|  |  |
| --- | --- |
| **Set a bit in C++** | |
| #include <iostream>  using namespace std;  int main() {  int num = 5; // Binary: 0101  int bitmask = 1 << 2; // Binary: 0100  int result = num | bitmask; // Binary: 0101 (Decimal: 5)    cout << result << endl; // Output: 5  return 0;  } | **Dry Run in Table:**   | **Step** | **Variable** | **Value** | **Explanation** | | --- | --- | --- | --- | | 1 | num | 5 (Binary: 0101) | Initialize num as 5. | | 2 | bitmask | 4 (Binary: 0100) | Compute bitmask = 1 << 2 (left shift 1 by 2 positions). | | 3 | `num | bitmask` | 5 (Binary: 0101) | | 4 | result | 5 (Binary: 0101) | Store the result of `num | | 5 | cout | 5 | Print the value of result (which is 5). |   **Detailed Explanation of Key Operations:**   1. **Step 1: Initialize num** num is set to 5. Its binary representation is 0101. 2. **Step 2: Create bitmask** bitmask = 1 << 2 shifts the binary 0001 two positions to the left. The result is 0100, which is 4 in decimal. 3. **Step 3: Perform Bitwise OR (|)** result = num | bitmask → 0101 (num) 0100 (bitmask) The result of 0101 | 0100 is 0101, which is 5 in decimal. 4. **Step 4: Store and Output result** The result of the bitwise OR operation is 5. The program prints the result: **Output**: 5   **Final Output:**  5 |
| 5 | |