Practical - 1

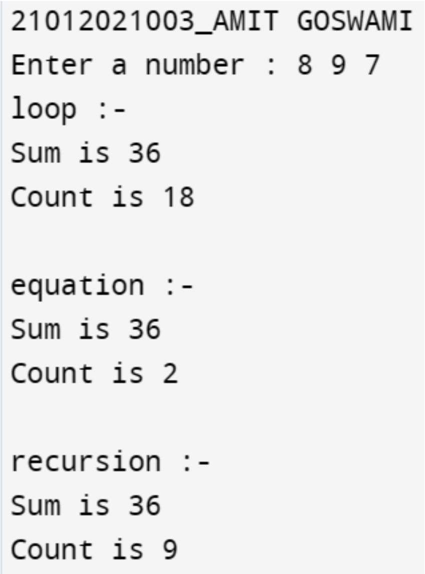
Implement a function for each of following problems and count the number of steps executed/Time taken by each function on various inputs and write complexity of each function. Also draw a comparative chart. In each of the following function N will be passed by user.

1. To calculate sum of 1 to N number using loop.
2. To calculate sum of 1 to N number using equation.
3. To calculate sum of 1 to N numbers using recursion.

Code :

```
#include<stdio.h>
int count_loop=0;
int count_eq=0;
int count_rec=0;
void sum_loop(int n){
int i,sum=0;
count_loop++;
for(i=1;i<=n;i++){
count_loop++;
sum+=i;
count_loop++;
}
count_loop++;
printf("\n\nloop :-");
printf("\nSum is %d",sum);
printf("\nCount is %d",count_loop);
}
void sum_eq(int n){
int sum;
count_eq++;
sum=(n*(n+1))/2;
count_eq++;
printf("\n\nequation :-");
printf("\nSum is %d",sum);
printf("\nCount is %d",count_eq);
}
int sum_rec(int n){
if(n<=1){
count_rec++;
count_rec++;
return n;
}
else{
count_rec++;
return n+sum_rec(n-1);
}
}
void main(){
int n,sum;
printf("21012021003_AMIT GOSWAMI");
printf("Enter a number : ");
scanf("%d",&n);
sum_loop(n);
sum_eq(n);
sum=sum_rec(n);
printf("\n\nrecursion :-");
printf("\nSum is %d",sum);
printf("\nCount is %d",count_rec);
}
```

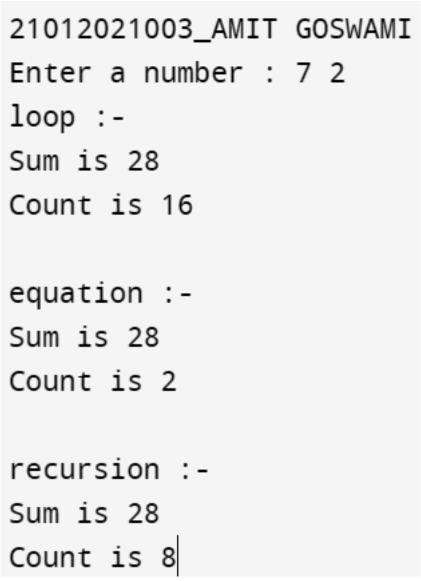
Output:



```
21012021003_AMIT GOSWAMI
Enter a number : 8 9 7
loop :-
Sum is 36
Count is 18

equation :-
Sum is 36
Count is 2

recursion :-
Sum is 36
Count is 9
```



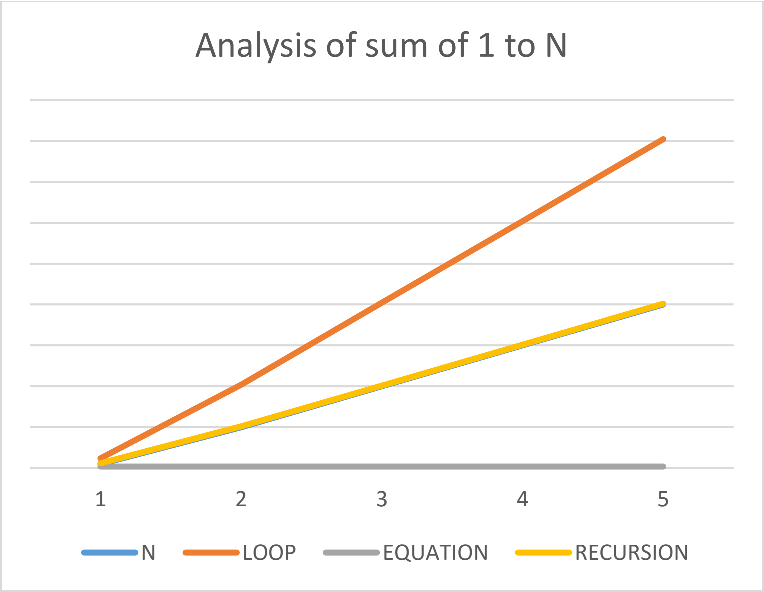
```
21012021003_AMIT GOSWAMI
Enter a number : 7 2
loop :-
Sum is 28
Count is 16

equation :-
Sum is 28
Count is 2

recursion :-
Sum is 28
Count is 8
```

Analysis:

N	LOOP	EQUATION	RECURSION
5	12	2	6
50	102	2	51
100	202	2	101
150	302	2	151
200	402	2	201



Conclusion:

After analyzing the table we can conclude that equation algorithm gives constant value of count hence it is the best case, recursion algorithm is half of loop algorithm so it is average and loop algorithm is the worst.