Problem 1: Rainy Day Decision

Problem Statement:

You are planning an outing with your friends, but you are unsure whether to go out or stay indoors. You have access to weather information that tells you if it is raining or not. You will only go out if it is not raining.

Input: A single word, either "rainy" or "clear".

Output: A single line containing either "Go out" or "Stay indoors".

Solution:

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

int main() {
    char weather[10];
    scanf("%s", weather);

    if (strcmp(weather, "clear") == 0) {
        printf("Go out\n");
    } else {
        printf("Stay indoors\n");
    }
    return 0;
}
```

Explanation:

- We include the necessary headers: `<stdio.h>` for input/output functions and `<string.h>` for string comparison.
- We declare a string variable `weather` to hold the input.
- We read the weather condition using `scanf`.

- We use `strcmp` to compare the input with "clear". If they are the same, we print "Go out". Otherwise, we print "Stay indoors".

Problem 2: Eligibility to Vote

Problem Statement:

In many countries, the minimum age for voting is 18. You are tasked with writing a program that checks if a person is eligible to vote based on their age.

Input: A single integer representing the age of the person.

Output: A single line containing either "Eligible to vote" or "Not eligible to vote".

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int age;
    scanf("%d", &age);

    if (age >= 18) {
        printf("Eligible to vote\n");
    } else {
        printf("Not eligible to vote\n");
    }
    return 0;
}
```

Explanation:

- We read the person's age using `scanf`.
- We check if the age is 18 or older. If true, we print "Eligible to vote"; otherwise, we print "Not eligible to vote".

Problem 3: Electricity Bill Calculation

Problem Statement:

You are tasked with calculating the electricity bill for a household. The billing rates are as follows:

- For the first 100 units, the rate is 5 per unit.
- For the next 100 units (101 to 200), the rate is 7 per unit.
- For units above 200, the rate is 10 per unit.

Input: A single integer representing the number of units consumed.
Output: A single integer representing the total bill amount.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int units;
    scanf("%d", &units);
    int bill = 0;

    if (units <= 100) {
        bill = units * 5;
    } else if (units <= 200) {
        bill = 100 * 5 + (units - 100) * 7;
    } else {
        bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
    }

    printf("%d\n", bill);
    return 0;
}
```

Explanation:

- We read the number of units consumed using `scanf`.

- We calculate the bill based on the number of units using conditional statements:
    - If the units are 100 or fewer, we multiply by 5.
    - If the units are between 101 and 200, we calculate the bill for the first 100 units and add the cost for the additional units at the rate of 7.
    - If the units are above 200, we calculate the total cost by adding the costs for all three ranges.

Problem 4: Grade Classification

Problem Statement:

Students in a school are graded based on their marks. The grading system is as follows:

- 90 to 100 marks: Grade A
- 80 to 89 marks: Grade B
- 70 to 79 marks: Grade C
- 60 to 69 marks: Grade D
- Below 60 marks: Grade F

Input: A single integer representing the student's marks (between 0 and 100).
Output: A single character representing the student's grade (A, B, C, D, or F).

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int marks;
    scanf("%d", &marks);

    if (marks >= 90) {
        printf("A\n");
    } else if (marks >= 80) {
        printf("B\n");
    } else if (marks >= 70) {
        printf("C\n");
    } else if (marks >= 60) {
```

```
        printf("D\n");
    } else {
        printf("F\n");
    }
    return 0;
}
```

Explanation:

- We read the student's marks using `scanf`.
- We use a series of conditional statements to check the range of marks and print the corresponding grade.

Problem 5: Parking Fee

Problem Statement:

A parking lot charges fees based on the number of hours a car is parked. The charges are as follows:

- 1 to 3 hours: 10 units per hour.
- 4 to 6 hours: 8 units per hour.
- More than 6 hours: 5 units per hour.

Input: A single integer representing the number of hours the car was parked.
Output: A single integer representing the total parking fee.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int hours;
    scanf("%d", &hours);
    int fee = 0;

    if (hours <= 3) {
```

```
        fee = hours * 10;
    } else if (hours <= 6) {
        fee = 3 * 10 + (hours - 3) * 8;
    } else {
        fee = 3 * 10 + 3 * 8 + (hours - 6) * 5;
    }

    printf("%d\n", fee);
    return 0;
}
```

Explanation:

- We read the number of hours the car was parked.
- We calculate the fee using conditional statements:
    - For the first 3 hours, we charge 10 units per hour.
    - For hours between 4 and 6, we charge 10 for the first 3 hours and 8 for the additional hours.
    - For hours above 6, we calculate the fee for all hours accordingly.

Problem 6: BMI Calculator

Problem Statement:

The Body Mass Index (BMI) is calculated using the formula:

*BMI=weight (kg)height (m)2|text{BMI} = |frac{|text{weight (kg)}}{|text{height (m)}^2}*BMI=height (m)2weight (kg)

Based on the BMI value, the following classifications apply:

- BMI < 18.5: Underweight
- 18.5 ≤ BMI < 24.9: Normal weight
- 25.0 ≤ BMI < 29.9: Overweight
- BMI ≥ 30.0: Obese

Input: Two integers: weight in kilograms and height in centimeters.
Output: A single line containing the classification ("Underweight", "Normal weight", "Overweight", or "Obese").

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int weight, height;
    scanf("%d %d", &weight, &height);

    float height_m = height / 100.0; // Convert height to meters
    float bmi = weight / (height_m * height_m);

    if (bmi < 18.5) {
        printf("Underweight\n");
    } else if (bmi < 24.9) {
        printf("Normal weight\n");
    } else if (bmi < 29.9) {
        printf("Overweight\n");
    } else {
        printf("Obese\n");
    }
    return 0;
}
```

Explanation:

- We read the weight and height from input.
- We convert height from centimeters to meters.
- We calculate BMI using the formula provided.
- We classify the BMI using conditional statements and print the corresponding category.

Problem 7: Speed Limit Check

Problem Statement:

A driver must follow speed limits while driving. The following penalties apply if the driver exceeds the speed limit:

- Speed ≤ 60: "No fine"
- 61 ≤ Speed ≤ 80: "Small fine"
- Speed > 80: "Heavy fine"

Input: A single integer representing the speed of the vehicle.
Output: A single line containing "No fine", "Small fine", or "Heavy fine".

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int speed;
    scanf("%d", &speed);

    if (speed <= 60) {
        printf("No fine\n");
    } else if (speed <= 80) {
        printf("Small fine\n");
    } else {
        printf("Heavy fine\n");
    }
    return 0;
}
```

Explanation:

- We read the speed from input.
- We use conditional statements to check the speed range and print the corresponding penalty.

Problem 8: Bus Fare Calculation

Problem Statement:

A bus company charges fares based on the distance traveled:

- For distances up to 10 km, the fare is 20 units.
- For distances between 11 km and 50 km, the fare is 50 units.
- For distances above 50 km, the fare is 100 units.

Input: A single integer representing the distance traveled in kilometers.
Output: A single integer representing the total fare.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int distance;
    scanf("%d", &distance);

    int fare = 0;
    if (distance <= 10) {
        fare = 20;
    } else if (distance <= 50) {
        fare = 50;
    } else {
        fare = 100;
    }

    printf("%d\n", fare);
    return 0;
}
```

Explanation:

- We read the distance traveled.
- We use conditional statements to determine the fare based on the distance and print the corresponding fare.

Problem 9: Restaurant Discount

Problem Statement:

A restaurant offers discounts based on the total bill amount:

- If the bill is greater than 1000 units, a 10% discount is applied.
- If the bill is between 500 and 1000 units, a 5% discount is applied.
- No discount is applied if the bill is less than 500 units.

Input: A single integer representing the total bill amount.
Output: A single integer representing the final amount after applying any discount.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int bill;
    scanf("%d", &bill);

    if (bill > 1000) {
        bill *= 0.9; // 10% discount
    } else if (bill >= 500) {
        bill *= 0.95; // 5% discount
    }

    printf("%d\n", bill);
    return 0;
}
```

Explanation:

- We read the total bill amount.
- We check the bill amount using conditional statements and apply the corresponding discount by adjusting the bill amount.
- Finally, we print the final amount after applying the discount.

Problem 10: Fuel Efficiency

Problem Statement:

A car's fuel efficiency is measured in kilometers per liter (km/L). The car is classified based on its efficiency as follows:

- Efficiency < 10 km/L: "Poor"
- 10 ≤ Efficiency < 15 km/L: "Average"
- 15 ≤ Efficiency < 20 km/L: "Good"
- Efficiency ≥ 20 km/L: "Excellent"

Input: Two integers: the total distance traveled (in kilometers) and the fuel consumed (in liters).
Output: A single line containing the classification ("Poor", "Average", "Good", or "Excellent").

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int distance, fuel;
    scanf("%d %d", &distance, &fuel);

    float efficiency = (float)distance / fuel;

    if (efficiency < 10) {
        printf("Poor\n");
    } else if (efficiency < 15) {
        printf("Average\n");
    } else if (efficiency < 20) {
        printf("Good\n");
    } else {
        printf("Excellent\n");
    }
    return 0;
```

```
}
```

Explanation:

- We read the total distance and fuel consumed.
- We calculate the fuel efficiency by dividing the distance by the fuel consumed.
- We classify the efficiency using conditional statements and print the corresponding classification.

Problem 11: Temperature Classification

Problem Statement:

A weather station classifies the temperature as follows:

- Below 0°C: "Freezing"
- 0°C to 10°C: "Cold"
- 11°C to 20°C: "Cool"
- 21°C to 30°C: "Warm"
- Above 30°C: "Hot"

Input: A single integer representing the temperature in degrees Celsius.
Output: A single line containing one of the classifications ("Freezing", "Cold", "Cool", "Warm", or "Hot").

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int temperature;
    scanf("%d", &temperature);

    if (temperature < 0) {
        printf("Freezing\n");
    } else if (temperature <= 10) {
        printf("Cold\n");
```

```
    } else if (temperature <= 20) {
        printf("Cool\n");
    } else if (temperature <= 30) {
        printf("Warm\n");
    } else {
        printf("Hot\n");
    }
    return 0;
}
```

Explanation:

- We read the temperature from input.
- We classify the temperature using conditional statements and print the corresponding classification.

Problem 12: Water Usage Bill

Problem Statement:

A household's water bill is calculated based on the amount of water used:

- Up to 30 cubic meters: 15 units per cubic meter.
- Between 31 and 100 cubic meters: 10 units per cubic meter.
- Above 100 cubic meters: 5 units per cubic meter.

Input: A single integer representing the amount of water used in cubic meters.
Output: A single integer representing the total bill.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int usage;
    scanf("%d", &usage);
    int bill = 0;
```

```
    if (usage <= 30) {
        bill = usage * 15;
    } else if (usage <= 100) {
        bill = 30 * 15 + (usage - 30) * 10;
    } else {
        bill = 30 * 15 + 70 * 10 + (usage - 100) * 5;
    }

    printf("%d\n", bill);
    return 0;
}
```

Explanation:

- We read the amount of water used.
- We calculate the bill using conditional statements based on the usage range and print the total bill.

Problem 13: Store Discount

Problem Statement:

A store offers the following discounts:

- If the customer buys more than 10 items, they receive a 20% discount.
- If the customer buys between 5 and 10 items, they receive a 10% discount.
- No discount for fewer than 5 items.

Input: Two integers: the number of items purchased and the price per item.
Output: A single integer representing the final price after applying the appropriate discount.

Solution:

```
c
Copy code
#include <stdio.h>
```

```
int main() {
    int items, price_per_item;
    scanf("%d %d", &items, &price_per_item);

    int total_price = items * price_per_item;

    if (items > 10) {
        total_price *= 0.8; // 20% discount
    } else if (items >= 5) {
        total_price *= 0.9; // 10% discount
    }

    printf("%d\n", total_price);
    return 0;
}
```

Explanation:

- We read the number of items and the price per item.
- We calculate the total price before any discounts.
- We apply the discount based on the number of items purchased and print the final price.

Problem 14: Movie Ticket Price

Problem Statement:

A cinema offers different ticket prices based on age:

- Below 12 years: 5 units.
- 12 to 60 years: 10 units.
- Above 60 years: 7 units.

Input: A single integer representing the age of the customer.
Output: A single integer representing the ticket price.

Solution:

c

```
Copy code
#include <stdio.h>

int main() {
    int age;
    scanf("%d", &age);

    if (age < 12) {
        printf("5\n");
    } else if (age <= 60) {
        printf("10\n");
    } else {
        printf("7\n");
    }
    return 0;
}
```

Explanation:

- We read the age of the customer.
- We classify the ticket price based on the age using conditional statements and print the corresponding price.

Problem 15: Discounted Fuel Price

Problem Statement:

A fuel station offers discounts based on the quantity of fuel purchased:

- For purchases below 20 liters: no discount.
- For purchases between 20 and 40 liters: 5% discount.
- For purchases above 40 liters: 10% discount.

Input: Two integers: the number of liters of fuel purchased and the price per liter.
Output: A single integer representing the final price of fuel after applying the discount.

Solution:

c

```
Copy code
#include <stdio.h>

int main() {
    int liters, price_per_liter;
    scanf("%d %d", &liters, &price_per_liter);

    int total_price = liters * price_per_liter;

    if (liters >= 20 && liters <= 40) {
        total_price *= 0.95; // 5% discount
    } else if (liters > 40) {
        total_price *= 0.90; // 10% discount
    }

    printf("%d\n", total_price);
    return 0;
}
```

Explanation:

- We read the number of liters and the price per liter.
- We calculate the total price before any discounts.
- We apply the discount based on the quantity of fuel purchased and print the final price.

Problem 16: Overtime Pay

Problem Statement:

An employee is paid overtime based on the number of hours worked:

- Regular hours: 40 hours per week.
- Overtime pay: 1.5 times the hourly rate for hours worked beyond 40 hours.

Input: Two integers: the number of hours worked and the hourly rate.
Output: A single integer representing the total pay, including any overtime pay.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int hours_worked, hourly_rate;
    scanf("%d %d", &hours_worked, &hourly_rate);

    int total_pay = 0;

    if (hours_worked <= 40) {
        total_pay = hours_worked * hourly_rate;
    } else {
        total_pay = 40 * hourly_rate + (hours_worked - 40) *
hourly_rate * 1.5;
    }

    printf("%d\n", total_pay);
    return 0;
}
```

Explanation:

- We read the number of hours worked and the hourly rate.
- We calculate the total pay considering the regular hours and any overtime.
- We print the total pay.

Problem 17: Loan Eligibility

Problem Statement:

You are tasked with determining if a person is eligible for a loan. The criteria for loan eligibility are as follows:

- Age must be at least 21 years.
- Monthly income must be at least 3000 units.

Input: Two integers: the age of the person and their monthly income.
Output: A single line containing either "Eligible for loan" or "Not eligible for loan".

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int age, income;
    scanf("%d %d", &age, &income);

    if (age >= 21 && income >= 3000) {
        printf("Eligible for loan\n");
    } else {
        printf("Not eligible for loan\n");
    }
    return 0;
}
```

Explanation:

- We read the person's age and income.
- We check both eligibility criteria using a conditional statement and print the corresponding message.

Problem 18: Shopping Cart Total

Problem Statement:

A store calculates the total cost of items purchased based on the quantity and price per item:

- If the total exceeds 100 units, a 10% discount is applied.

Input: Two integers: the number of items and the price per item.
Output: A single integer representing the final price after applying any discount.

Solution:

```
c
```

```
Copy code
#include <stdio.h>

int main() {
    int items, price_per_item;
    scanf("%d %d", &items, &price_per_item);

    int total_price = items * price_per_item;

    if (total_price > 100) {
        total_price *= 0.9; // 10% discount
    }

    printf("%d\n", total_price);
    return 0;
}
```

Explanation:

- We read the number of items and the price per item.
- We calculate the total price before any discounts.
- If the total price exceeds 100, we apply the discount and print the final price.

Problem 19: Tax Calculation

Problem Statement:

A person's income tax is calculated based on the following slabs:

- Up to 5000 units: no tax.
- 5001 to 15000 units: 10% tax.
- Above 15000 units: 20% tax.

Input: A single integer representing the income.
Output: A single integer representing the total tax amount.

Solution:

c

```
Copy code
#include <stdio.h>

int main() {
    int income;
    scanf("%d", &income);
    int tax = 0;

    if (income <= 5000) {
        tax = 0;
    } else if (income <= 15000) {
        tax = (income - 5000) * 0.1; // 10% tax
    } else {
        tax = (10000 * 0.1) + (income - 15000) * 0.2; // 10% on first
10k and 20% on remaining
    }

    printf("%d\n", tax);
    return 0;
}
```

Explanation:

- We read the income from input.
- We calculate the tax based on the income slab using conditional statements and print the total tax amount.

Problem 20: Exam Result

Problem Statement:

A student's exam result is classified as follows:

- Score ≥ 60: "Pass"
- Score < 60: "Fail"

Input: A single integer representing the score.
Output: A single line containing either "Pass" or "Fail".

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int score;
    scanf("%d", &score);

    if (score >= 60) {
        printf("Pass\n");
    } else {
        printf("Fail\n");
    }
    return 0;
}
```

Explanation:

- We read the score from input.
- We check if the score is 60 or above using a conditional statement and print the corresponding result.

Problem 21: Elevator Capacity

Problem Statement:

An elevator has a maximum weight limit of 500 kg. If the total weight exceeds 500 kg, the elevator will not operate. Write a program that takes the total weight of the people in the elevator and prints "Overload" if the weight exceeds 500 kg, otherwise "Safe to operate".

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
```

```
    int weight;
    scanf("%d", &weight);

    if (weight > 500) {
        printf("Overload\n");
    } else {
        printf("Safe to operate\n");
    }
    return 0;
}
```

Problem 22: Bus Seat Allocation

Problem Statement:

A bus has a limited number of seats:

- The first 10 seats are reserved for priority passengers.
- The next 20 seats are for general passengers.
- After that, standing passengers are allowed.

Write a program that checks the seat availability based on the number of passengers.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int position;
    scanf("%d", &position);

    if (position <= 10) {
        printf("Priority\n");
    } else if (position <= 30) {
        printf("General\n");
    } else {
        printf("Standing\n");
```

```
    }
    return 0;
}
```

Problem 23: Petrol Pump Price

Problem Statement:

A petrol pump offers a discount based on the day of the week:

- On Monday, there is a 5% discount.

Write a program that takes the day of the week (1 for Monday, 2 for Tuesday, etc.) and the price of petrol, and calculates the price after applying the discount if applicable.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int day, price;
    scanf("%d %d", &day, &price);

    if (day == 1) {
        price *= 0.95; // 5% discount
    }

    printf("%d\n", price);
    return 0;
}
```

Problem 24: Simple Calculator

Problem Statement:

Implement a simple calculator that supports addition, subtraction, multiplication, and division. Write a program that performs the calculation based on the operator provided.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int a, b;
    char op;
    scanf("%d %d %c", &a, &b, &op);

    switch (op) {
        case '+':
            printf("%d\n", a + b);
            break;
        case '-':
            printf("%d\n", a - b);
            break;
        case '*':
            printf("%d\n", a * b);
            break;
        case '/':
            if (b != 0) {
                printf("%d\n", a / b);
            } else {
                printf("Division by zero error\n");
            }
            break;
        default:
            printf("Invalid operator\n");
    }
    return 0;
}
```

Problem 25: Traffic Light Timing

Problem Statement:

A traffic light operates on a timer:

- For the first 30 seconds, the light is green.
- For the next 10 seconds, the light is yellow.
- For the next 60 seconds, the light is red.

Write a program that determines the light's current color based on the number of seconds elapsed.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int seconds;
    scanf("%d", &seconds);

    seconds %= 100; // Cycle repeats every 100 seconds

    if (seconds < 30) {
        printf("Green\n");
    } else if (seconds < 40) {
        printf("Yellow\n");
    } else {
        printf("Red\n");
    }
    return 0;
}
```

Problem 26: Hotel Room Pricing

Problem Statement:

A hotel offers three types of rooms:

- Standard: 100 units per night.
- Deluxe: 200 units per night.
- Suite: 500 units per night.

Write a program that calculates the total cost of the stay based on the number of nights and the room type.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int nights, room_type;
    scanf("%d %d", &nights, &room_type);

    int cost;
    if (room_type == 1) {
        cost = nights * 100;
    } else if (room_type == 2) {
        cost = nights * 200;
    } else {
        cost = nights * 500;
    }

    printf("%d\n", cost);
    return 0;
}
```

Problem 27: Electricity Bill Calculation

Problem Statement:

An electricity provider charges customers based on their monthly consumption:

- For consumption up to 100 units: 5 units per kWh.

- For consumption between 101 and 300 units: 8 units per kWh.
- For consumption above 300 units: 12 units per kWh.

Write a program that calculates the total electricity bill based on the consumption in kWh.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int consumption;
    scanf("%d", &consumption);

    int bill;

    if (consumption <= 100) {
        bill = consumption * 5;
    } else if (consumption <= 300) {
        bill = (100 * 5) + ((consumption - 100) * 8);
    } else {
        bill = (100 * 5) + (200 * 8) + ((consumption - 300) * 12);
    }

    printf("%d\n", bill);
    return 0;
}
```

Problem 28: Student Grade Calculation

Problem Statement:

A school assigns grades based on exam scores:

- If the score is ≥ 90, the grade is "A".
- If the score is between 80 and 89, the grade is "B".
- If the score is between 70 and 79, the grade is "C".
- If the score is between 60 and 69, the grade is "D".

- If the score is less than 60, the grade is "F".

Write a program that calculates the grade based on the score.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int score;
    scanf("%d", &score);

    if (score >= 90) {
        printf("A\n");
    } else if (score >= 80) {
        printf("B\n");
    } else if (score >= 70) {
        printf("C\n");
    } else if (score >= 60) {
        printf("D\n");
    } else {
        printf("F\n");
    }
    return 0;
}
```

Problem 29: Cinema Ticket Pricing

Problem Statement:

A cinema charges tickets based on the day of the week:

- On weekdays (Monday to Friday), the ticket price is 100 units.
- On weekends (Saturday and Sunday), the ticket price is 150 units.

Write a program that calculates the ticket price based on the day of the week.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int day;
    scanf("%d", &day);

    if (day >= 1 && day <= 5) {
        printf("100\n");
    } else {
        printf("150\n");
    }
    return 0;
}
```

Problem 30: Internet Service Provider

Problem Statement:

An internet service provider offers three types of packages:

- Package A: Up to 50 GB for 500 units.
- Package B: Up to 100 GB for 800 units.
- Package C: Unlimited data for 1200 units.

Write a program that calculates the monthly bill based on the chosen package.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int package;
    scanf("%d", &package);

    int bill;
```

```
    if (package == 1) {
        bill = 500;
    } else if (package == 2) {
        bill = 800;
    } else {
        bill = 1200;
    }

    printf("%d\n", bill);
    return 0;
}
```

Problem 31: Vehicle Speed Check

Problem Statement:

A speed camera issues fines based on the speed of a vehicle:

- Speed ≤ 60 km/h: No fine.
- Speed between 61 and 100 km/h: Fine of 100 units.
- Speed above 100 km/h: Fine of 200 units.

Write a program that determines the fine based on the vehicle's speed.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int speed;
    scanf("%d", &speed);

    int fine;
    if (speed <= 60) {
        fine = 0;
    } else if (speed <= 100) {
        fine = 100;
```

```
    } else {
        fine = 200;
    }

    printf("%d\n", fine);
    return 0;
}
```

Problem 32: Flight Baggage Allowance

Problem Statement:

An airline allows free checked baggage up to a certain weight:

- Baggage ≤ 15 kg: No extra charge.
- Baggage between 16 and 30 kg: Charge of 50 units.
- Baggage above 30 kg: Charge of 100 units.

Write a program that calculates the extra charge based on the baggage weight.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int weight;
    scanf("%d", &weight);

    int charge;
    if (weight <= 15) {
        charge = 0;
    } else if (weight <= 30) {
        charge = 50;
    } else {
        charge = 100;
    }
```

```
    printf("%d\n", charge);
    return 0;
}
```

Problem 33: Taxi Fare Based on Distance

Problem Statement:

A taxi company calculates fares based on the distance traveled:

- For distances ≤ 3 km: Flat fare of 30 units.
- For distances between 4 and 10 km: 10 units per km.
- For distances above 10 km: 8 units per km.

Write a program that calculates the taxi fare based on the distance traveled.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int distance;
    scanf("%d", &distance);

    int fare;
    if (distance <= 3) {
        fare = 30;
    } else if (distance <= 10) {
        fare = 30 + (distance - 3) * 10;
    } else {
        fare = 30 + (7 * 10) + ((distance - 10) * 8);
    }

    printf("%d\n", fare);
    return 0;
}
```

Problem 34: Restaurant Bill with Tip

Problem Statement:

A restaurant calculates the tip based on the total bill:

- If the bill is less than 100 units: 5% tip.
- If the bill is between 100 and 500 units: 10% tip.
- If the bill is more than 500 units: 15% tip.

Write a program that calculates the total bill including the tip.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int bill;
    scanf("%d", &bill);

    int tip;
    if (bill < 100) {
        tip = bill * 0.05;
    } else if (bill <= 500) {
        tip = bill * 0.10;
    } else {
        tip = bill * 0.15;
    }

    printf("%d\n", bill + tip);
    return 0;
}
```

Problem 35: Car Loan Payment Calculation

Problem Statement:

A bank offers car loans with different interest rates based on the loan amount:

- Loan ≤ 200,000 units: 5% interest.
- Loan between 200,001 and 500,000 units: 7% interest.
- Loan above 500,000 units: 10% interest.

Write a program that calculates the total interest to be paid on the loan.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int loan;
    scanf("%d", &loan);

    int interest;
    if (loan <= 200000) {
        interest = loan * 0.05;
    } else if (loan <= 500000) {
        interest = loan * 0.07;
    } else {
        interest = loan * 0.10;
    }

    printf("%d\n", interest);
    return 0;
}
```

Problem 36: Fuel Cost Based on Distance

Problem Statement:

A car consumes 1 liter of fuel per 15 kilometers traveled, and the price of fuel is 80 units per liter. Calculate the fuel cost based on the distance traveled.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int distance;
    scanf("%d", &distance);

    int fuel_needed = distance / 15; // Calculate fuel needed
    if (distance % 15 != 0) {
        fuel_needed++; // Round up for any remaining distance
    }

    int cost = fuel_needed * 80; // Calculate cost
    printf("%d\n", cost);
    return 0;
}
```

Problem 37: Mobile Data Usage Bill

Problem Statement:

A mobile network charges users based on their data usage:

- Up to 2 GB: 100 units per GB.
- Between 3 GB and 10 GB: 80 units per GB.
- Above 10 GB: 50 units per GB.

Write a program that calculates the total data usage bill.

Solution:

```
c
Copy code
```

```c
#include <stdio.h>

int main() {
    int data_usage;
    scanf("%d", &data_usage);

    int bill;

    if (data_usage <= 2) {
        bill = data_usage * 100;
    } else if (data_usage <= 10) {
        bill = (2 * 100) + ((data_usage - 2) * 80);
    } else {
        bill = (2 * 100) + (8 * 80) + ((data_usage - 10) * 50);
    }

    printf("%d\n", bill);
    return 0;
}
```

Problem 38: Laundry Service Charges

Problem Statement:

A laundry service charges for washing clothes based on the number of items:

- Up to 5 items: 20 units per item.
- 6 to 10 items: 15 units per item.
- Above 10 items: 10 units per item.

Write a program that calculates the total laundry bill based on the number of items.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
```

```
    int items;
    scanf("%d", &items);

    int bill;
    if (items <= 5) {
        bill = items * 20;
    } else if (items <= 10) {
        bill = (5 * 20) + ((items - 5) * 15);
    } else {
        bill = (5 * 20) + (5 * 15) + ((items - 10) * 10);
    }

    printf("%d\n", bill);
    return 0;
}
```

Problem 39: Hotel Room Charge Calculation

Problem Statement:

A hotel charges for rooms based on the number of nights and the type of room:

- Standard room: 100 units per night.
- Deluxe room: 200 units per night.
- Suite: 500 units per night.

Write a program that calculates the total cost of staying at the hotel.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int nights, room_type;
    scanf("%d %d", &nights, &room_type);

    int cost;
```

```
    if (room_type == 1) {
        cost = nights * 100;
    } else if (room_type == 2) {
        cost = nights * 200;
    } else {
        cost = nights * 500;
    }

    printf("%d\n", cost);
    return 0;
}
```

Problem 40: Gym Membership Plan

Problem Statement:

A gym offers different membership plans:

- 1-month: 50 units.
- 6-months: 250 units.
- 12-months: 450 units.

Write a program that calculates the total cost of the gym membership based on the plan chosen.

Solution:

```
c
Copy code
#include <stdio.h>

int main() {
    int plan;
    scanf("%d", &plan);

    int cost;
    if (plan == 1) {
        cost = 50;
    } else if (plan == 2) {
```

```
        cost = 250;
    } else {
        cost = 450;
    }

    printf("%d\n", cost);
    return 0;
}
```