Bitset → an array of bool (Stores Binary of a number)

→ bitset<8> bl(n);

⇒ cout << bl; // output - 00000010 (n=2)

⇒ for(int i=0; i<8; i++){
    cout << bl[i]; // output - 0000 0010 (n=2)
}

1) Initialization

bitset<32> bset1; // initializes with all bits 0 → // 00000000 00000000 00000000 00000000

bitset<32> bset2(20); // initialized with bits of 20

bitset<32> bset3(string("1100")); // initialized with binary string → // 00000000 00000000 00000000 00001100

2) array indexing operator []

bset1[6] = 1; // setting the 6th index.

3) count function

int number of 1 = bset3.count();

// it returns no. of set bits in bitset.

4) size function

bset1.size(); // return total number of bits in bitset

int numberofo = bset1.size() - number of 1; // number of 0

5) test function

cout << bset1.test(i); // return 1 if bit is set else return 0.

6) any function / none function

cout << bset1.any(); // return true if atleast 1 bit.

cout << bset1.none(); // return true, if none of the bit is set.

7) set() function

1 way bset1.set(); // sets all bits

2nd way bset1.set(pos, b); // makes bset1[pos] = b
ex bset1.set(4, 0);

3 way bset1.set(pos); // makes bset1[pos] = 1
i.e default is 1

Page No.
Date :

8) reset function

↳ same as set()
but it makes 0

↑

eg set8.reset(2); //makes set8[2]=0
set8.reset(); //makes all bits 0.

9) flip function (0⇔1)

set8.flip(2); → //flips 2nd bit

set8.flip(); //flips all bits.

Note

a^=b ⇒ a=a^b;

same as a+=b ⇒ a=a+b;

a&=b ⇒ a=a&b;

a|=b ⇒ a=a|b;

a^=b ⇒ a=a^b;

a>>=b ⇒ a=a>>b;