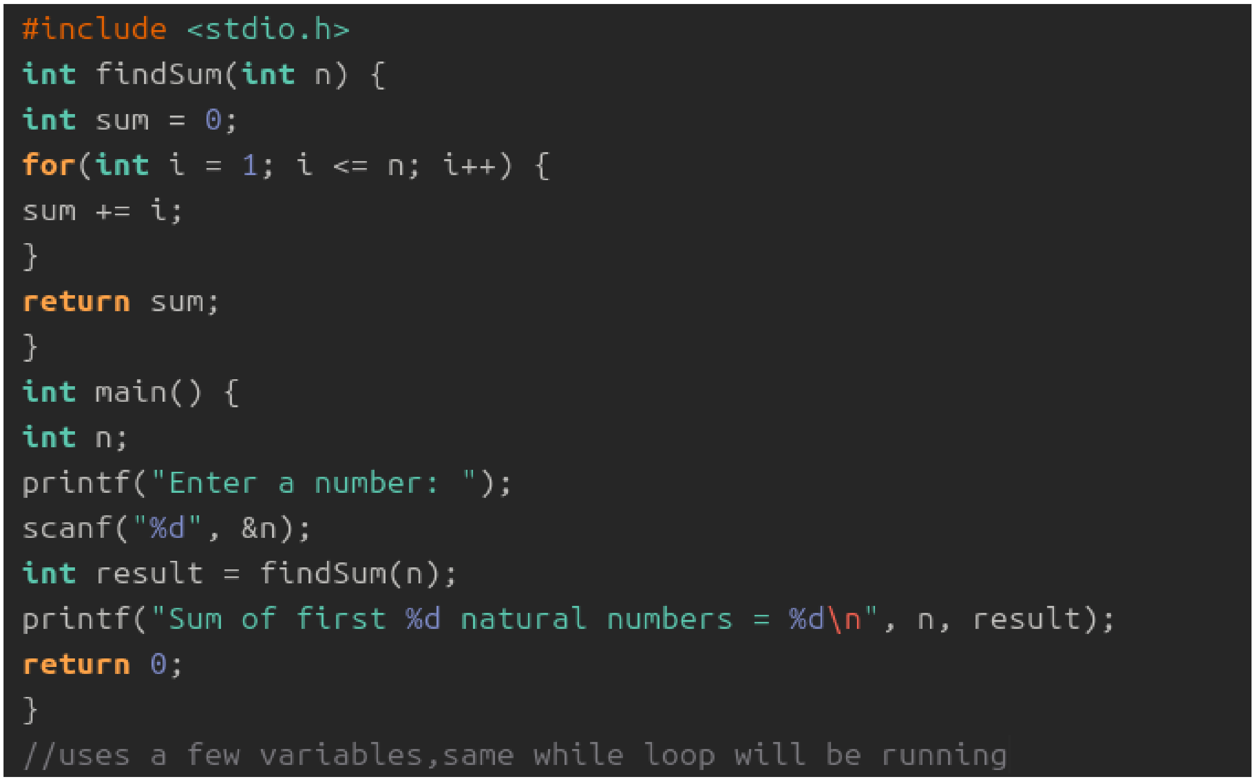
NAME:CH.JYOTHIRMAI

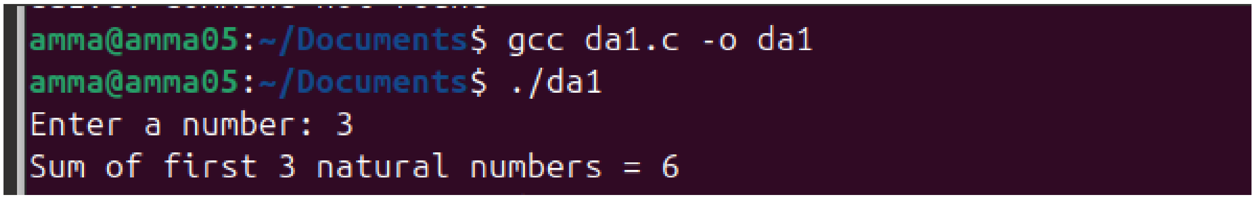
ROLL NO:CH.SC.U4CSE24107

DAA WEEK-1

1. write program to find sum on n natural numbers using user define function



```
#include <stdio.h>
int findSum(int n) {
int sum = 0;
for(int i = 1; i <= n; i++) {
sum += i;
}
return sum;
}
int main() {
int n;
printf("Enter a number: ");
scanf("%d", &n);
int result = findSum(n);
printf("Sum of first %d natural numbers = %d\n", n, result);
return 0;
}
//uses a few variables,same while loop will be running
```



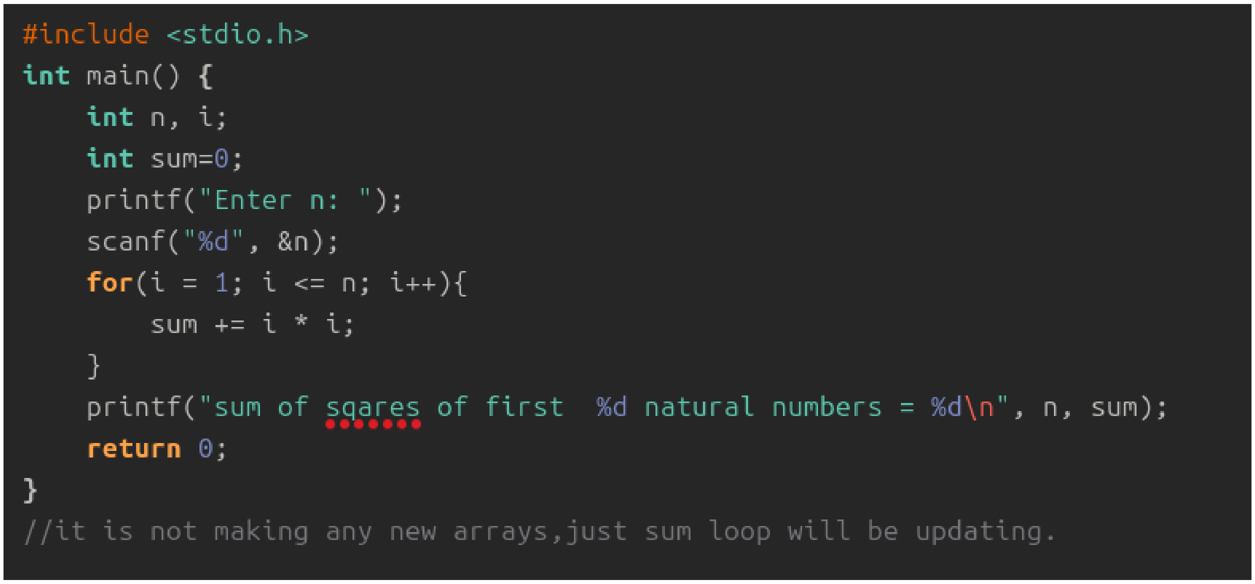
```
amma@amma05:~/Documents$ gcc da1.c -o da1
amma@amma05:~/Documents$ ./da1
Enter a number: 3
Sum of first 3 natural numbers = 6
```

JUSTIFICATION:

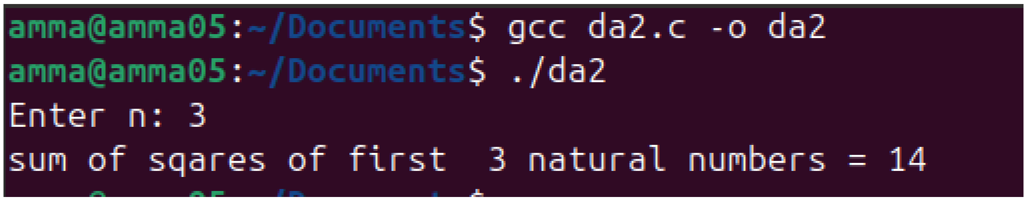
O(1) Fixed no.of integer variables used

Memory does not increase as input n gets larger

2. write a program to find sum of squares of first n natural numbers



```c
#include <stdio.h>
int main() {
    int n, i;
    int sum=0;
    printf("Enter n: ");
    scanf("%d", &n);
    for(i = 1; i <= n; i++){
        sum += i * i;
    }
    printf("sum of sqares of first  %d natural numbers = %d\n", n, sum);
    return 0;
}
//it is not making any new arrays,just sum loop will be updating.
```



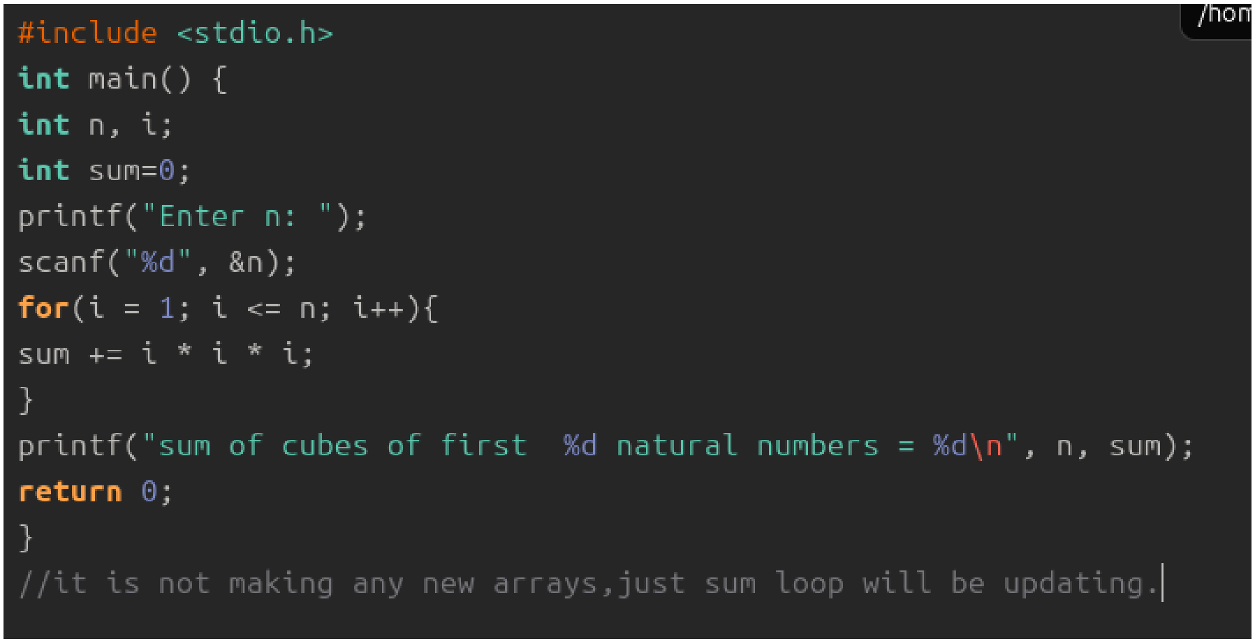
```
amma@amma05:~/Documents$ gcc da2.c -o da2
amma@amma05:~/Documents$ ./da2
Enter n: 3
sum of sqares of first  3 natural numbers = 14
```

JUSTIFICATION:

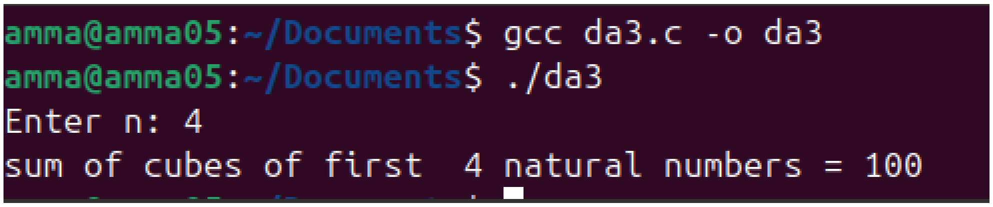
O(1)

It will not make new arrays, only sum loop updates single variable

3. write a program sum of cubes of first n natural numbers



```
#include <stdio.h>
int main() {
int n, i;
int sum=0;
printf("Enter n: ");
scanf("%d", &n);
for(i = 1; i <= n; i++){
sum += i * i * i;
}
printf("sum of cubes of first  %d natural numbers = %d\n", n, sum);
return 0;
}
//it is not making any new arrays,just sum loop will be updating.
```



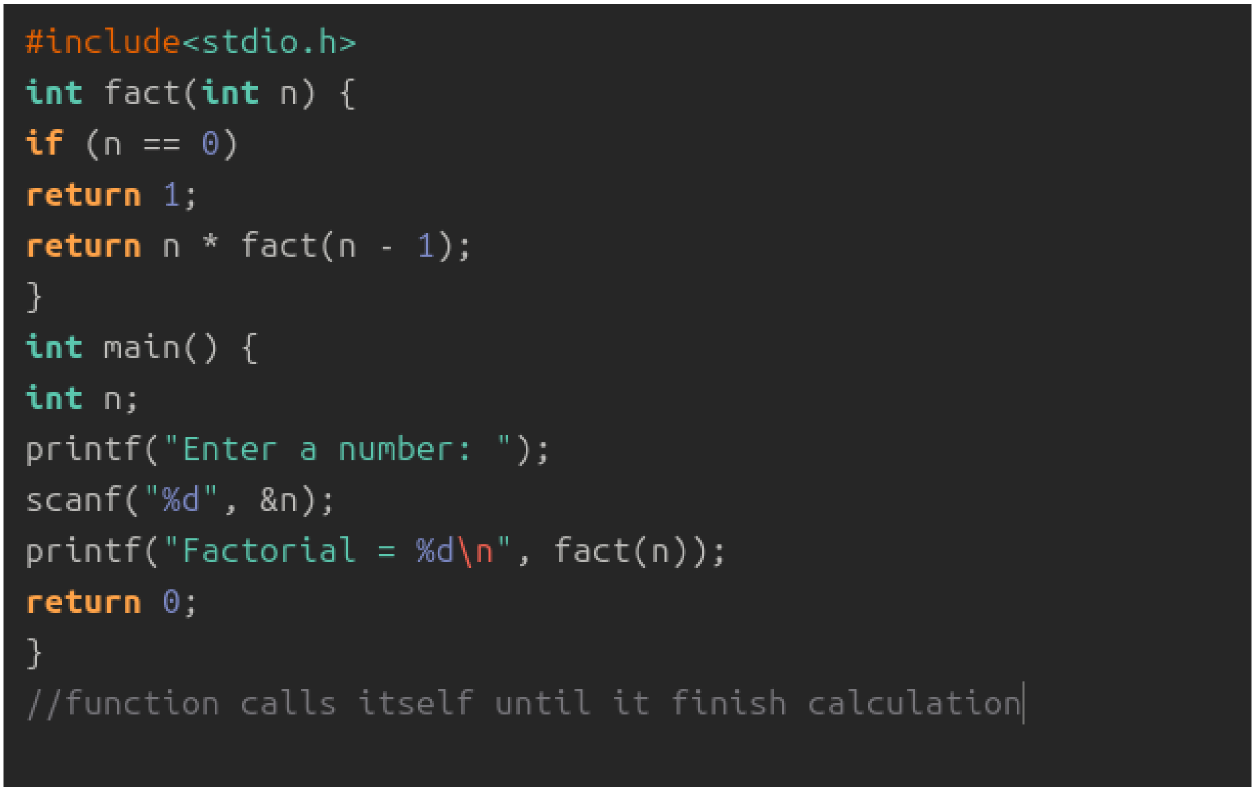
```
amma@amma05:~/Documents$ gcc da3.c -o da3
amma@amma05:~/Documents$ ./da3
Enter n: 4
sum of cubes of first  4 natural numbers = 100
```

JUSTIFICATION:

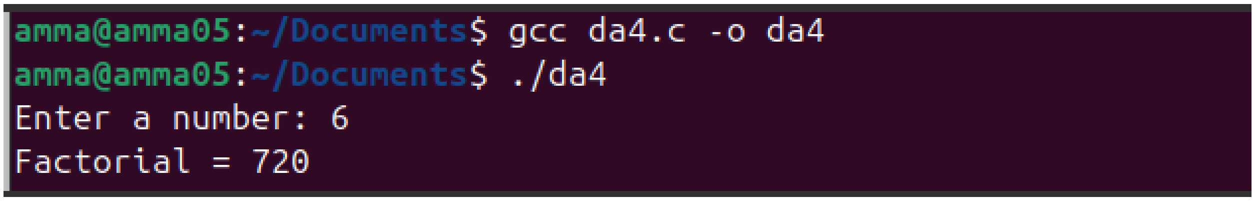
O(1)

It will not make new arrays, only sum loop updates single variable

4. write a program to find factorial of a given integer using recursion



```
#include<stdio.h>
int fact(int n) {
if (n == 0)
return 1;
return n * fact(n - 1);
}
int main() {
int n;
printf("Enter a number: ");
scanf("%d", &n);
printf("Factorial = %d\n", fact(n));
return 0;
}
//function calls itself until it finish calculation
```



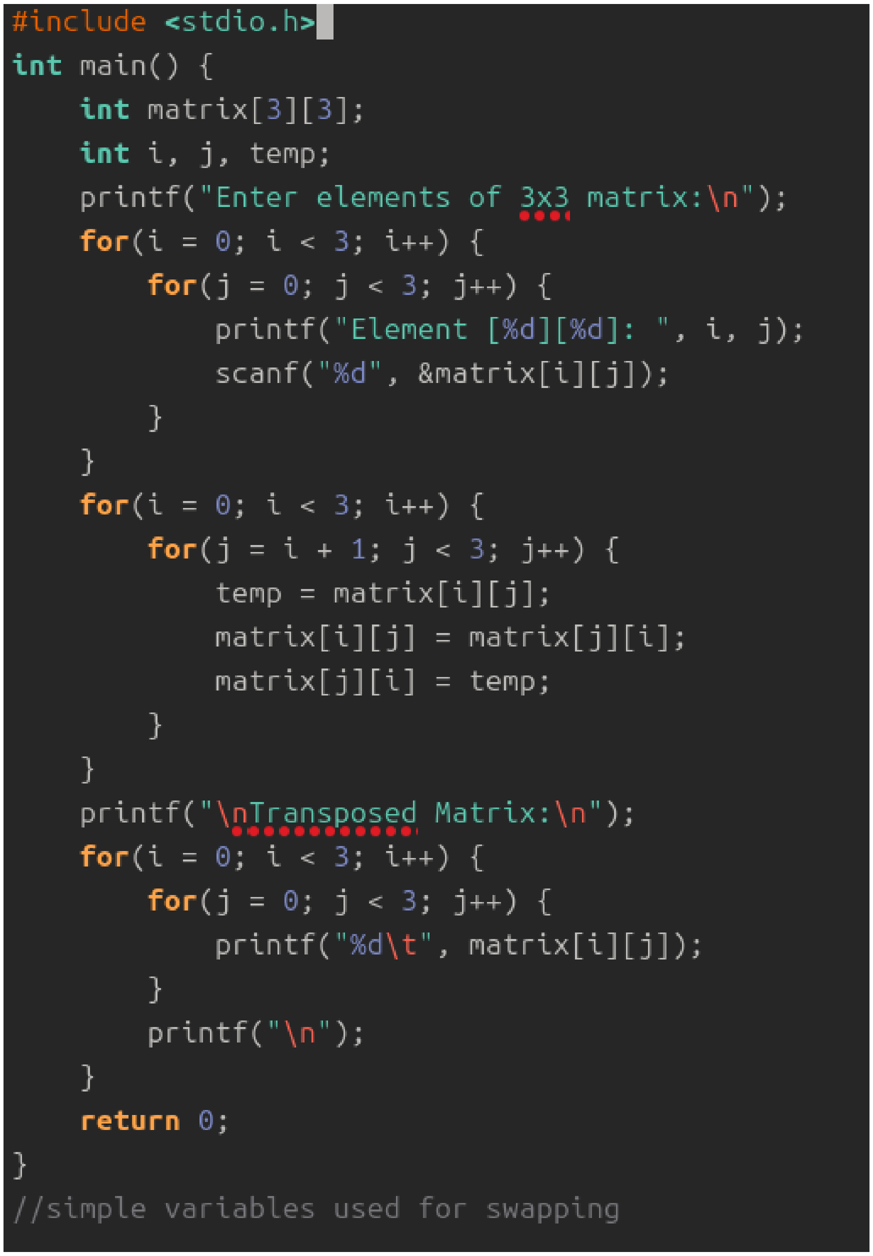
```
amma@amma05:~/Documents$ gcc da4.c -o da4
amma@amma05:~/Documents$ ./da4
Enter a number: 6
Factorial = 720
```

JUSTIFICATION:

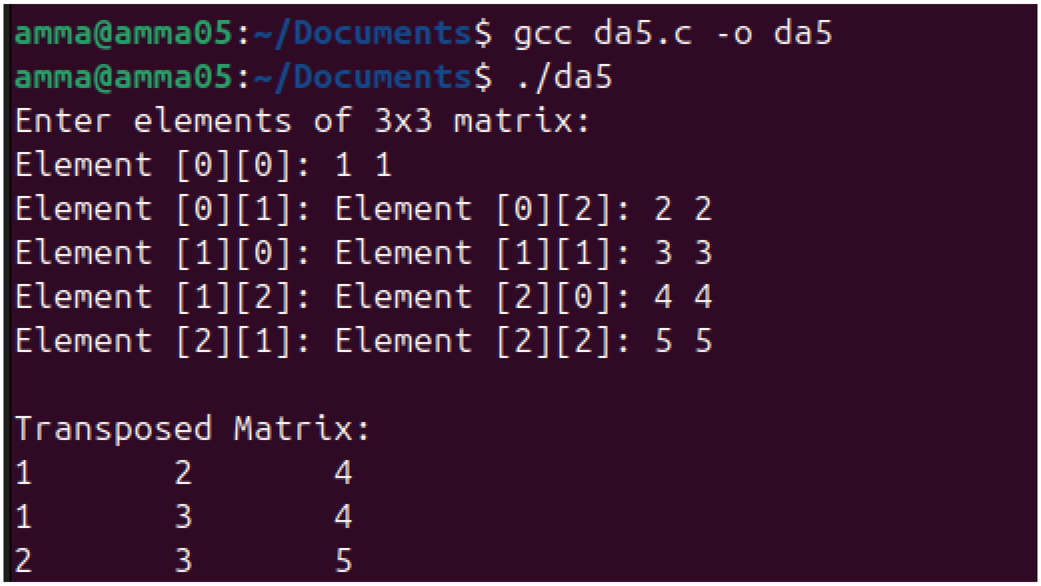
O(n)

Function calls itself until it completes the calculation

5. write a program for transposing a 3*3 matrices



```c
#include <stdio.h>
int main() {
    int matrix[3][3];
    int i, j, temp;
    printf("Enter elements of 3x3 matrix:\n");
    for(i = 0; i < 3; i++) {
        for(j = 0; j < 3; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }
    for(i = 0; i < 3; i++) {
        for(j = i + 1; j < 3; j++) {
            temp = matrix[i][j];
            matrix[i][j] = matrix[j][i];
            matrix[j][i] = temp;
        }
    }
    printf("\nTransposed Matrix:\n");
    for(i = 0; i < 3; i++) {
        for(j = 0; j < 3; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}
//simple variables used for swapping
```



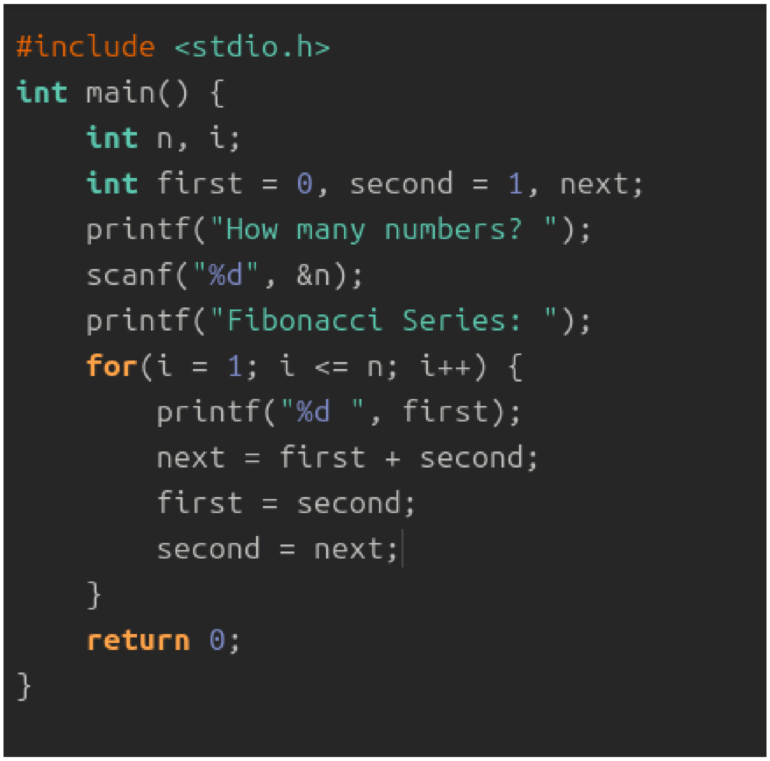
```
amma@amma05:~/Documents$ gcc da5.c -o da5
amma@amma05:~/Documents$ ./da5
Enter elements of 3x3 matrix:
Element [0][0]: 1 1
Element [0][1]: Element [0][2]: 2 2
Element [1][0]: Element [1][1]: 3 3
Element [1][2]: Element [2][0]: 4 4
Element [2][1]: Element [2][2]: 5 5

Transposed Matrix:
1       2       4
1       3       4
2       3       5
```

JUSTIFICATION:

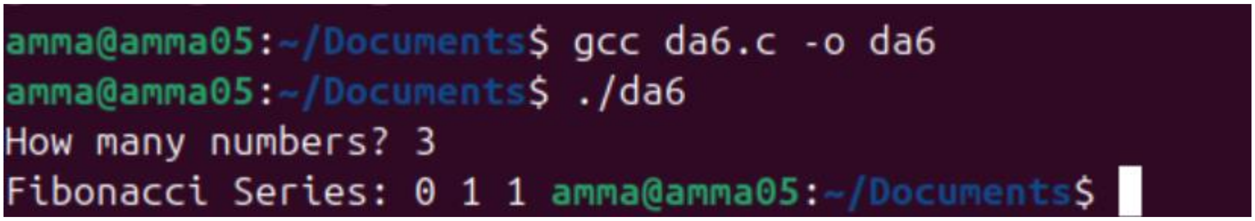
O(1)

Memory used does not grow with input size.

6. write a program to find fibonacci series



```c
#include <stdio.h>
int main() {
    int n, i;
    int first = 0, second = 1, next;
    printf("How many numbers? ");
    scanf("%d", &n);
    printf("Fibonacci Series: ");
    for(i = 1; i <= n; i++) {
        printf("%d ", first);
        next = first + second;
        first = second;
        second = next;
    }
    return 0;
}
```



```
amma@amma05:~/Documents$ gcc da6.c -o da6
amma@amma05:~/Documents$ ./da6
How many numbers? 3
Fibonacci Series: 0 1 1 amma@amma05:~/Documents$
```

JUSTIFICATION:

O(1) This code uses fixed no.of variables to store integers.

We are not storing whole fibonacci numbers in array.