1.

```
int arr[] = {4,5,6};   //1
int n = arr.length;    //1 , n = 3
int sum = 0;           //1 , sum = 0 // sum = 4 // sum = 9 // sum = 15
for(int i =0 ; i<n; i++){  //1, i = 0;i<3 // 2, i=1;i<3 //3,i=2;i<3//4 -- n+1
sum = sum + arr[i];      // 1, sum = 4 //2, sum = 9 //3,sum = 15  -- n
}
```

so consider : **2n + 4** and remove numbers remain as : n

so **o(n)**

2.

```
for(int i =0;i<n;i++){       //n+1  //inside n+1 that will execute n times
for(int j=0;j<n;j++){         // n*n+1   //inside n+1 that will execute n times
system.out.println("*");    //n*n
}
system.out.println();    //n
}
```

So consider as **$O(n^2)$.**

3.

```
for(int i=0; i< n ;i++){             // n+1
      for(int j=0;j<n; j++){         //n*(n+1)
                c[i][j] = 0;             //n*n
for(int k =0;k <n; k++){                //n*n*(n+1)
        c[i][j]= c[i][j] + a[i][k]*b[k][j];    //n*n*n
}
}
}
```

So consider as **$2n+3n^2+n^3$ = $o(n^3)$**

4.

```
for(int i=0; i< n ;i++){      //n+1   , n=5
      for(int j=0;j<=i; j++){   //n(n+1)/2*n
              s.o.p("*");     //n(n+1)/2
       }
s.o.p();              //n
}
```

i	j		no of times j executed	
0	0	+1(false)	1	+1
1	0,1	+1	2	+1
2	0,1,2	+1	3	+1
3	0,1,2,3	+1	4	+1
4	0,1,2,3,4	+1	5	+1

So consider(1+2+3+4+5)+5

As **$n(n+1)/2+n$ = $4n + n^2$ = $o(n^2)$**