Bus Dilemma

Problem Description

A bus travels to **N** different stops, and at each stop some amount of people get in and get out. You are given an array **A**, where integer **A[i]** gives the amount of people who have got onto the bus (if positive), or the amount of people who have gotten off the bus (if negative) at the stop i.

You are also given a positive integer **B**, that denotes the capacity of the bus, that is the maximum number of people the bus can hold. The bus can hold **0** to **B** number of people at any time.

Initially, the bus can have some number of people inside of it, you have to find the total number of possible ways of how many people were initially in the bus before the first stop, such that at any time there are always **0** to **B** number of people in the bus.

If it is not possible to find any valid number of ways, return **0**.

$$\begin{array}{ccccc} & 0 & 1 & 2 & 3 \\ A = & [2 & 4 & -1 & 3] \\ PS = & 2 & 6 & 5 & 8 \end{array}$$

$B = 10$

min = 2 max = 8

lo = 0 hi = 2 (B − max)

ans = 3 (hi − lo + 1)

↳ 0, 1, 2

$$\begin{array}{ccccc} & 0 & 1 & 2 & 3 \\ A = & [-2 & -4 & 5 & 3] \\ PS = & -2 & -6 & -1 & 2 \end{array}$$

$B = 10$

min = −6 max = 2

lo = 6 (−min) hi = 8 (B − max)

ans = 3

↳ 6, 7, 8

$$A = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ [-2 & -4 & -1 & 2] \end{array}$$

$B = 10$

$PS = -2 \quad -6 \quad -7 \quad -5$

$min = -7 \qquad max = -2$

$lo = 7$ $(-min)$, $hi = 10$ (B)

ans = 4

↳ 7, 8, 9, 10

→ find max and min of prefix sum

```
lo = 0;
hi = 0;

if (min > 0) {
    lo = 0;
}
else {
    lo = -min;
}

if (max > 0) {
    hi = B - max;
}
else {
    hi = B;
}
```

ans ⇒ all values from lo to hi

when ans is 0 → no possible ways

A = [-2 3 -6 2]

ps = -2 1 -5 -3

B = 4

min = -5   max = 1

```
if (-min > B) {
        return 0;
}
```

A = [2 3 5 1]

ps = 2 5 10 11

B = 6

min = 2   max = 11

```
if (max > B) {
        return 0;
}
```

```java
public class Solution {
    public int solve(int[] A, int B) {
        //find max and min of prefix sum value
        long ps = 0;
        long min = A[0];
        long max = A[0];

        for(int i=0; i < A.length;i++) {
            ps = ps + A[i];

            if(ps > max) {
                max = ps;
            }
            if(ps < min) {
                min = ps;
            }
        }

        if(-min > B || max > B) {
            return 0;
        }

        long lo=0, hi=0;

        if(min > 0) {
            lo = 0;
        }
        else {
            lo = -min;
        }

        if(max > 0) {
            hi = B - max;
        }
        else {
            hi = B;
        }

        int count = (int)(hi-lo+1);
        return count;

    }
}
```

Q.2 Boundary traversal

Return clockwise boundary traversal for a N x M matrix.

$n = 4, \quad m = 5$

	0	1	2	3	4
0	5	9	7	4	8
1	1	3	6	2	3
2	10	4	15	6	12
3	12	0	8	41	39

$m-1$ values left to right

$n-1$ values top to bottom

$m-1$ values right to left

$n-1$ values bottom to top

len of ans array $= \boxed{m-1 + n-1 + m-1 + n-1}$

$\cancel{2}m + \cancel{2}n - \cancel{4}$

TC : $O(n+m)$

```java
public class Solution {
    public int[] solve(int[][] A) {
        int n = A.length;
        int m = A[0].length;
        int len = m-1 + n-1 + m-1 + n-1;
        int[]ans = new int[len];
        int k = 0; //to help me fill values in ans array

        int i=0,j=0;

        //m-1 values left to right
        for(int c=1; c <= m-1;c++) {
            ans[k] = A[i][j];
            j++;
            k++;
        }

        //n-1 values top to bottom
        for(int c=1; c <= n-1;c++) {
            ans[k] = A[i][j];
            i++;
            k++;
        }

        //m-1 values right to left
         for(int c=1; c <= m-1;c++) {
            ans[k] = A[i][j];
            j--;
            k++;
        }

        //n-1 values bottom to top
        for(int c=1; c <= n-1;c++) {
            ans[k] = A[i][j];
            i--;
            k++;
        }

        return ans;
    }
}
```

Q. 3 Create a binary number with A 1's followed by B 0's and return decimal equivalent.

$A = 3 \quad B = 2$

$i = 2$ to $i = 4$

$$\begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array} \longrightarrow 28$$

ans :- $2^2 + 2^3 + 2^4$

$A = 2 \quad B = 3$

$i = 3$ to $i = 4$

$$\begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{array} \longrightarrow 24$$

ans : $2^3 + 2^4$

loop : $i = B$ to $i < A+B$

ans += 2^i

$2^i \Rightarrow 1 << i$

```java
public class Solution {
    public int solve(int A, int B) {
        int ans = 0;

        for(int i=B; i < A + B;i++) {
            ans = ans + (1 << i); //ans = ans + 2 pow i
        }

        return ans;
    }
}
```

itr : A