



ASSIGNMENT 02 **PROGRAMMING FUNDAMENTALS**

INSTRUCTIONS:

- 1) Plagiarism is strictly forbidden. Each individual must complete the work independently as it is part of the learning process.
- 2) The naming format must be strictly followed as given in the instruction note:

23P-0603_Aimalkhan_Q1.c

23P-0603_Aimalkhan_Q2.c

23P-0603_Aimalkhan_Q3.c

23P-0603_Aimalkhan_Q4.c

- 3) Anyone who fails to follow the naming format will receive zero marks.
- 4) Everything covered in class so far is included in the assignment, so don't go beyond that.
- 5) Comment your code properly.
- 6) Upload four .c files.
- 7) Anyone failed to follow these instructions will get a straight zero.
- 8) Don't use AI for this assignment try to do it on your own it will be helpful in your sessional exam.

Q1. Write a C program that takes a 4-digit number as input. Using only if statements, check if it is a special palindrome:

Checking for palindrome.

It is special if, in addition, the sum of the middle two digits is a prime number.

Example:

Input: 3113 → Output: Special Palindrome

Input: 5225 → Output: Not Special Palindrome

Q2: A university offers scholarships based on multiple conditions. Write a C program using only nested-if-else to determine the grade + scholarship eligibility.

Rules:

1. A student must have attendance $\geq 75\%$ and internal assessment marks ≥ 40 to be eligible.
2. If eligible, assign grade based on final exam marks:

Marks ≥ 90 → Grade A+

Marks ≥ 80 and < 90 → Grade A

Marks ≥ 70 and < 80 → Grade B

Marks ≥ 60 and < 70 → Grade C

Marks ≥ 50 and < 60 → Grade D

Marks < 50 → Fail

3. In addition, scholarship is given if:

Attendance $\geq 90\%$ AND Grade is A+ → 100% Scholarship

Attendance $\geq 85\%$ AND Grade is A → 50% Scholarship

Otherwise → No Scholarship

Example:

Attendance = 92, Internal = 55, Final = 91 → Grade A+, Scholarship 100%

Attendance = 86, Internal = 47, Final = 82 → Grade A, Scholarship 50%

Attendance = 70, Internal = 60, Final = 85 → Not Eligible

Q3. Write a C program using a switch statement to calculate the electricity bill for different types of consumers.

Steps:

1. Ask the user to select consumer type:

1 → Domestic

2 → Commercial

3 → Industrial

2. Then ask for the number of units consumed.
3. Using the switch (for consumer type) and if-else (for slab calculation), calculate the bill according to the following rules:

- **Domestic:**

First 100 units → Rs. 5 per unit

Next 200 units → Rs. 7 per unit

Above 300 units → Rs. 10 per unit

- **Commercial:**

First 200 units → Rs. 8 per unit

Next 300 units → Rs. 12 per unit

Above 500 units → Rs. 15 per unit

- **Industrial:**

First 500 units → Rs. 15 per unit

Above 500 units → Rs. 20 per unit

Extra condition: If units > 1000, add a surcharge of 10% on total bill.

4. Finally, display the total bill with a proper message.

Example Runs:

Input: Consumer = Domestic, Units = 350 → Output: Rs. 2450

Input: Consumer = Industrial, Units = 1200 → Output: Rs. 22000 + surcharge = Rs. 24200

Q4 You have to design a mini encoding and decoding system using only the following topics:

- **if**
- **if-else**
- **nested if-else**
- **switch statement**

The purpose of the program is to take two inputs from the user:

1. The first letter of the user's name (only one character).
2. The identity number (a 3-digit number, e.g.123).it should be a three-digit number.

The system will then encode this information into a secret code and later be able to decode it back to the original input.

Encoding Rules (step by step):

1. **Check the first letter of the name (using if-else):**

If the letter is a vowel (a, e, i, o, u), then you must add 5 to each digit of the ID number.

If the letter is a consonant, then you must subtract 2 from each digit of the ID number.

Example:

Letter = A (a vowel), ID = 123 → New ID = 678 (1+5=6, 2+5=7, 3+5=8).

Letter = B (a consonant), ID = 456 → New ID = 234 (4-2=2, 5-2=3, 6-2=4).

2. Transform the name letter (using switch):

You must encode the letter into its reverse alphabet. For example:

A → Z

B → Y

C → X

D → W

... and so on.

(So, you are required to use a `switch` statement here, not `if-else`.)

3. Special Cases (use if-else-if ladder):

After transformation of digits, sometimes the new digit may become:

greater than 9 → subtract 10 to keep it within 0–9 range.

less than 0 → add 10 to bring it back within 0–9 range.

(Example: if digit = 8 and you add 5 → 13, then store it as 3).

Decoding Rules (reverse the process):

1. Check the encoded name's first letter (using switch again):
 - o Z → A, Y → B, X → C, ... etc.
2. If the decoded name is a vowel, then subtract 5 from each digit of the encoded ID number.
If it is a consonant, then add 2 back to each digit.
3. Again, if digits go above 9 or below 0, fix them using if-else-if ladder.

Example Walkthrough:

Input:

Name's first letter: A
ID number: 123

Encoding:

A is a vowel → Add 5 → (1+5=6, 2+5=7, 3+5=8) → 678

A → Z (switch rule)

Encoded Result = Z678

Decoding:

Z → A (reverse using switch)

Since A is a vowel → Subtract 5 → (6-5=1, 7-5=2, 8-5=3) → 123

Final Decoded Result = A123

Hint for Students:

Use if and if-else for vowel/consonant checking.

Use nested if-else for checking each digit of the ID and modifying it.

Use an if-else-if ladder for handling special cases when digits go out of range (less than 0 or greater than 9).

Use switch for converting letters into reverse alphabet during both encoding and decoding.