

[Back to Contest \(/contest/weekly-contest-279/\)](/contest/weekly-contest-279/)

Difficulty: Medium

- `Bitset(int size)` Initializes the Bitset with `size` bits, all of which are `0`.
- `void fix(int idx)` Updates the value of the bit at the index `idx` to `1`. If the value was already `1`, no change occurs.
- `void unfix(int idx)` Updates the value of the bit at the index `idx` to `0`. If the value was already `0`, no change occurs.
- `void flip()` Flips the values of each bit in the Bitset. In other words, all bits with value `0` will now have value `1` and vice versa.
- `boolean all()` Checks if the value of **each** bit in the Bitset is `1`. Returns `true` if it satisfies the condition, `false` otherwise.
- `boolean one()` Checks if there is **at least one** bit in the Bitset with value `1`. Returns `true` if it satisfies the condition, `false` otherwise.
- `int count()` Returns the **total number** of bits in the Bitset which have value `1`.
- `String toString()` Returns the current composition of the Bitset. Note that in the resultant string, the character at the i^{th} index should coincide with the value at the i^{th} bit of the Bitset.

```

Bitset bs = new Bitset(5); // bitset = "00000".
bs.fix(3);    // the value at idx = 3 is updated to 1, so bitset = "00010".
bs.fix(1);    // the value at idx = 1 is updated to 1, so bitset = "01010".
bs.flip();    // the value of each bit is flipped, so bitset = "10101".
bs.all();     // return False, as not all values of the bitset are 1.
bs.unfix(0);  // the value at idx = 0 is updated to 0, so bitset = "00101".
bs.flip();    // the value of each bit is flipped, so bitset = "11010".
bs.one();     // return True, as there is at least 1 index with value 1.
bs.unfix(0);  // the value at idx = 0 is updated to 0, so bitset = "01010".
bs.count();   // return 2, as there are 2 bits with value 1.
bs.toString(); // return "01010", which is the composition of bitset.

```

- `1 <= size <= 105`
- `0 <= idx <= size - 1`
- At most `105` calls will be made **in total** to `fix`, `unfix`, `flip`, `all`, `one`, `count`, and `toString`.
- At least one call will be made to `all`, `one`, `count`, or `toString`.
- At most 5 calls will be made to `toString`.


```
1 ▾ /**
2   * @param {number} size
3   */
4 ▾ var Bitset = function(size) {
5
6   };
7
8 ▾ /**
9   * @param {number} idx
10  * @return {void}
11  */
```

```
12 Bitset.prototype.fix = function(idx) {
13 }
14 };
15
16 /**
17  * @param {number} idx
18  * @return {void}
19  */
20 Bitset.prototype.unfix = function(idx) {
21 }
22 };
23
24 /**
25  * @return {void}
26  */
27 Bitset.prototype.flip = function() {
28 }
29 };
30
31 /**
32  * @return {boolean}
33  */
34 Bitset.prototype.all = function() {
35 }
36 };
37
38 /**
39  * @return {boolean}
40  */
41 Bitset.prototype.one = function() {
42 }
43 };
44
45 /**
46  * @return {number}
47  */
48 Bitset.prototype.count = function() {
49 }
50 };
51
52 /**
53  * @return {string}
54  */
55 Bitset.prototype.toString = function() {
56 }
57 };
58
59 /**
60  * Your Bitset object will be instantiated and called as such:
61  * var obj = new Bitset(size)
62  * obj.fix(idx)
63  * obj.unfix(idx)
64  * obj.flip()
65  * var param_4 = obj.all()
66  * var param_5 = obj.one()
67  * var param_6 = obj.count()
68  * var param_7 = obj.toString()
69  */
```

☐ Custom Testcase

Copyright © 2022 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)