Class 2 Class Program Pull Up Method

```
uint8_t val;  // Declare an 8-bit unsigned variable to store the PORTB value
void main(void)
{
    TRISB = 0xF0;  // Configure upper nibble (RB4-RB7) as input (1), lower nibble (RB0-RB3) as output
    TRISC = 0x00;  // Configure PORTC as output (all bits set to 0 for output)
    PORTB = 0x00;  // Initialize PORTB to 0 (not necessary as pull-up resistors will handle inputs)

    while(1)  // Infinite loop to continuously monitor inputs and update outputs
    {
        val = PORTB;  // Read the value from PORTB (RB4-RB7 are used as input)
        switch (val)  // Check the input value and decide the output on PORTC
        {
            case 0xE0:  // If RB5, RB6, RB7 are HIGH, and RB4 is LOW (0b1110 0000)
            {
                PORTC = 0x02;  // Set RC1 HIGH (0000 0010 in binary)
                break;
            }
            case 0xD0:  // If RB4, RB6, RB7 are HIGH, and RB5 is LOW (0b1101 0000)
            {
                PORTC = 0x04;  // Set RC2 HIGH (0000 0100 in binary)
                break;
            }
            case 0xB0:  // If RB4, RB5, RB7 are HIGH, and RB6 is LOW (0b1011 0000)
            {
                PORTC = 0x06;  // Set RC1 and RC2 HIGH (0000 0110 in binary)
                break;
            }
            case 0x70:  // If RB4, RB5, RB6 are HIGH, and RB7 is LOW (0b0111 0000)
            {
                PORTC = 0x00;  // Turn OFF all PORTC outputs
                break;
            }
            default:  // If none of the specific cases match
            {
                PORTC = 0x00;  // Keep PORTC OFF
            }
        }
    }
    return;  // This statement is never reached due to the infinite loop
}

```