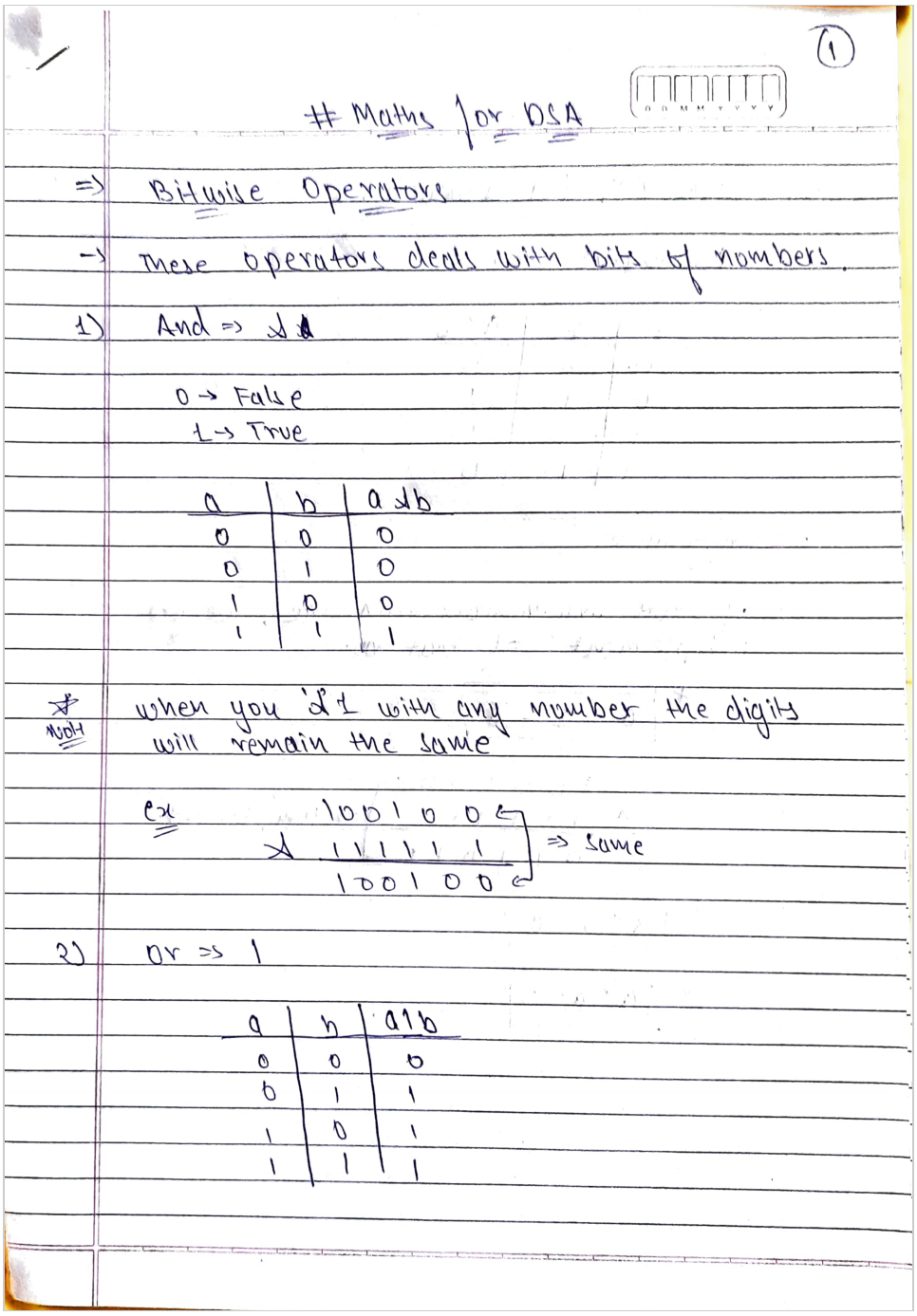
Math For DSA

31 March 2022 10:33



Maths for DSA

1

=> Bitwise Operators

-> These operators deals with bits of numbers.

1) And => &

0 -> False
1 -> True

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

Note: when you '&' 1 with any number the digits will remain the same

Ex
```
  1001000
& 1111111
---------
  1001000   => same
```

2) or => |

a	b	a | b
0	0	0
0	1	1
1	0	1
1	1	1

3) XOR (^) (If & only if)

(exclusive OR)

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

⇒ Observations:

If XOR any number with one, we get complement of that no.

* $a \wedge 1 = \bar{a}$

```
  1 0 0 1 0 0
^ 1 1 1 1 1 1   ⇒ complement
  0 1 1 0 1 1
```

* $a \wedge 0 = a$

* $a \wedge a = 0$

D D M M Y Y Y Y

Number System

(1) Decimal :- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
(Base 10)

$(357)_{10}$, $(10)_{10}$

(2) Binary :- only 0 & 1
(Base 2)

$(10)_{10} = (1010)_2$

$(7)_{10} = (111)_2$

(3) Octal :→ 0, 1, 2, 3, 4, 5, 6, 7
(Base 8)

$(8)_{10} = (10)_8$
$(9)_{10} = (11)_8$

(4) Hexa decimal :-
(Base 16)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
A, B, C, D, E, F

$(10)_{10} = (A)_{16}$ & $(12)_{10} = (C)_{16}$

Binary	Decimal	octal	Hex
0 0 0 0	0	0	0
0 0 0 1	1	1	1
0 0 1 0	2	2	2
0 0 1 1	3	3	3
0 1 0 0	4	4	4
0 1 0 1	5	5	5
0 1 1 0	6	6	6
0 1 1 1	7	7	7
1 0 0 0	8	10	8
1 0 0 1	9	11	9
1 0 1 0	10	12	A
1 0 1 1	11	13	B
1 1 0 0	12	14	C
1 1 0 1	16	15	D
1 1 1 0	14	16	E
1 1 1 1	15	17	F

=> Conversions :-

① Decimal to base b

=> Convert $(17)_{10}$ to base 2

keep dividing by base, take remainders in opposite

2	17	
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

$(17)_{10} \Rightarrow (10001)_2$

=> $(17)_{10}$ => $(?)_8$

8	17	
8	2	1 ↑
	0	2

$(17)_{10}$ => $(21)_8$

(2) Convert any base b to Decimal :-

=> $(10001)_2 = (?)_{10}$

Steps :- Multiply & add the power of base with digit

=> $1\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$

=> $16 + 0 + 0 + 0 + 1 \Rightarrow (17)_{10}$

=> $(21)_8$ => $(?)_{10}$

$2\times8^1 + 1\times8^0$

$= 16+1$ => $(17)_{10}$

Continuing with operators

(4) left shift operator (<<)

-> Shifts all the bits of a number to the left

Ex: $(10)_{10} \Rightarrow (1010)_2$

$(1010)_2 << 1$

1 0 1 0

$\Rightarrow (10100)_2 \Rightarrow (20)_{10}$

$a << 1 = 2a$

$\Downarrow$

$a << b = a \times 2^b$

(3) Right shift (>>)

=> shifts all the bits toward right

$(00\,1100\,1)_2 >> 1$

$(00\,1100)_2 \Rightarrow (1100)_2$

Note: Ignore leading zeroes

$a >> b = \frac{a}{2^b}$

Question

Q. Given a no. n find if its even or odd.

Note: Every no. is calculated in binary form

12+7 => $(1100)_2 + (111)_2$

=> 01100

00111

$(10011)_2$ => $2^4 + 2 + 1 = (19)_{10}$

=> Let's take an example

MSB LSB

(11001011)

this will be always even as it is multiplied by power 2.

But this bit will determine no. is even or odd

Note: If LSB is 1 no. is odd & LSB is 0 no. is even.

11001011

& 00000001

00000001

no. is odd

So,

If (a & 1 == 1) => odd

Else => Even.

2.Q All the elements in the array are repeating twice except 1, find that element

arr = [2, 3, 4, 1, 2, 1, 3, 6, 4]

=> Approach :-

i) Use XOR operator

ii) XOR have Associative property
i.e $2 \wedge 3 \wedge 4 = 4 \wedge 3 \wedge 2 = 3 \wedge 4 \wedge 2$

iii) And we know that
$a \wedge a = 0$

Hence all the duplicate elements will be eleminated & we will get the ans

Time complexity :- O(n)

Space complexity :- O(1)

3.Q Find ith bit of a no.

8 7 6 5 4 3 2 1
1 0 1 0 0 0 1 0

=> Find 5th bit

1 0 1 0 0 0 1 0
& 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0

If we are able to & number

=> we can & required bit with 1 & others with zero

=> we will get the ans.

=> 0 0 0 1 0 0 0 0 → (n-1) 0's

mask

This is called Bit masking

=> mask with (n-1) zeroes

* How to do that

Ans :- 1 << (n-1)

In this case :- 1 << 4

General Ans :- n & (1 << (n-1))

4.Q Set the i^{th} bit ?

OR given bit with 1

```
  1 0 1 0 1 1 0
OR 0 0 1 0 0 0 0
(1 0 1 0 1 1 0)2
```

General Ans:- n | (1<<(n-1))

5.Q Reset the i^{th} bit ?

```
  1 0 1 0 1 1 0
  0 0
  1 1 0 1 1 1 1
  1 0 0 0 1 1 0
```

Reset bit

→ How to get this

↓

Complement of (1<<(n-1))

General Ans:- n & ((1<<(n-1))^1)

6) arr = [3, 2, 3, 1, 7, 7, 8, 7, 8, 8]

10
10
11
10
1111
1111
10000
1111
10000
+ 10000
3374
%3333
0011 ⇒ Ans ⇒ 3

[Add All bits]

2	8	0
2	4	0
2	1	0
2	0	1
	0	

3 | 51
3
21

⇒ 2×3 + 7×3 + 8×3 + 3
⇒ 6 + 21 + 24 + 3
⇒ 51 + 3 = 54

54
3
18
3
6
3
2

2	54	0
2	27	1
2	13	1
2	6	1
2	3	0
2	1	1
	0	

1 1 0 1 1 0

7.Q Pascal's Triangle

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10  10  5   1
```

⇒ Find the sum of n^{th} row.

→ Sum of each row :- ${}^nC_0 + {}^nC_1 + {}^nC_n = 2^n$

For n^{th} row, sum = 2^{n-1}

Ans :- (1 << (n-1))

8.Q Check if the no. is power of 2 or not

⇒ we know that, $n = (n-1) + 1$

let's take an example

1000 = 0111 + 1

1000 & 0111 = 0 ✓

or

1010 & 1001 = 1000 ✗

If n == 0 return false

If n & (n-1) == 0, then it is power of 2

Q.1 Calculate a^b

Let's take an example of 3^6

now, $3^6 = 3 \times 3 \times 3 \times 3 \times 3 \times 3$

Time complexity:- O(n)

=> Optimised approach

$3^6 = 3(110)$ => Let's take this as n

This as base

Here, n = 110 , ans = 1

Now when, n & 1 => 0, Ignore

If n & 1 => 1, ans *= base

Function Power (int base, int power)

```
ans = 1
Power
while (a>0)

    If ((power & 1) == 1)
        ans = ans * base

    base = base x base
    power = power >> 1;
```

Q.9 Given a number n, find the no of set bits in it.

⇒ To count no. of set bits, do

$n = n \& (n-1)$

This will remove right most set bit

Ex $(7)_{10} = (1\ 1\ 1)_2$, count = 0

$1\ 1\ 1 = n$ count++

$\& \quad 1\ 1\ 0 \Rightarrow 6 \Rightarrow n-1$

$1\ 1\ 0$

right most set bit removed

now, $1\ 1\ 0 \Rightarrow n$ count++

$\& \quad 1\ 0\ 1 \Rightarrow 5 \Rightarrow n-1$

$1\ 0\ 0$

right most set bit removed

now $1\ 0\ 0 \Rightarrow n$

$\& \quad 0\ 1\ 1 \Rightarrow 3 \Rightarrow n-1$ count++

$0\ 0\ 0$

right most set bit removed

Ans = count = 3

```
while (n>0) {
    count++;
    n = n & (n-1)
}
```

10.0 Find XOR of no.s from 0 to a

No.s	XOR 0->a
0	0
1	0^1 = 1
2	1^2 = 3
3	3^3 = 0
4	0^4 = 4
5	4^5 = 1
6	1^6 = 7
7	7^7 = 0
8	0^8 = 8
9	8^9 = 1
10	1^10 = 11
11	11^11 = 0
12	0^12 = 0

1000
1001

=> when n%4 == 0, XOR of (0->n) is n itself

=> when n%4 == 1, XOR of (0->n) is 1

=> when n%4 == 2, XOR = n+1

=> when n%4 == 3, XOR = 0

Complexity, O(1)

11.0 Find XOR of no.s from a to b

=> Now we can calculate XOR from (0-b)

0 ... 0 .... b

we can calculate this

=> But we have extra XOR from [0->(a-1)]

=> If we XOR [0->b] ^ [0->a-1]

we will get our ans, as $a \wedge a = 0$

0 ... a ... b

Let this XOR f(b)

0 .... (a-1)^a ... b

Let this be f(a)

Now in f(b), we have extra XOR,

Now if we again XOR f(b) with f(a)

we will get the ans, as $a \wedge a = 0$

Now, Ans = f(b) ^ f(a)

11.9 Flip the Image (Leetcode)

(Google)

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Reverse every row}} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Invert Every element}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

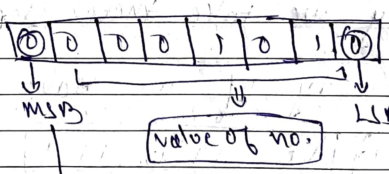
⇒ In each row - flip bit & swap

Negative Binary Numbers

⇒ Let's say we store a number in 1 byte

1 byte = 8 bits

10 =>



This tells us whether the no. is +ve or -ve

1 -> -ve

0 -> +ve

Steps 1. complement of no. & add 1 to it

also known as 2's complement

$(10)_{10} = (1010)_2$

$= (00001010)_2$

Step 1:- 11110101

Step 2:-

$$\begin{array}{r} 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\ +\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\ \hline 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0 \end{array}$$

(-10)

$(-10)_{10} \Rightarrow (11110110)_2$

Q. why we take these steps?

⇒ Let's say If we have to store

10 | 10 11 0 11 1 | this in 1 byte

This will be discarded as extra space is no there

=> Now, we know that if we subtract any number from 0, we will get -ve of that no. So,

Let's say we subtract 10 from 0

$$-\left[\begin{array}{r} 1\ 00000000 \\ 00000010 \\ \hline 0 \end{array}\right] \Rightarrow (1)$$

If we add one over here, it does not matter as it will be discarded

Now, let's look at a pattern

power of 2 =>
- $1000 = 111+1$
- $10000 = 1111+1$

Now in (1) we wer subtracting 10 from power of 0's

So, $100000000 - 00001010$

$\Rightarrow 11111111+1 - 00001010$

Rearrange

$\Rightarrow \underbrace{11111111 - 00001010}_{\text{0}} + 1 \quad (2)$

This will give us (10) complement of 10

=> As given in eq. (2) this is what exactly we were doing to get -ve of a no.

Range of Numbers

1 byte = | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Total no. of numbers = 2×2×... = $2^8 = 256$

Total 256 numbers we can store.

=> Now we now that MSB represents sign of no.

=> So the actual no. is stored in remaining 7 bits.

=> Now, total no. of numbers $= 2^7 = 128$

=> So, 128 on negative & 128 on positive

-128 to 128

=> Now lets count no. of numbers of negative side

-128 - 1 = [128]

=> On positive side

128 - 0 = 129

Now, 128 + 129 != 256

=> So, as zero is also positive we have to remove one no. from ⊕ve side which is 128

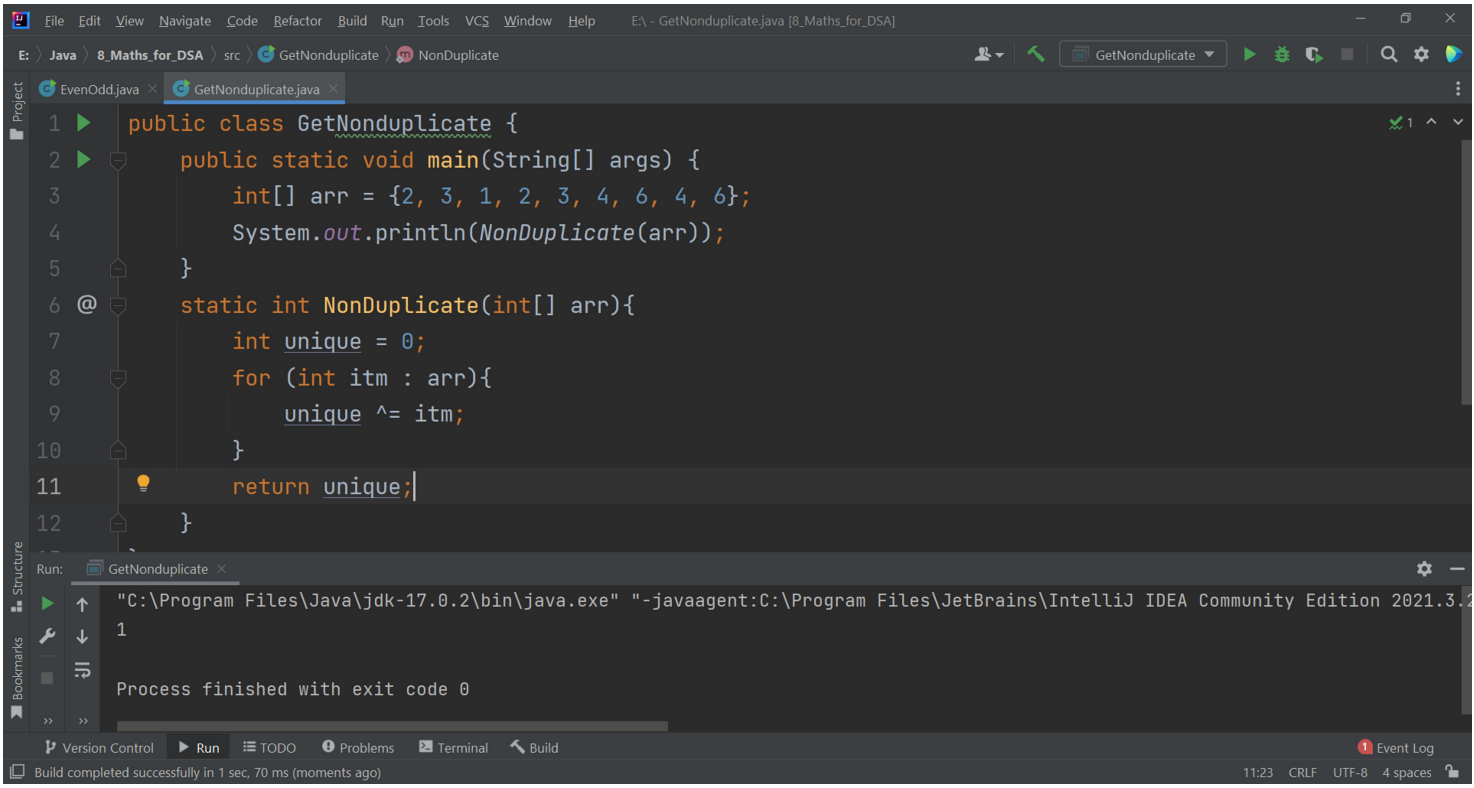
now [-128 to 127]

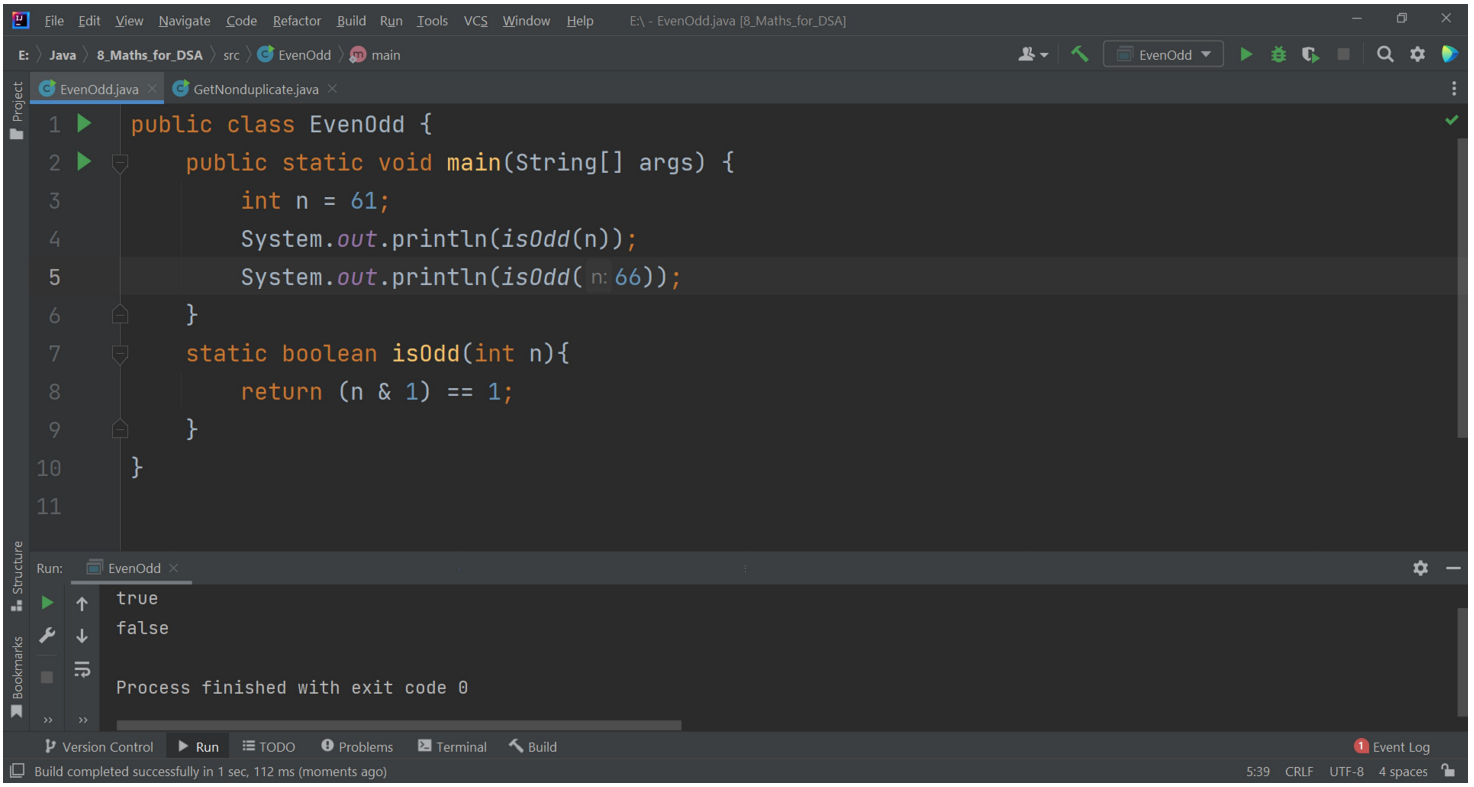
from this range we can store no. in 1 byte

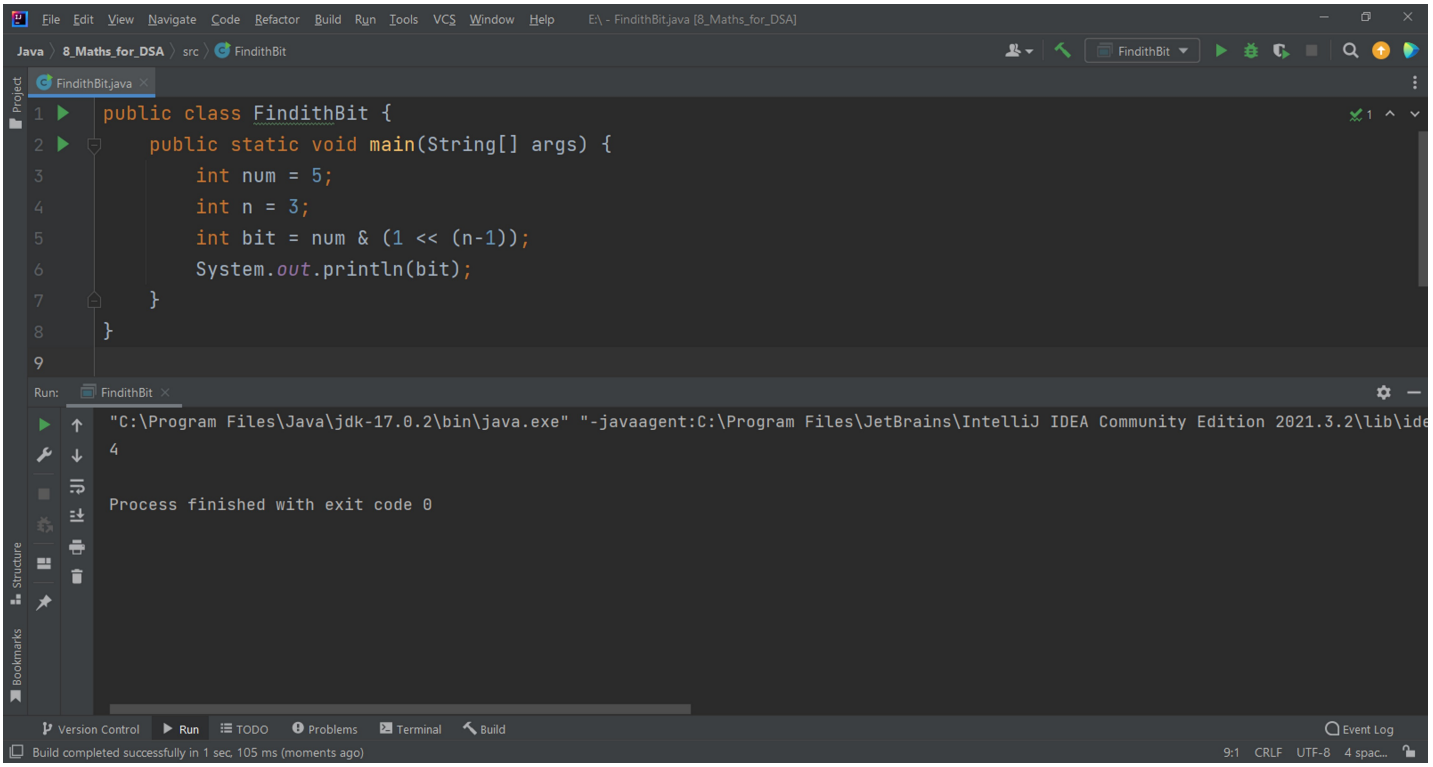
Formula to calculate Range

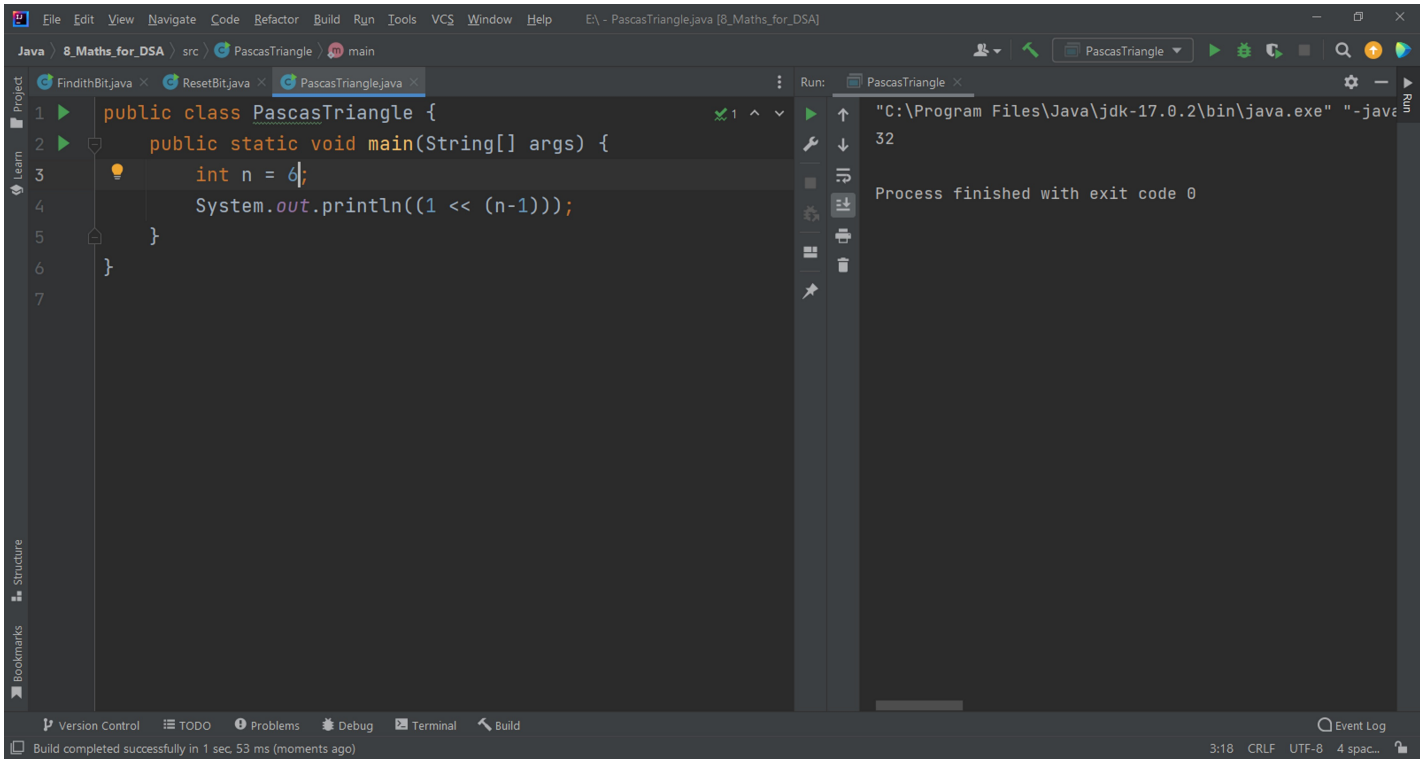
$$-2^{n-1} \text{ to } 2^{n-1} - 1$$

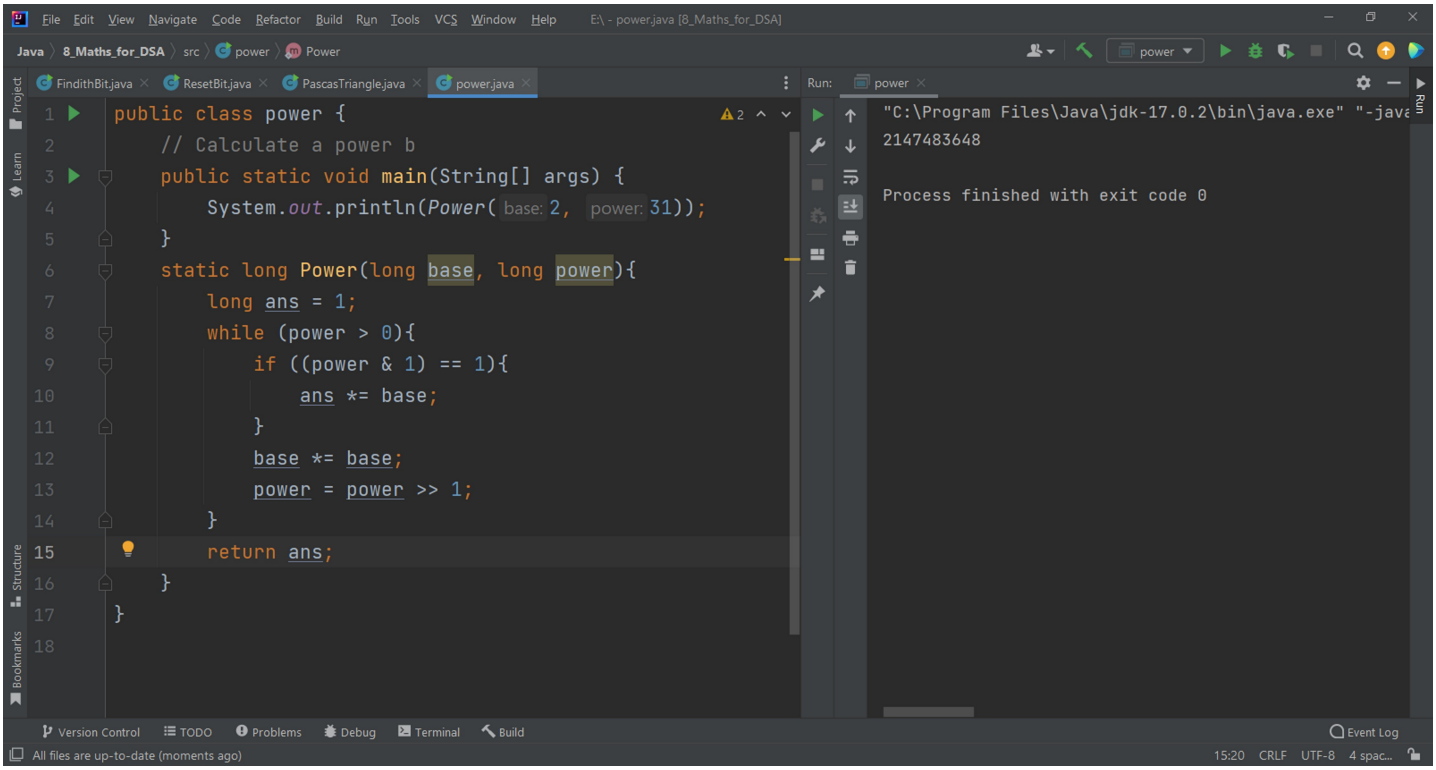
where n is no. of bits.


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
E:\ - GetNonduplicate.java [8_Maths_for_DSA]
E: Java 8_Maths_for_DSA src GetNonduplicate NonDuplicate
GetNonduplicate
EvenOdd.java
GetNonduplicate.java
public class GetNonduplicate {
public static void main(String[] args) {
int[] arr = {2, 3, 1, 2, 3, 4, 6, 4, 6};
System.out.println(NonDuplicate(arr));
}
static int NonDuplicate(int[] arr){
int unique = 0;
for (int itm : arr){
unique ^= itm;
}
return unique;
}
Run: GetNonduplicate
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2
1
Process finished with exit code 0
Version Control Run TODO Problems Terminal Build
Event Log
Build completed successfully in 1 sec, 70 ms (moments ago)
11:23 CRLF UTF-8 4 spaces


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
E:\ - EvenOdd.java [8_Maths_for_DSA]
E: Java 8_Maths_for_DSA src EvenOdd main
EvenOdd
EvenOdd.java
GetNonduplicate.java
public class EvenOdd {
public static void main(String[] args) {
int n = 61;
System.out.println(isOdd(n));
System.out.println(isOdd(n: 66));
}
static boolean isOdd(int n){
return (n & 1) == 1;
}
}
Run: EvenOdd
true
false
Process finished with exit code 0
Version Control Run TODO Problems Terminal Build
Event Log
Build completed successfully in 1 sec, 112 ms (moments ago)
5:39 CRLF UTF-8 4 spaces

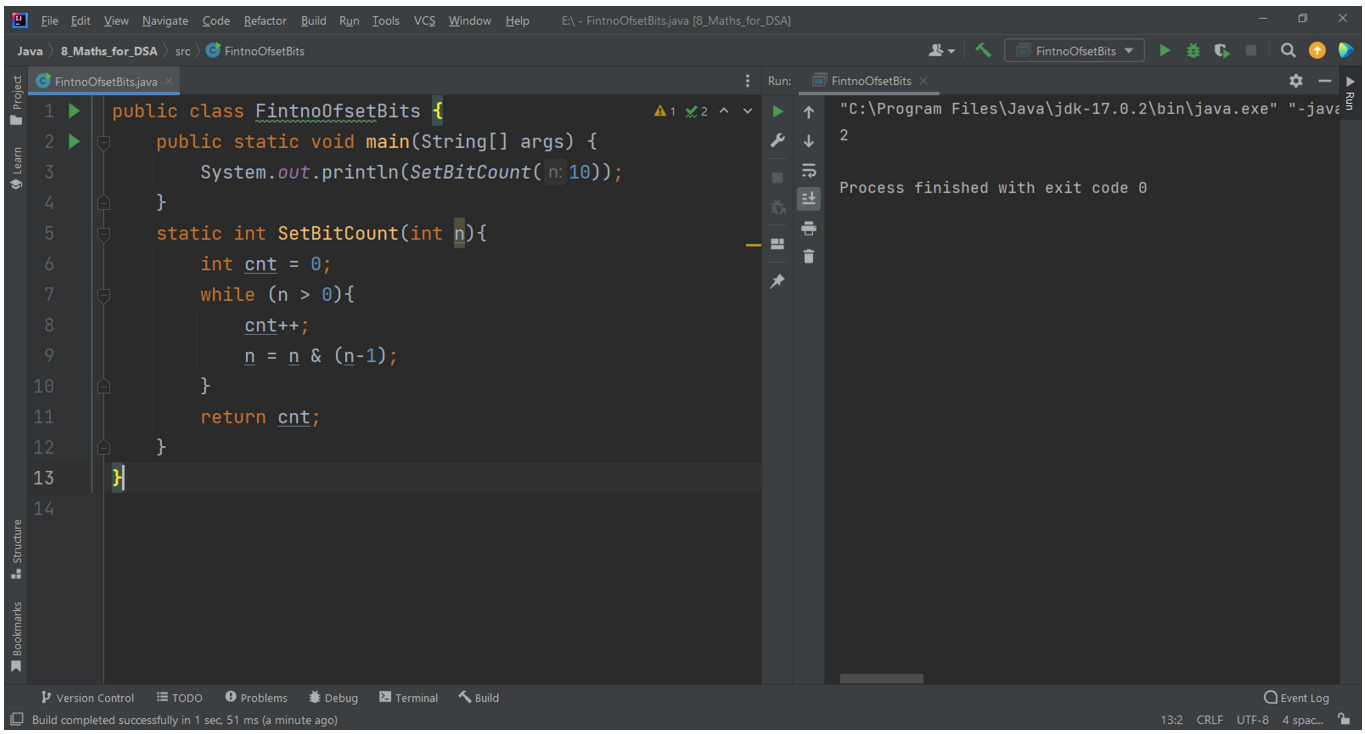

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
E:\ - FindithBit.java [8_Maths_for_DSA]
Java 8_Maths_for_DSA src FindithBit
FindithBit.java
public class FindithBit {
public static void main(String[] args) {
int num = 5;
int n = 3;
int bit = num & (1 << (n-1));
System.out.println(bit);
}
}
Run: FindithBit
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\ide
4
Process finished with exit code 0
Version Control Run TODO Problems Terminal Build
Event Log
Build completed successfully in 1 sec, 105 ms (moments ago)
9:1 CRLF UTF-8 4 spac...


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
E:\ - PascasTriangle.java [8_Maths_for_DSA]
Java 8_Maths_for_DSA src PascasTriangle main
FindithBit.java ResetBit.java PascasTriangle.java
public class PascasTriangle {
public static void main(String[] args) {
int n = 6;
System.out.println((1 << (n-1)));
}
}
Run: PascasTriangle
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-java
32
Process finished with exit code 0
Version Control TODO Problems Debug Terminal Build
Event Log
Build completed successfully in 1 sec, 53 ms (moments ago)
3:18 CRLF UTF-8 4 spac...



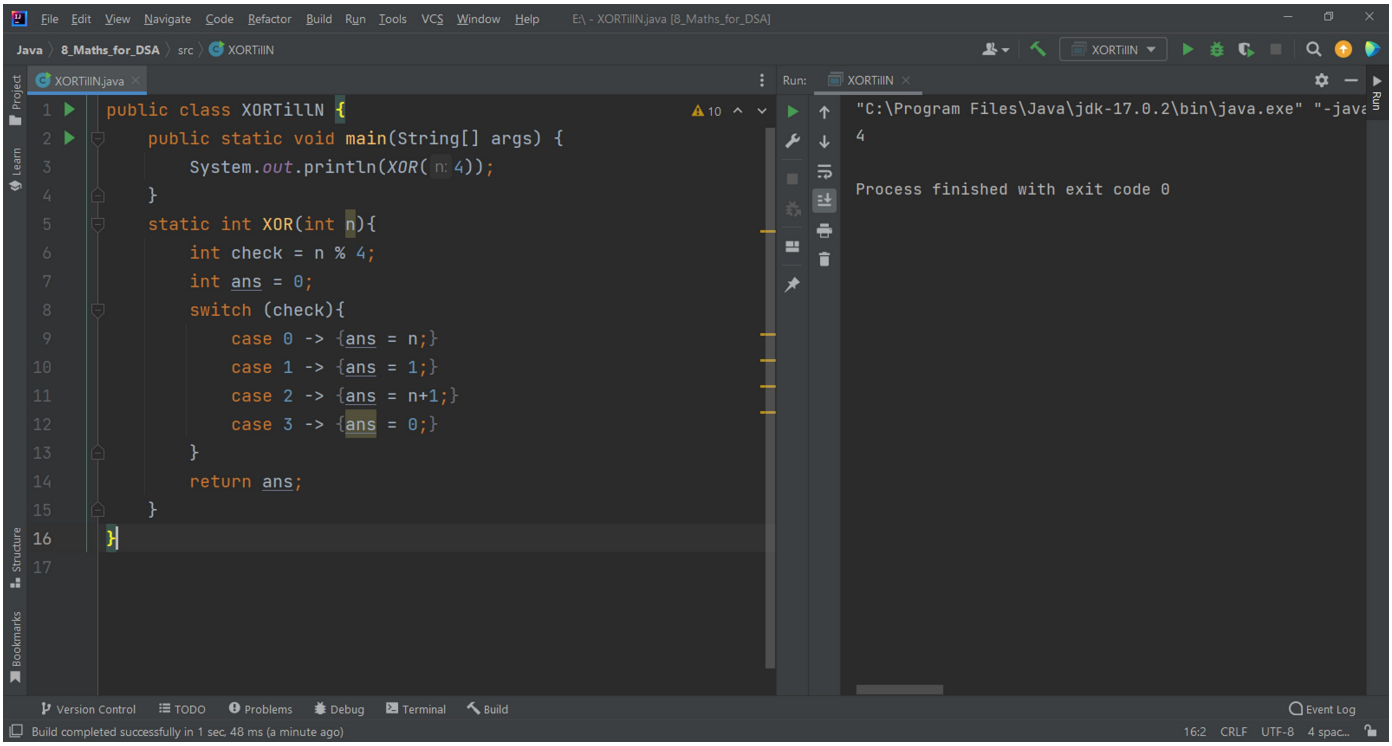
```java
public class power {
    // Calculate a power b
    public static void main(String[] args) {
        System.out.println(Power( base: 2,  power: 31));
    }
    static long Power(long base, long power){
        long ans = 1;
        while (power > 0){
            if ((power & 1) == 1){
                ans *= base;
            }
            base *= base;
            power = power >> 1;
        }
        return ans;
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-java
2147483648

Process finished with exit code 0
```



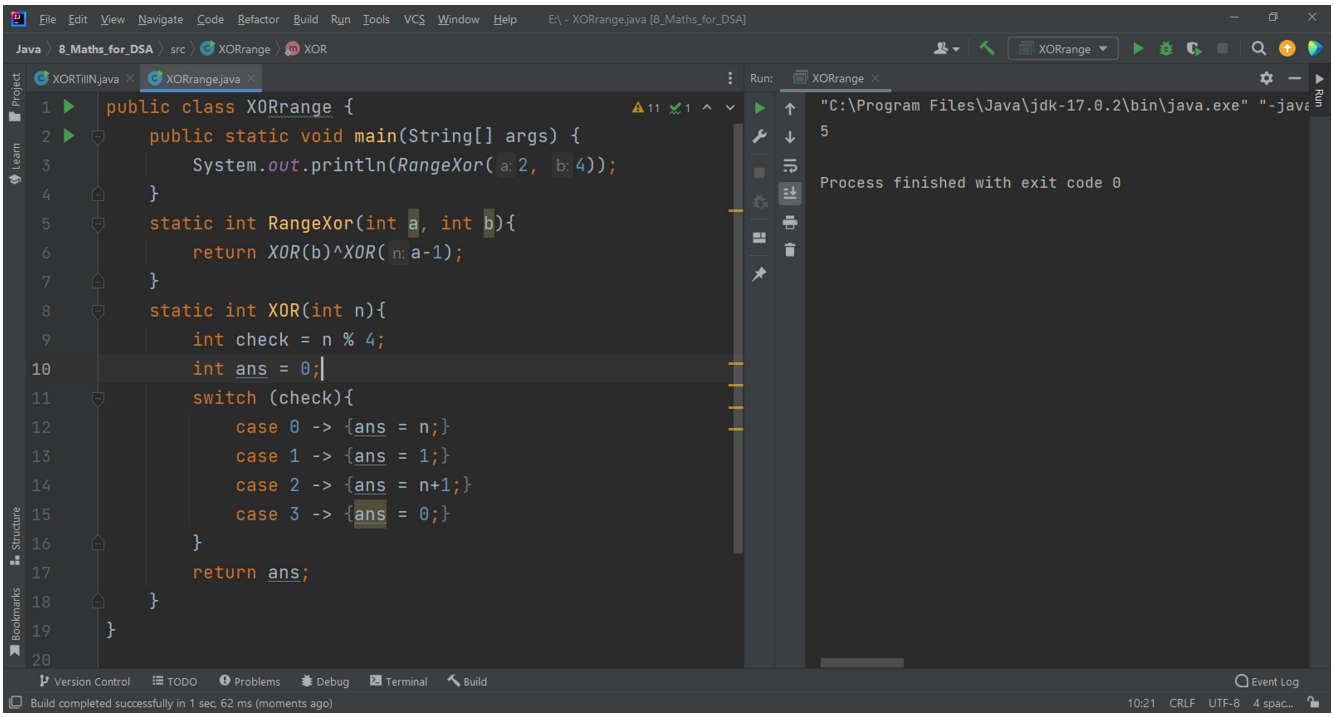
```java
public class FintnoOfsetBits {
    public static void main(String[] args) {
        System.out.println(SetBitCount( n: 10));
    }
    static int SetBitCount(int n){
        int cnt = 0;
        while (n > 0){
            cnt++;
            n = n & (n-1);
        }
        return cnt;
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-java
2

Process finished with exit code 0
```



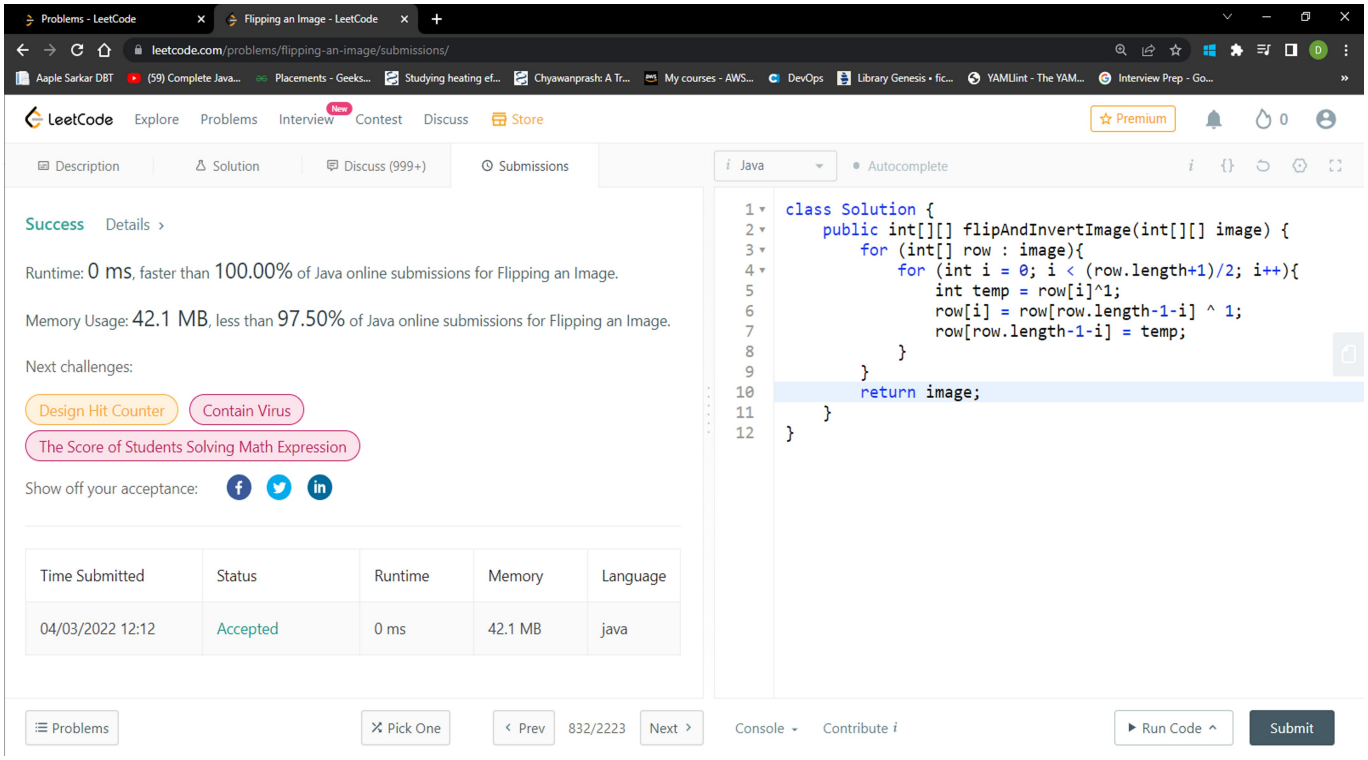
```java
public class XORTillN {
    public static void main(String[] args) {
        System.out.println(XOR( n: 4));
    }
    static int XOR(int n){
        int check = n % 4;
        int ans = 0;
        switch (check){
            case 0 -> {ans = n;}
            case 1 -> {ans = 1;}
            case 2 -> {ans = n+1;}
            case 3 -> {ans = 0;}
        }
        return ans;
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-java
4

Process finished with exit code 0
```



```java
public class XORrange {
    public static void main(String[] args) {
        System.out.println(RangeXor( a: 2,  b: 4));
    }
    static int RangeXor(int a, int b){
        return XOR(b)^XOR( n: a-1);
    }
    static int XOR(int n){
        int check = n % 4;
        int ans = 0;
        switch (check){
            case 0 -> {ans = n;}
            case 1 -> {ans = 1;}
            case 2 -> {ans = n+1;}
            case 3 -> {ans = 0;}
        }
        return ans;
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-java
5

Process finished with exit code 0
```


Problems - LeetCode
Flipping an Image - LeetCode
leetcode.com/problems/flipping-an-image/submissions/
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
Premium
Description
Solution
Discuss (999+)
Submissions
Java
Autocomplete
Success
Details ›
Runtime: 0 ms, faster than 100.00% of Java online submissions for Flipping an Image.
Memory Usage: 42.1 MB, less than 97.50% of Java online submissions for Flipping an Image.
Next challenges:
Design Hit Counter
Contain Virus
The Score of Students Solving Math Expression
Show off your acceptance:
Time Submitted
Status
Runtime
Memory
Language
04/03/2022 12:12
Accepted
0 ms
42.1 MB
java
class Solution {
public int[][] flipAndInvertImage(int[][] image) {
for (int[] row : image){
for (int i = 0; i < (row.length+1)/2; i++){
int temp = row[i]^1;
row[i] = row[row.length-1-i] ^ 1;
row[row.length-1-i] = temp;
}
}
return image;
}
}
Problems
Pick One
Prev
832/2223
Next
Console
Contribute i
Run Code
Submit