enter starting health score: ");
    scanf("%d", &ptr_patient->healthScore);

    printf("enter number of days: ");
    scanf("%d", &total_days);

    struct DailyReport reports[50];

    for (i = 0; i < total_days; i++) {
        printf("enter day number: ");
        scanf("%d", &reports[i].day);

        printf("enter score change for day %d: ", reports[i].day);
        scanf("%d", &reports[i].scoreChange);
    }

    final_score = find_final_score(reports, 0, total_days, ptr_patient->healthScore);

    printf("\npatient name: %s\n", ptr_patient->name);
    printf("final health score after %d days: %d\n", total_days, final_score);

    return 0;
}
```

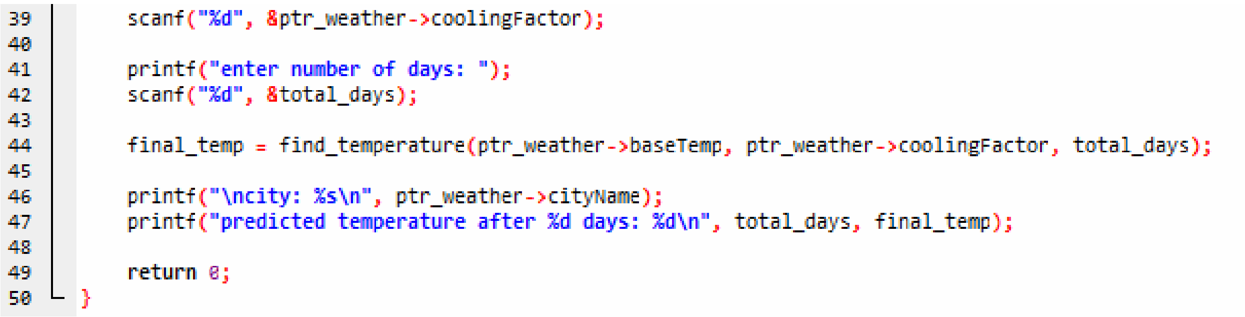


```
enter patient name: usman
enter age: 21
enter starting health score: 70
enter number of days: 3
enter day number: 1
enter score change for day 1: 5
enter day number: 2
enter score change for day 2: -3
enter day number: 3
enter score change for day 3: 4

patient name: usman
final health score after 3 days: 76

--------------------------------
Process exited after 41.12 seconds with return value 0
Press any key to continue . . .
```

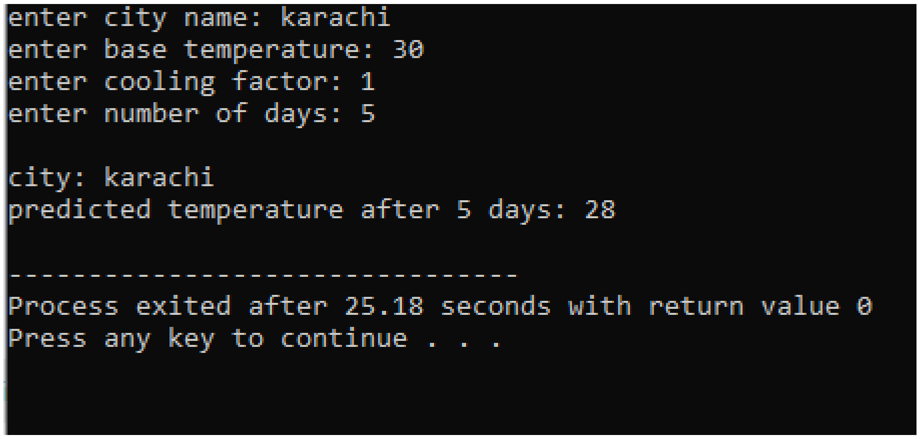
Question #4



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

struct WeatherData {
    char cityName[50];
    int baseTemp;
    int coolingFactor;
};

int find_temperature(int current_temp, int cooling_factor, int day_number) {
    if (day_number == 0) {
        return current_temp;
    }

    int random_variation = (rand() % 7) - 3;

    int new_temp = current_temp + random_variation - cooling_factor;

    return find_temperature(new_temp, cooling_factor, day_number - 1);
}

int main() {
    struct WeatherData w1;
    struct WeatherData *ptr_weather;

    ptr_weather = &w1;

    int total_days;
    int final_temp;

    srand(5);
    printf("enter city name: ");
    scanf("%s", ptr_weather->cityName);

    printf("enter base temperature: ");
    scanf("%d", &ptr_weather->baseTemp);

    printf("enter cooling factor: ");
    scanf("%d", &ptr_weather->coolingFactor);
```

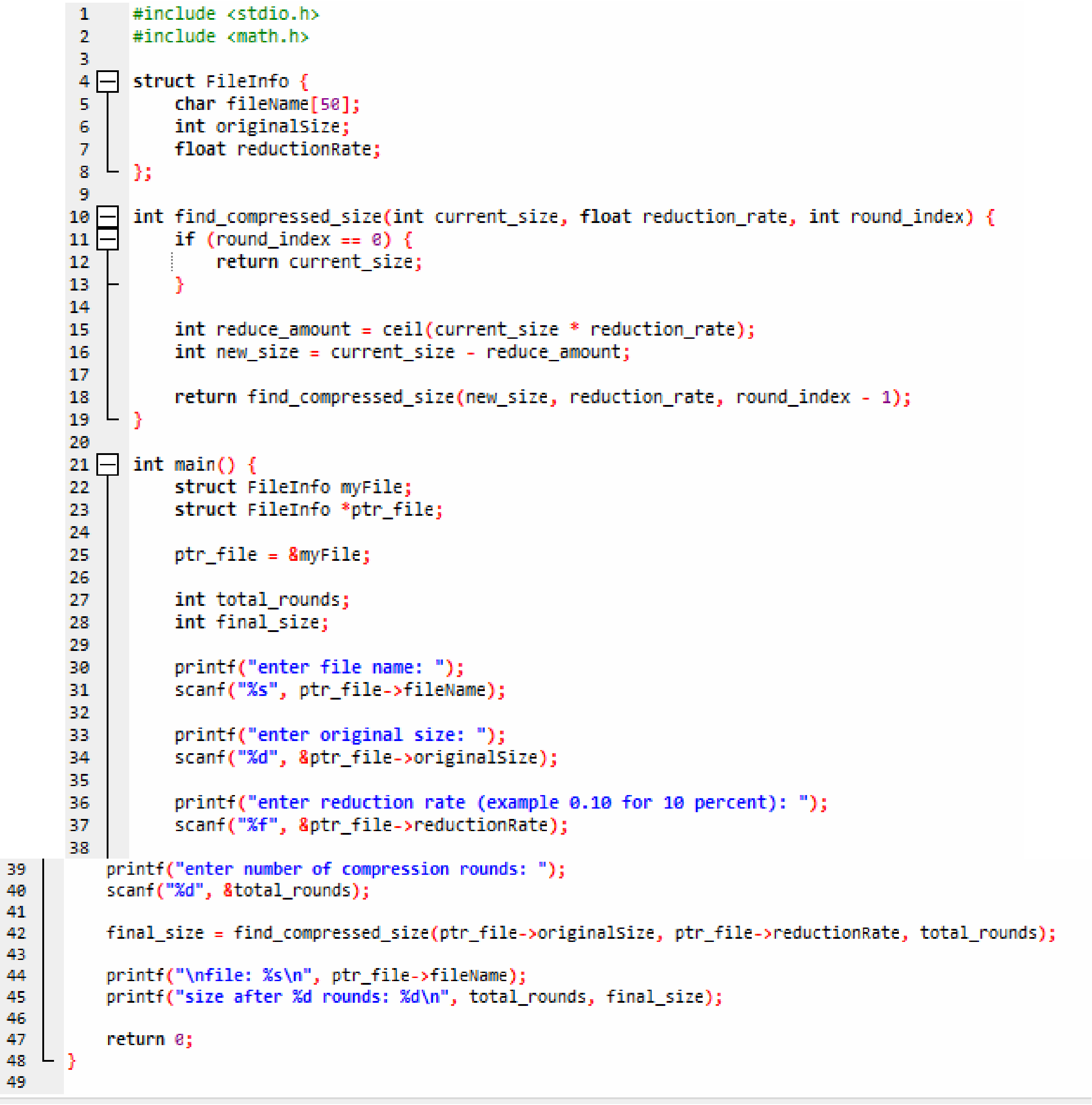


```c
    scanf("%d", &ptr_weather->coolingFactor);

    printf("enter number of days: ");
    scanf("%d", &total_days);

    final_temp = find_temperature(ptr_weather->baseTemp, ptr_weather->coolingFactor, total_days);

    printf("\ncity: %s\n", ptr_weather->cityName);
    printf("predicted temperature after %d days: %d\n", total_days, final_temp);

    return 0;
}
```

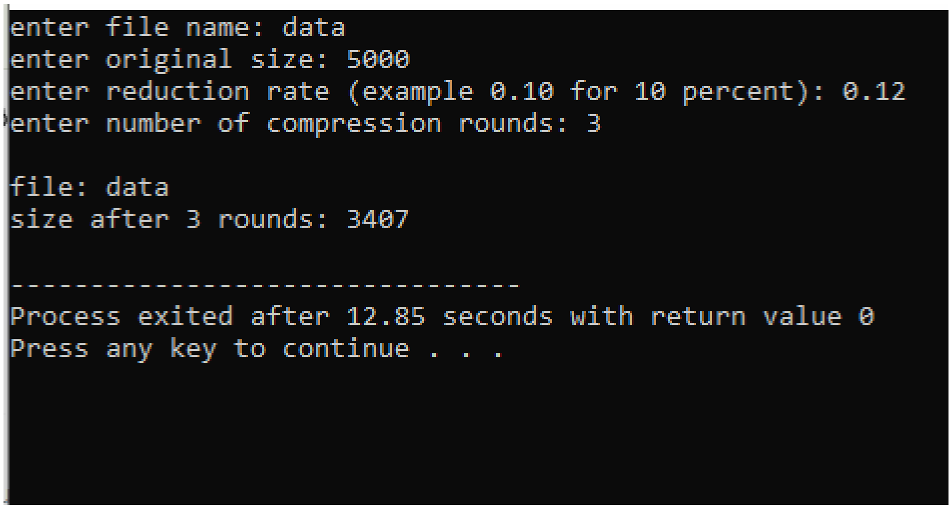


```
enter city name: karachi
enter base temperature: 30
enter cooling factor: 1
enter number of days: 5

city: karachi
predicted temperature after 5 days: 28

--------------------------------
Process exited after 25.18 seconds with return value 0
Press any key to continue . . .
```

Question #5

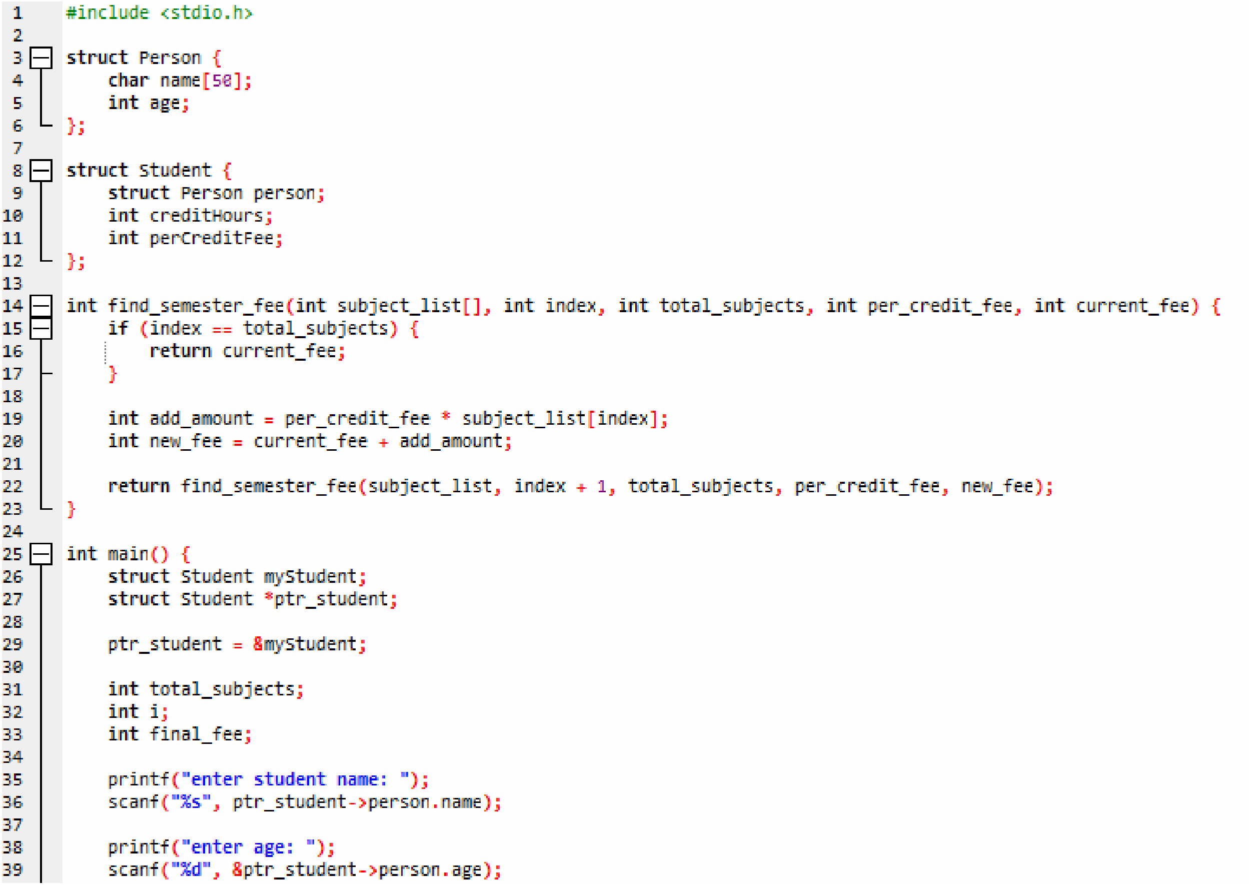


```c
#include <stdio.h>
#include <math.h>

struct FileInfo {
    char fileName[50];
    int originalSize;
    float reductionRate;
};

int find_compressed_size(int current_size, float reduction_rate, int round_index) {
    if (round_index == 0) {
        return current_size;
    }

    int reduce_amount = ceil(current_size * reduction_rate);
    int new_size = current_size - reduce_amount;

    return find_compressed_size(new_size, reduction_rate, round_index - 1);
}

int main() {
    struct FileInfo myFile;
    struct FileInfo *ptr_file;

    ptr_file = &myFile;

    int total_rounds;
    int final_size;

    printf("enter file name: ");
    scanf("%s", ptr_file->fileName);

    printf("enter original size: ");
    scanf("%d", &ptr_file->originalSize);

    printf("enter reduction rate (example 0.10 for 10 percent): ");
    scanf("%f", &ptr_file->reductionRate);

    printf("enter number of compression rounds: ");
    scanf("%d", &total_rounds);

    final_size = find_compressed_size(ptr_file->originalSize, ptr_file->reductionRate, total_rounds);

    printf("\nfile: %s\n", ptr_file->fileName);
    printf("size after %d rounds: %d\n", total_rounds, final_size);

    return 0;
}
```

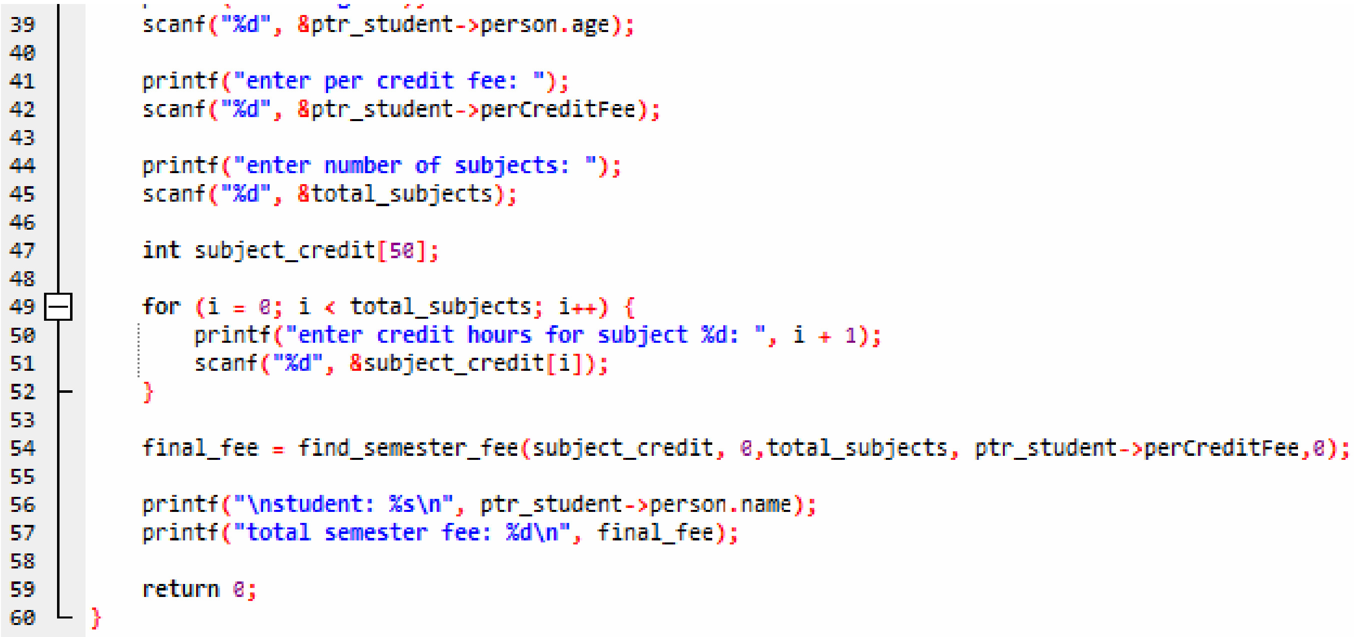


```
enter file name: data
enter original size: 5000
enter reduction rate (example 0.10 for 10 percent): 0.12
enter number of compression rounds: 3

file: data
size after 3 rounds: 3407

--------------------------------
Process exited after 12.85 seconds with return value 0
Press any key to continue . . .
```

Question #6

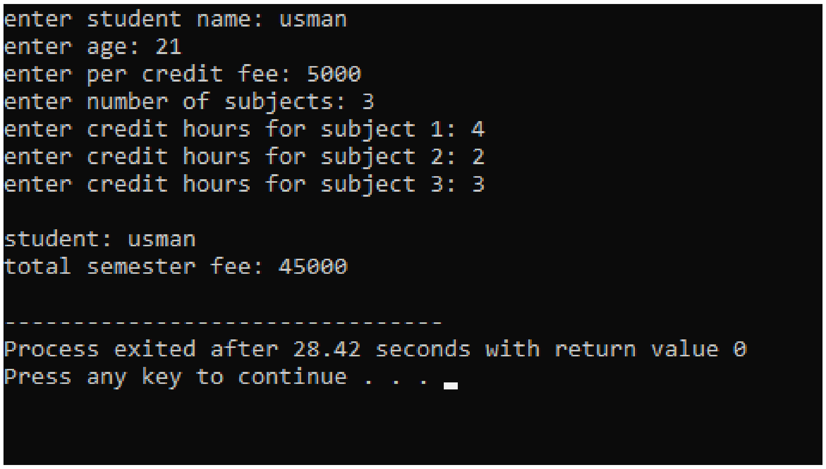


```c
#include <stdio.h>

struct Person {
    char name[50];
    int age;
};

struct Student {
    struct Person person;
    int creditHours;
    int perCreditFee;
};

int find_semester_fee(int subject_list[], int index, int total_subjects, int per_credit_fee, int current_fee) {
    if (index == total_subjects) {
        return current_fee;
    }

    int add_amount = per_credit_fee * subject_list[index];
    int new_fee = current_fee + add_amount;

    return find_semester_fee(subject_list, index + 1, total_subjects, per_credit_fee, new_fee);
}

int main() {
    struct Student myStudent;
    struct Student *ptr_student;

    ptr_student = &myStudent;

    int total_subjects;
    int i;
    int final_fee;

    printf("enter student name: ");
    scanf("%s", ptr_student->person.name);

    printf("enter age: ");
    scanf("%d", &ptr_student->person.age);
```



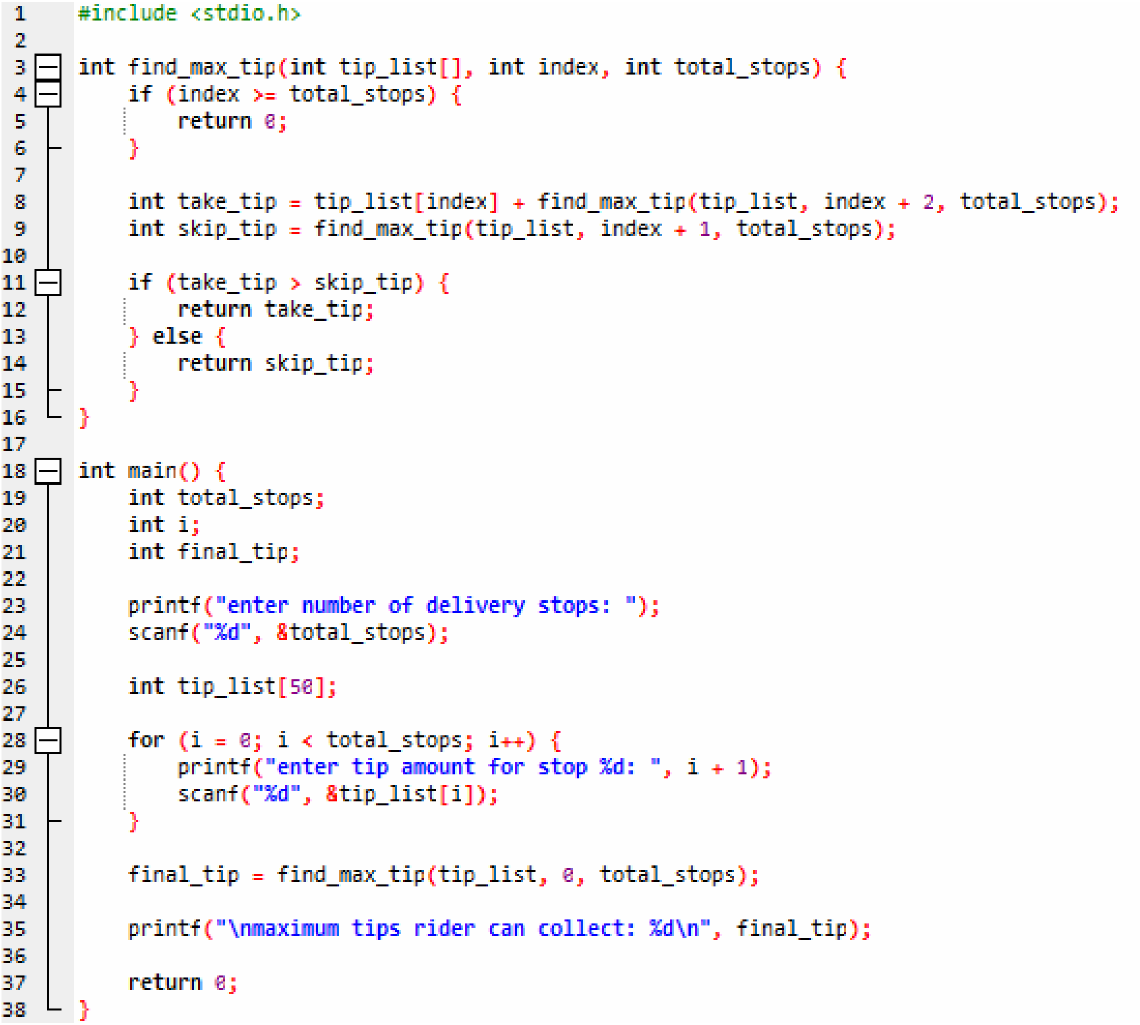
    scanf("%d", &ptr_student->person.age);

```c

    printf("enter per credit fee: ");
    scanf("%d", &ptr_student->perCreditFee);

    printf("enter number of subjects: ");
    scanf("%d", &total_subjects);

    int subject_credit[50];

    for (i = 0; i < total_subjects; i++) {
        printf("enter credit hours for subject %d: ", i + 1);
        scanf("%d", &subject_credit[i]);
    }

    final_fee = find_semester_fee(subject_credit, 0,total_subjects, ptr_student->perCreditFee,0);

    printf("\nstudent: %s\n", ptr_student->person.name);
    printf("total semester fee: %d\n", final_fee);

    return 0;
}
```

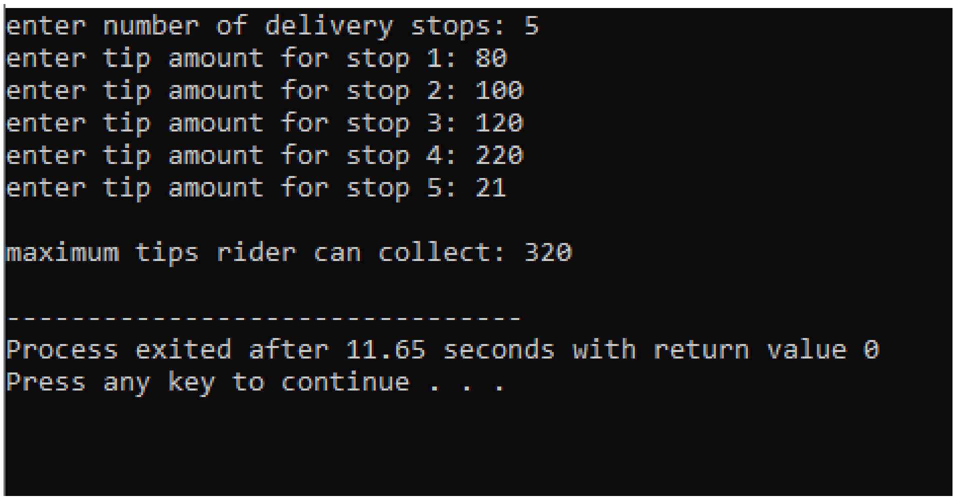


```
enter student name: usman
enter age: 21
enter per credit fee: 5000
enter number of subjects: 3
enter credit hours for subject 1: 4
enter credit hours for subject 2: 2
enter credit hours for subject 3: 3

student: usman
total semester fee: 45000

--------------------------------
Process exited after 28.42 seconds with return value 0
Press any key to continue . . .
```

Question #7



```c
#include <stdio.h>

int find_max_tip(int tip_list[], int index, int total_stops) {
    if (index >= total_stops) {
        return 0;
    }

    int take_tip = tip_list[index] + find_max_tip(tip_list, index + 2, total_stops);
    int skip_tip = find_max_tip(tip_list, index + 1, total_stops);

    if (take_tip > skip_tip) {
        return take_tip;
    } else {
        return skip_tip;
    }
}

int main() {
    int total_stops;
    int i;
    int final_tip;

    printf("enter number of delivery stops: ");
    scanf("%d", &total_stops);

    int tip_list[50];

    for (i = 0; i < total_stops; i++) {
        printf("enter tip amount for stop %d: ", i + 1);
        scanf("%d", &tip_list[i]);
    }

    final_tip = find_max_tip(tip_list, 0, total_stops);

    printf("\nmaximum tips rider can collect: %d\n", final_tip);

    return 0;
}
```



```
enter number of delivery stops: 5
enter tip amount for stop 1: 80
enter tip amount for stop 2: 100
enter tip amount for stop 3: 120
enter tip amount for stop 4: 220
enter tip amount for stop 5: 21

maximum tips rider can collect: 320

--------------------------------
Process exited after 11.65 seconds with return value 0
Press any key to continue . . .
```