① A()
{
i=1; s=1;
While (s<=n)
{ i++;
s=s+i;
Print ("ravi");
}
}

i	1	2	3	4	5	6	7	... k
s	1	3	6	10	15	21	28	... n

$n = \frac{k(k+1)}{2}$

Hint what is 28 ?

$28 = 1+2+3+4+5+6+7$

↑ sum of n natural Number

$\frac{k^2+k}{2} = n$

$k^2 = n$

$(k = \sqrt{n})$

② A()
{
i=1
for (i=1; i^2<=n; i++)
{
print ("aman");
}

$i^2 = n$

$i = \sqrt{n}$

$O(\sqrt{n})$

③ A()

```
{ int i, j, k, n;
  for (i=1; i<=n; i++)
  {
    for (j=1; j<=i; j++)
    {
      for (k=1; k<=100; k++)
      {
        printf ("ravindra");
      }
    }
  }
```

Hint first loop contain i, > dependent
Second loop contain j. & i both
third loop contain only k

i = 1	i = 2	i = 3
j = 1 time	j = 2 time	j = 3 time
k = 100 time	k = 2*100	k = 3*100

1*100 + 2*100 + 3*100 ... + n*100

100 (1 + 2 + 3 + 4 ... n)

100 $\frac{n(n+1)}{2}$ = $O(n^2)$

(4)

```
A()
{
  int i,j,k,n;
  for (i=1; i<=n ;i++)
   {
    for(j=1; j<=i^2 ;j++)
    {
      for (k=1; k<=n/2 ; k++)
        {
          Print("Aman");
        }
    }
   }
```

1st for loop = i only → dependent

2nd for loop = i & j Both

3rd for loop = k

i = 1	i=2	i=3 ———	i=n
j = 1	j=4	j=9	j=n3
k = n/2	k=4*n/2	k=9*n/2	k=n^3 * n/2

$O(n^4)$

(5) A()
{ int i, j, k;
for (i = n/2 ; i <= n ; i++) → n/2 times
{
for (j = 1 ; j <= n/2 ; j++) → n/2 times
{ for (k = 1 ; k <= n ; k = k * 2) → $\log_2 n$ times
{
printf ("I Amar");
} } }

1st for loop → i only
2nd for loop → j only
3rd for loop → k only
No dependancy

$= n/2 \times n/2 \times \log_2 n$

$= O(n^2 \log_2 n)$

(6) A()
{
int i, j, k;
for (i = n/2 ; i <= n ; i++) → n/2
for (j = 1 ; j <= n ; j = 2 * j) → $\log_2 n$
for (k = 1 ; k <= n ; k = k * n) → $\log_2 n$
{
printf ("Amar");
}
}

$= n/2 \times \log_2 n \times \log_2 n$

$= n/2 \times (\log_2 n)^2$

$= O(n \log_2 n)^2$ — Ans