

Doubt class

if greatest <= (l, l) sum = cl
if widest <= (w, w) sum = pr

① Power of 2

M-1 :-

$$1 \rightarrow 1$$

$$2 \rightarrow 10$$

$$4 \rightarrow 100$$

$$8 \rightarrow 1000$$

$$16 \rightarrow 10000$$

Subtracting 1
from each

$$\begin{array}{rcl} 0 & \rightarrow & 0 \\ 1 & \rightarrow & 1 \\ 3 & \rightarrow & 11 \\ 7 & \rightarrow & 111 \\ 15 & \rightarrow & 1111 \end{array}$$

Ex: $4 \& 3 \Rightarrow$

011
<u>000</u>

∴ 4 is a power of 2.

if $[n \& (n-1)] == 0 \Rightarrow$ then that particular no is power of 2

→ Powers of 2 have only one '1' in binary.

→ subtracting 1 flips all the bits after that 1.

→ Adding AND(&) ing them becomes zero.

T.C = O(1)

Ex: 9 & 8

$$\begin{array}{r} 1000 \\ 1001 \\ \hline 1000 \end{array}$$

∴ 9 & 8 has been given $\Rightarrow \neq 0 \therefore$ Not a power of 2.

M-2: while ($n \% 2 == 0$)

$n = n/2;$

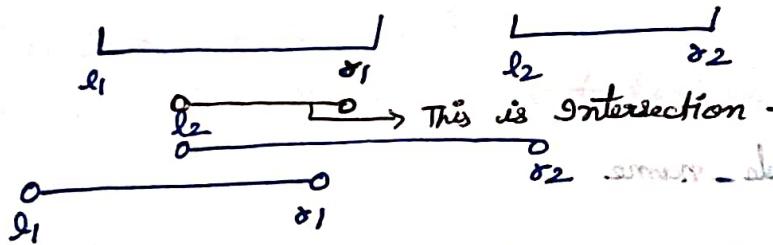
T.C: $O(\log N)$

if ($n == 1$) return 1;
return 0;

M-3: if ($(n/2) \% 2 == 0$) \rightarrow This is Wrong approach

M-4: If the binary representation of the number has only one '1', then it is a power of 2.

2. Two Intervals :-



$l_2 = \max(l_1, l_2) \rightarrow$ starting pt

$s_1 = \min(s_1, s_2) \rightarrow$ Ending pt

if (starting <= ending)

 intersection exist

else

 intersection does not exist

$0 \leftarrow 0$
 $1 \leftarrow 1$
 $11 \leftarrow 2$
 $111 \leftarrow 5$
 $1111 \leftarrow 21$

$\xleftarrow{\text{↓ Euclidean}}$

$01 \leftarrow 1$
 $001 \leftarrow 2$
 $0001 \leftarrow 8$
 $00001 \leftarrow 31$

3. GCD

M-1: Use in-built STL funcn: $\text{gcd}(a, b) = [(a \& b) \oplus \dots]$

M-2: Use Euclidean GCD Algorithm

4. Lucky Numbers:-

 A no. consisting only of 4 & 7

Traverse every digit of the no. using modulo 10 and check if the digits are 4 & 7 only and stop if any other no. found.

Code :-

```

int lucky (int x) {
    while (x) {
        int digit = x % 10;
        if (digit != 4 && digit != 7) return 0;
        x /= 10;
    }
    return 1;
}

```

void solve (int a, int b) {

```

int a, b;
cin >> a >> b;
for (int i = a; i <= b; i++) {
    if (lucky (i)) {
        cout << i << " ";
    }
}

```

5. sum of odd no. b/w X & Y :-

If x is even \rightarrow start = $x + 1$
If x is odd \rightarrow start = $x + 2$

$\begin{cases} \text{Even} & \rightarrow \text{Ending} = Y - 1 \\ \text{odd} & \rightarrow \text{Ending} = Y - 2 \end{cases}$

$$S_n = \frac{m}{2} (a+l) \rightarrow AP$$

$\rightarrow \text{start} \leq \text{Ending}$

Code :-

```

int x, y;
cin >> x >> y;
int starting = min (x, y); ending = max (x, y);
if (x % 2) starting += 2;
else starting += 1;
if (y % 2) ending -= 2;
else ending -= 1;

```

T.C : O(1)

if (starting <= ending) {

int m = ((ending - starting) / 2) + 1;

int ans = (m * (starting + ending)) / 2;

cout << ans << endl;

}
else {

cout << 0 << endl;

}

3

6: Three Numbers:-

$$x+y+z = s$$

$$0 \leq x, y, z \leq k$$

$$x+y = s-z$$

$$x+y = T$$

Range of T: $s-k \leq T \leq s \approx 0 \leq T \leq 2k$

Ex: $K = 8$

$$x+y = 7$$

$$0+7$$

$$1+6$$

$$2+5$$

$$3+4$$

$$4+3$$

$$5+2$$

$$6+1$$

$$7+0$$

$$x+y = 3$$

$$0+3$$

$$1+2$$

$$2+1$$

$$3+0$$

$$4+0$$

$$5+0$$

$$6+0$$

$$7+0$$

$$8+0$$

$$9+0$$

$$10+0$$

$$11+0$$

$$12+0$$

$$13+0$$

$$14+0$$

$$15+0$$

$$16+0$$

$$17+0$$

$$18+0$$

$$19+0$$

$$20+0$$

$$21+0$$

$$22+0$$

$$23+0$$

$$24+0$$

$$25+0$$

$$26+0$$

$$27+0$$

$$28+0$$

$$29+0$$

$$30+0$$

$$31+0$$

$$32+0$$

$$33+0$$

$$34+0$$

$$35+0$$

$$36+0$$

$$37+0$$

$$38+0$$

$$39+0$$

$$40+0$$

$$41+0$$

$$42+0$$

$$43+0$$

$$44+0$$

$$45+0$$

$$46+0$$

$$47+0$$

$$48+0$$

$$49+0$$

$$50+0$$

$$51+0$$

$$52+0$$

$$53+0$$

$$54+0$$

$$55+0$$

$$56+0$$

$$57+0$$

$$58+0$$

$$59+0$$

$$60+0$$

$$61+0$$

$$62+0$$

$$63+0$$

$$64+0$$

$$65+0$$

$$66+0$$

$$67+0$$

$$68+0$$

$$69+0$$

$$70+0$$

$$71+0$$

$$72+0$$

$$73+0$$

$$74+0$$

$$75+0$$

$$76+0$$

$$77+0$$

$$78+0$$

$$79+0$$

$$80+0$$

$$81+0$$

$$82+0$$

$$83+0$$

$$84+0$$

$$85+0$$

$$86+0$$

$$87+0$$

$$88+0$$

$$89+0$$

$$90+0$$

$$91+0$$

$$92+0$$

$$93+0$$

$$94+0$$

$$95+0$$

$$96+0$$

$$97+0$$

$$98+0$$

$$99+0$$

$$100+0$$

$$101+0$$

$$102+0$$

$$103+0$$

$$104+0$$

$$105+0$$

$$106+0$$

$$107+0$$

$$108+0$$

$$109+0$$

$$110+0$$

$$111+0$$

$$112+0$$

$$113+0$$

$$114+0$$

$$115+0$$

$$116+0$$

$$117+0$$

$$118+0$$

$$119+0$$

$$120+0$$

$$121+0$$

$$122+0$$

$$123+0$$

$$124+0$$

$$125+0$$

$$126+0$$

$$127+0$$

$$128+0$$

$$129+0$$

$$130+0$$

$$131+0$$

$$132+0$$

$$133+0$$

$$134+0$$

$$135+0$$

$$136+0$$

$$137+0$$

$$138+0$$

$$139+0$$

$$140+0$$

$$141+0$$

$$142+0$$

$$143+0$$

$$144+0$$

$$145+0$$

$$146+0$$

$$147+0$$

$$148+0$$

$$149+0$$

$$150+0$$

$$151+0$$

$$152+0$$

$$153+0$$

$$154+0$$

$$155+0$$

$$156+0$$

$$157+0$$

$$158+0$$

$$159+0$$

$$160+0$$

$$161+0$$

$$162+0$$

$$163+0$$

$$164+0$$

$$165+0$$

$$166+0$$

$$167+0$$

$$168+0$$

$$169+0$$

$$170+0$$

$$171+0$$

$$172+0$$

$$173+0$$

$$174+0$$

$$175+0$$

$$176+0$$

$$177+0$$

$$178+0$$

$$179+0$$

$$180+0$$

$$181+0$$

$$182+0$$

$$183+0$$

$$184+0$$

$$185+0$$

$$186+0$$

$$187+0$$

$$188+0$$

$$189+0$$

$$190+0$$

$$191+0$$

$$192+0$$

$$193+0$$

$$194+0$$

$$195+0$$

$$196+0$$

$$197+0$$

$$198+0$$

$$199+0$$

$$200+0$$

$$201+0$$

$$202+0$$

$$203+0$$

$$204+0$$

$$205+0$$

$$206+0$$

$$207+0$$

$$208+0$$

$$209+0$$

$$210+0$$

$$211+0$$

$$212+0$$

$$213+0$$

$$214+0$$

$$215+0$$

$$216+0$$

$$217+0$$

$$218+0$$

$$219+0$$

$$220+0$$

$$221+0$$

$$222+0$$

Code :-

```

void solve () {
    int k, s;
    cin >> k >> s;
    /*
        x+y+z = s;
        0 <= s <= 3k
        x+y = s-z;
        x+y = T
        max(0, s-k) <= s-z <= s
    */
    int ans = 0;
    for (int i = max((int)0, s-k); i <= s; i++) {
        if (i <= k)
            ans += i+1;
        else if (i <= 2*k)
            int d = i-k;
            ans += (k-d+1);
        }
    cout << ans << endl;
}

```

T.C : O(K)

7. Another pyramid Problem :-

code :-

```

void solve () {
    int n;
    cin >> n;
    int row = n;
    int col = (2*n)-1;
    for (int i = 1; i <= row; i++) {
        for (int blank = 0; blank < n-i; blank++)
            cout << "*";
        for (int num = 1; num <= i; num++)
            cout << num << " ";
        for (int num = i-1; num >= 1; num--)
            cout << num << " ";
        for (int blank = 0; blank < n-i; blank++)
            cout << "*";
        cout << endl;
}

```

row = n

col = (2 * n) - 1

space to print for every row = n - i