Check if a number is positive, negative or zero using bit operators

Given a number N, check if it is positive, negative or zero without using conditional statements.

Examples:

```
Input : 30
Output : 30 is positive

Input : -20
Output : -20 is negative

Input: 0
Output: 0 is zero
```

The signed shift n>>31 converts every negative number into -1 and every other into 0.

When we do a -n>>31, if it is a positive number then it will return -1 as we are doing -n>>31 and the vice versa when we do for a negative number.

But when we do for 0 then n>>31 and -n>>31 both returns 0, so we get a formula:

1 + (n>>31) – (-n>>31)

..1) **when n is negative** :

1 +(-1)-0=0

..2) **when n is positive:**

1+0-(-1)=2

..3) **when n is 0:**

1+0-0=1

So we know it returns 0 when it is a negative number, it returns 1 when it is zero, and returns 2 when it is a positive number.

So to not use if statements we store at 0th index “negative”, 1st index “zero” and at 2nd index “positive”, and print according to it.