```
for (k=1; k<=n; k*=2)
{ for (j=1; j<=k; j++)
    { ...... }
}
```

Question:
time complexity
$= O(??)$

Solve:

iteration	k = outer loop	j = inner loop (at most)
0	1	$1 = 2^0$
1	2	$2 = 2^1$
2	4	$4 = 2^2$
3	8	$8 = 2^3$
4	16	$16 = 2^4$
⋮	⋮	⋮
i	n	$n = 2^i$

"generally, $2^{\text{iteration}}$ inner loops running at each step."

Total running time.

$T(n) = 1+2+4+8+16+ \ldots\ldots + n$

$\Rightarrow 2^0+2^1+2^2+2^3+2^4+ \ldots\ldots + 2^i$] geometric series

index = 0 1 2 3 4 i

how many terms here?

$0 \sim i \Rightarrow$ total $(i+1)$ number of terms

geometric sum formula for summing upto m, where $r>1$

$$r^0+r^1+r^2+\ldots+r^k \Rightarrow \sum_{k=0}^{m} r^k = \frac{r^{m+1}-1}{r-1}$$

so, $2^0+2^1+2^2+2^3+2^4+ \ldots\ldots + 2^i \rightarrow r=2$

$$\Rightarrow \sum_{k=0}^{i} r^k = \frac{r^{i+1}-1}{r-1}$$

$n = 2^i$

$$\Rightarrow \sum_{k=0}^{i} 2^k = \frac{2^{i+1} - 1}{2-1}$$

$$= \cancel{\frac{2^{i+1}-1}{1}}$$

$$= 2^{i+1} - 1$$

$$= 2^i . 2 - 1$$

$$= n . 2 - 1 \quad [\because n = 2^i]$$

$$\sum_{k=0}^{i} 2^k = 2n - 1$$

Now,

$T(n)$ = total running time complexity

$$T(n) = O(2n-1)$$

$$\boxed{T(n) = O(n)}$$

So,

```
for (k=1; k≤n ; k++)
  { for (j=1 ; j <= k ; j++)
     {
```

total time complexity $= O(n)$