

Question 1: The Smart Elevator

Concept: Basic if statement

Write a program for a smart elevator system. The elevator has a maximum weight capacity of 500 kg. Input the current weight in the elevator. If the weight exceeds 500 kg, display "OVERLOAD! Please exit the elevator." If it's within limits, display "Safe to go!"

Sample Input/Output:

Enter current weight (kg): 520

OVERLOAD! Please exit the elevator.

Question 2: The Temperature Converter with Comfort Check

Concept: if-else statement with relational operators

Create a program that converts Celsius to Fahrenheit and tells if the weather is comfortable. Input temperature in Celsius, convert it to Fahrenheit ($F = C \times 9/5 + 32$). If the Fahrenheit temperature is between 65 and 80, print "Perfect weather!", otherwise print "Not ideal weather."

Sample Input/Output:

Enter temperature in Celsius: 25

Temperature in Fahrenheit: 77.00

Perfect weather!

Question 3: The Gaming Score Rank System

Concept: if-else if-else ladder

Design a gaming rank system. Input a player's score and assign ranks:

- Score ≥ 5000 : "LEGENDARY"
- Score ≥ 3000 : "MASTER"
- Score ≥ 1500 : "EXPERT"
- Score ≥ 500 : "INTERMEDIATE"

- Score < 500: "BEGINNER"

Also display a motivational message based on their rank.

Sample Input/Output:

Enter your score: 3500

Rank: MASTER

Keep pushing! Legendary status awaits!

Question 4: The Smart Traffic Light System

Concept: Nested if-else with logical operators

Implement the smart traffic signal system mentioned in the lab. Input whether sensor 1 and sensor 2 detect vehicles (1 for yes, 0 for no).

- Both sensors detect vehicles: "GREEN LIGHT - 30 seconds"
- Only one sensor detects: "GREEN LIGHT - 45 seconds"
- No sensors detect: "RED LIGHT - Stay"

Sample Input/Output:

Sensor 1 (1/0): 1

Sensor 2 (1/0): 0

GREEN LIGHT - 45 seconds

Question 5: The Movie Ticket Pricing System

Concept: Multiple conditions with logical operators

Create a cinema ticket pricing system. Input age and day of week (1-7, where 1=Monday).

- Children (age < 12): \$5
- Students (age 12-22): \$8
- Adults (age > 22): \$12
- Special: 50% discount on Wednesdays (day 3) for everyone!

Sample Input/Output:

Enter age: 20

Enter day (1-7): 3

Student ticket with Wednesday discount: \$4.00

Question 6: The Restaurant Menu Selector 🍕

Concept: switch statement basics

Create a restaurant ordering system using switch. Display menu:

1. Pizza - \$12
2. Burger - \$8
3. Pasta - \$10
4. Salad - \$6
5. Soup - \$5

Input choice and quantity, calculate total bill.

Sample Input/Output:

=== MENU ===

1. Pizza - \$12
2. Burger - \$8
3. Pasta - \$10
4. Salad - \$6
5. Soup - \$5

Enter choice (1-5): 2

Enter quantity: 3

Your order: 3 Burger(s)

Total bill: \$24

Question 7: The Calculator with Error Handling 🧮

Concept: switch with character input and error checking

Build a calculator using switch that takes two numbers and an operator (+, -, *, /, %). Include error handling for division by zero and invalid operators.

Sample Input/Output:

Enter first number: 10

Enter operator (+, -, *, /, %): /

Enter second number: 0

Error: Division by zero is not allowed!

Question 8: The Grade Calculator with Performance Report

Concept: Combining if-else and switch

Input marks (0-100) and calculate grade using if-else:

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

Then use switch on the grade to provide detailed feedback about performance and study recommendations.

Sample Input/Output:

Enter marks (0-100): 85

Grade: B

Performance: Good work! You're doing well.

Recommendation: A little more effort can get you to A grade!

Question 9: The Water Park Entry System

Concept: Complex conditions with multiple criteria

Create a water park entry system. Check if a person can ride the "Thunder Wave":

- Height must be between 120cm and 200cm
- Age must be at least 10 years
- Must know swimming (input as 1 for yes, 0 for no)
- Health condition must be good (no heart problems - input as 1 for yes, 0 for no)

All conditions must be met. If rejected, specify which condition(s) failed.

Sample Input/Output:

Enter height (cm): 150
Enter age: 12
Can swim? (1/0): 1
Heart problems? (1/0): 0

✓ Welcome to Thunder Wave! Enjoy your ride!

Question 10: The ATM Transaction System

Concept: Comprehensive use of all decision structures

Design an ATM system with initial balance of \$1000. Show menu using printf:

1. Check Balance
2. Withdraw Money
3. Deposit Money
4. Quick Cash (switch for amounts: \$20, \$50, \$100, \$200)
5. Exit

For withdrawal: Check if amount is available, if amount is multiple of 10, and apply \$2 fee for withdrawals over \$100. For deposit: Only accept positive amounts less than \$5000.

Sample Input/Output:

```
=== ATM MENU ===  
Current Balance: $1000.00  
1. Check Balance  
2. Withdraw Money  
3. Deposit Money  
4. Quick Cash  
5. Exit  
Enter choice: 2  
Enter withdrawal amount: $150  
Transaction successful!  
Transaction fee: $2.00  
Amount withdrawn: $150.00  
New balance: $848.00
```

Bonus Challenge Question: The Rock-Paper-Scissors Game

Concept: Advanced decision making with modular arithmetic

Create a Rock-Paper-Scissors game where the computer's choice is generated using a simple formula based on user input. Input user choice (1=Rock, 2=Paper, 3=Scissors) and a "lucky number". Computer's choice = $(\text{lucky_number} \% 3) + 1$. Determine winner using the rules:

- Rock beats Scissors
- Scissors beats Paper
- Paper beats Rock
- Same choice = Draw

Display choices with ASCII art and declare the winner!

Sample Input/Output:

Enter your choice (1=Rock, 2=Paper, 3=Scissors): 1

Enter your lucky number: 7

You chose: Rock

Computer chose: Paper

Computer wins! Paper covers Rock!