Q. Take 'n' Integer from user & print the following series:-

(i) 1 2 3 . . . . N

(ii) N N-1 N-2 1

(iii) 1 3 5 . . . . . odd no. ≤ N

(iv) 2 4 6 . . . . . even no. ≤ N

(v) 0 1 1 2 3 5 . . . . last fib. no. ≤ N → fibonacci series

(vi) 1 2 4 8 16 . . . . $2^k \le N$

I/P ← 8

O/P → 1 2 3 4 5 6 7 8

8 7 6 5 4 3 2 1

1 3 5 7

2 4 6 8

0 1 1 2 3 5 8

1 2 4 8

fibonacci series:—

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

$fib(n) = fib(n-1) + f(n-2)$ $n > 2$ ← recurrence relation

$$\left.\begin{array}{l} fib(1) = 0 \\ fib(2) = 1 \end{array}\right\}$$ ← Base Cases

$curr = fp + sp$

$3 = 1 + 2$

$fp = sp$

$sp = curr$

$curr = 3$

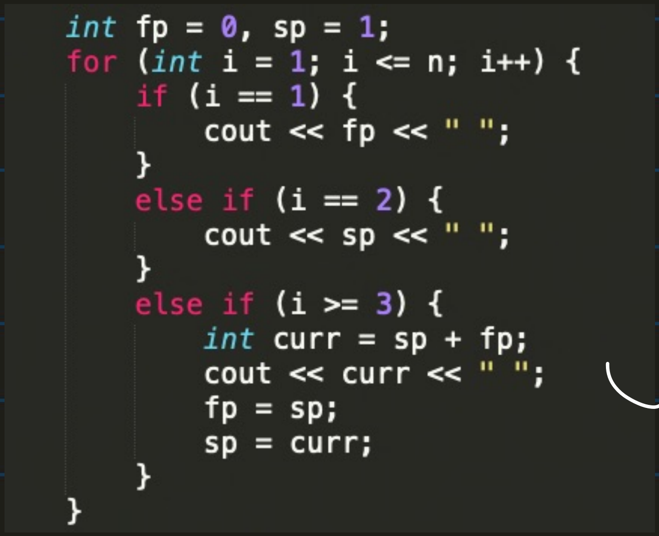
$fp = \cancel{1}$

$sp = \cancel{2}$

2

3

fibonacci code



```
int fp = 0, sp = 1;
for (int i = 1; i <= n; i++) {
    if (i == 1) {
        cout << fp << " ";
    }
    else if (i == 2) {
        cout << sp << " ";
    }
    else if (i >= 3) {
        int curr = sp + fp;
        cout << curr << " ";
        fp = sp;
        sp = curr;
    }
}
```

O(n)

n = 0

×

n = 1

0

n = 2

0 1

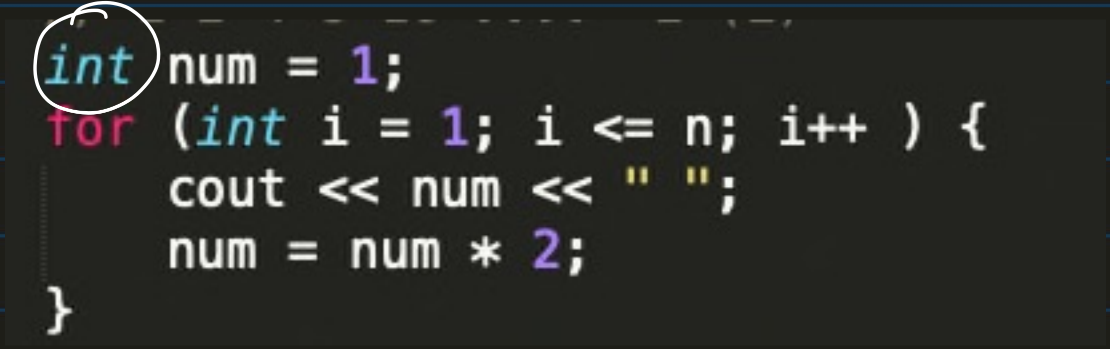
n = ~~4~~ 5

0 1 1 2 3

fp	sp	curr
~~0~~	~~1~~	3
~~1~~	~~1~~	
~~1~~	~~2~~	
2	3	

1 2 4 8 16 . . . . ,

2^0 2^1 2^2 2^3 . 2^4 . . . . .



```
int num = 1;
for (int i = 1; i <= n; i++ ) {
    cout << num << " ";
    num = num * 2;
}
```

O/P

n=5

num

~~1~~

~~2~~

4

1 2 4 8 16

$\simeq \boxed{10^9}$

$2^{100} \simeq 10^{30}$

Important method to generate powers of 2 : ★ ★

num = 1 = 2^0

bin ⟶ 0 0 0 0 0 1 ← 0

num = num << 1 ⟶ 0 0 0 0 1 0 ← 0 ⟶ 2 ⟶ 2^1

num = num << 1 ⟶ 0 0 0 1 0 0 ⟶ 2^2 ⟶ 4

Q You have to take n as i/p & print table of it .

n=10

10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

n=11

11 x 1 = 11
11 x 2 = 22
11 x 3 = 33
11 x 4 = 44
11 x 5 = 55
11 x 6 = 66
11 x 7 = 77
11 x 8 = 88
11 x 9 = 99
11 x 10 = 110

Q Program to take n as i/p & display following things

(i) no. of even terms

(ii) sum of even terms

(iii) no. of odd terms

(iv) sum of odd terms

→ 1 loop only

$0 \le n \le 10^{12}$

I/P: $n \leftarrow 8$

O/P: even terms: 5

even sum: 20

odd terms: 4

odd sum: 16