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\_rec=0;

int count\_eq=0;

void loop(int ); int rec(int ); void eq(int );

int main()

{ int n;

printf("enter value of n =");

scanf("%d",&n);

loop(n); eq(n);

int recursion=rec(n);

printf("sum of rec=%d\n",recursion);

printf("rec count=%d",count\_rec);

return 0;

}

void loop(int n){ int

sum\_loop=0; for(int i=0;i<=n;i++){ sum\_loop=sum\_loop+i;

count\_loop++;

}

printf("LOOP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"); printf("sum of loop=%d\n",sum\_loop); printf("loop

count=%d\n",count\_loop);

}

void eq(int n){ int sum\_eq=0; count\_eq; sum\_eq=(n\*(n+1))/2; count\_eq++; printf("\nEQUATION\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"); printf("sum of eq=%d\n",sum\_eq); printf("eq count=%d\n",count\_eq); printf("\nRECUSRION\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

int rec(int n){ if (n <=

1){

count\_rec++; return n;

} else {

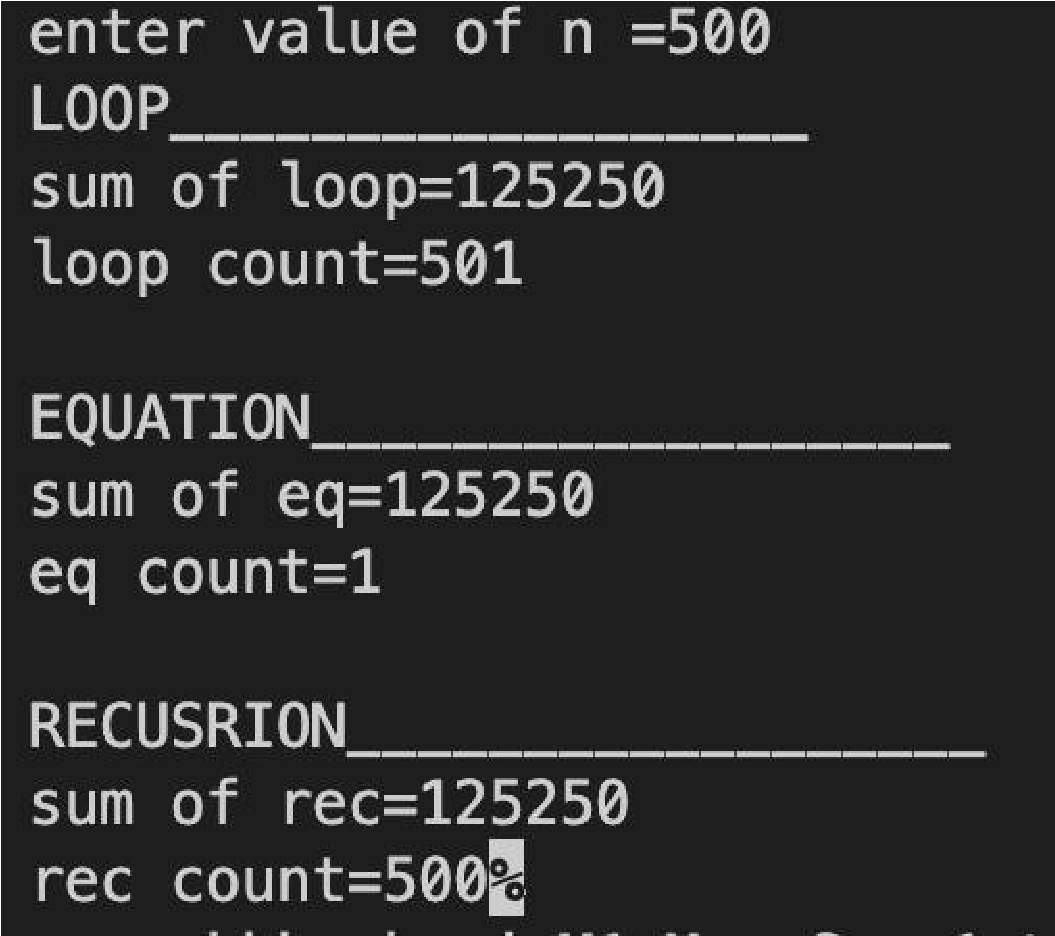
count\_rec++; return

n + rec(n - 1);

}

}

OUTPUT:



Graph :

