Explanation problem 1:

Bit-fields allow you to specify the number of bits used for an integer member.

  - roll_number : 4 uses only 4 bits.
  - age : 8 uses 8 bits.
  - marks : 3 uses 3 bits.

- These bit-fields are packed within bytes by the compiler, but alignment padding **may still occur** depending on the platform (e.g., GCC usually aligns to 1 byte or more).
- char address[50] and char name[30] are stored normally as arrays of characters.

Memory Usage

- roll_number (4 bits) + age (8 bits) + marks (3 bits) = **15 bits** → may be padded to 16 bits = 2 bytes.
- address[50] = 50 bytes
- name[30] = 30 bytes
- Likely total: 2 (bit-fields) + 50 + 30 = **82 bytes**, but sizeof(student_info) might return **84 bytes** due to padding.

Would you like a diagram showing how memory is arranged for this struct?

4o