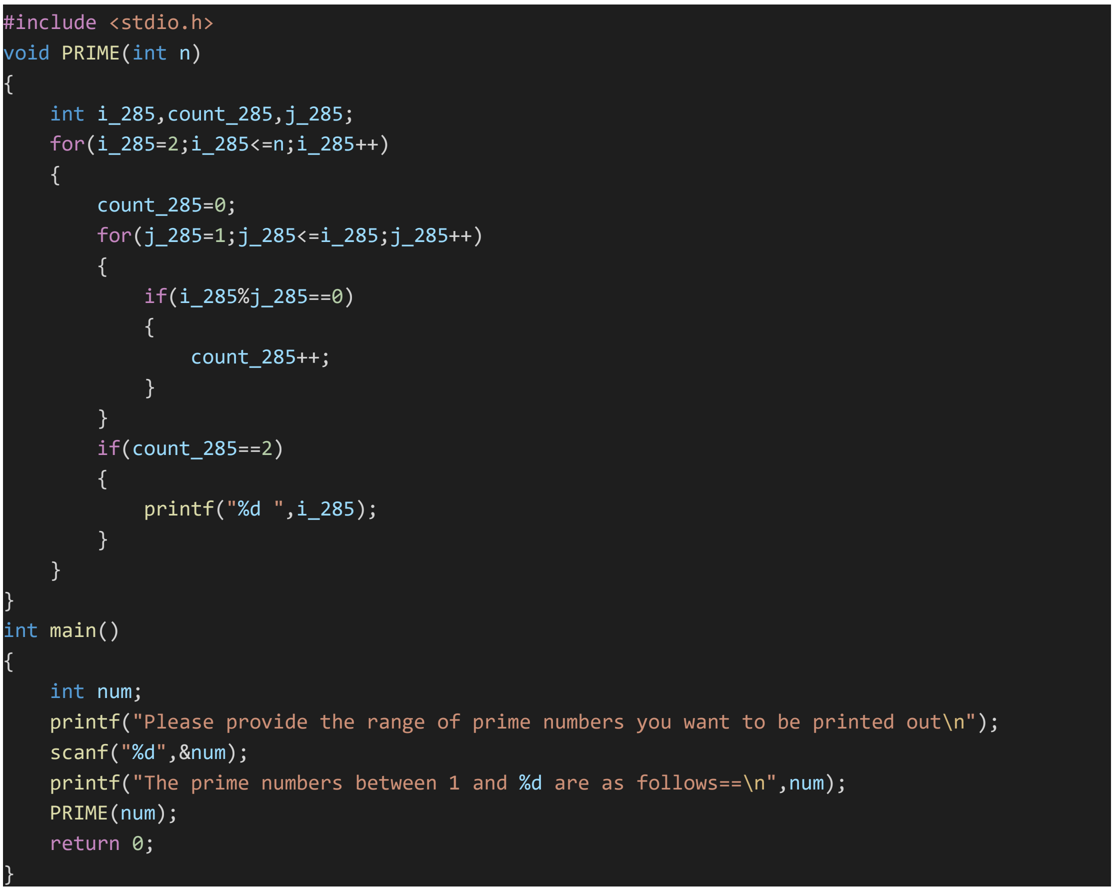
KIDUS ABEBE MEKONEN
Roll No- 2106285
Section A11
IT department

May 5 Lab questions

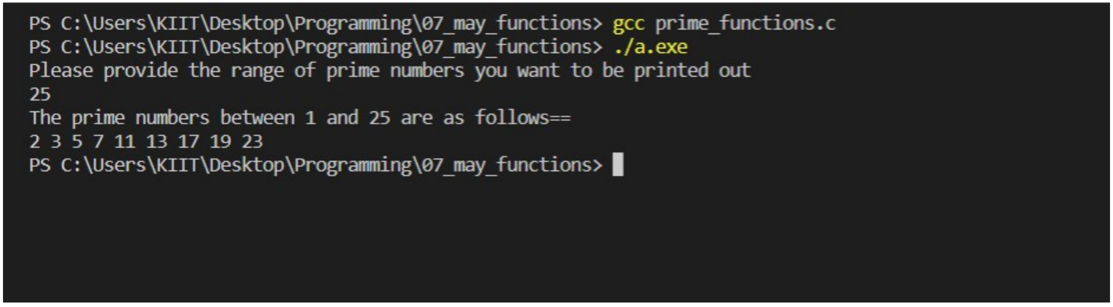
#1.WAP to generate all the prime numbers between 1 and n by using a user defined function (say isPRIME) to be used for prime number testing, where n is a value supplied by the user.

Code:



```c
#include <stdio.h>
void PRIME(int n)
{
    int i_285,count_285,j_285;
    for(i_285=2;i_285<=n;i_285++)
    {
        count_285=0;
        for(j_285=1;j_285<=i_285;j_285++)
        {
            if(i_285%j_285==0)
            {
                count_285++;
            }
        }
        if(count_285==2)
        {
            printf("%d ",i_285);
        }
    }
}
int main()
{
    int num;
    printf("Please provide the range of prime numbers you want to be printed out\n");
    scanf("%d",&num);
    printf("The prime numbers between 1 and %d are as follows==\n",num);
    PRIME(num);
    return 0;
}
```

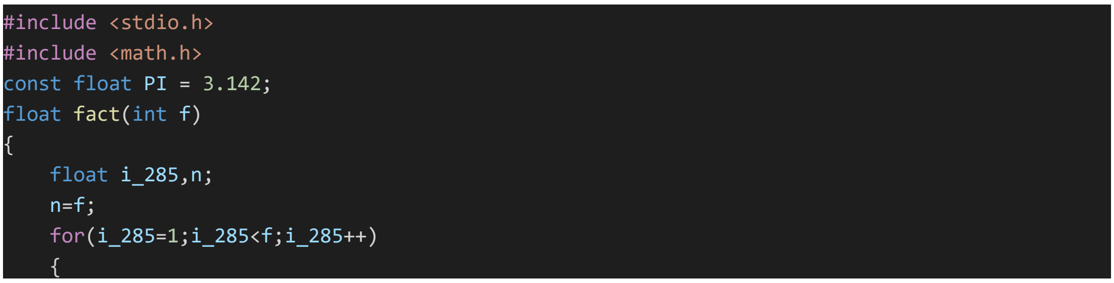
Output:



```
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> gcc prime_functions.c
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> ./a.exe
Please provide the range of prime numbers you want to be printed out
25
The prime numbers between 1 and 25 are as follows==
2 3 5 7 11 13 17 19 23
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> 
```

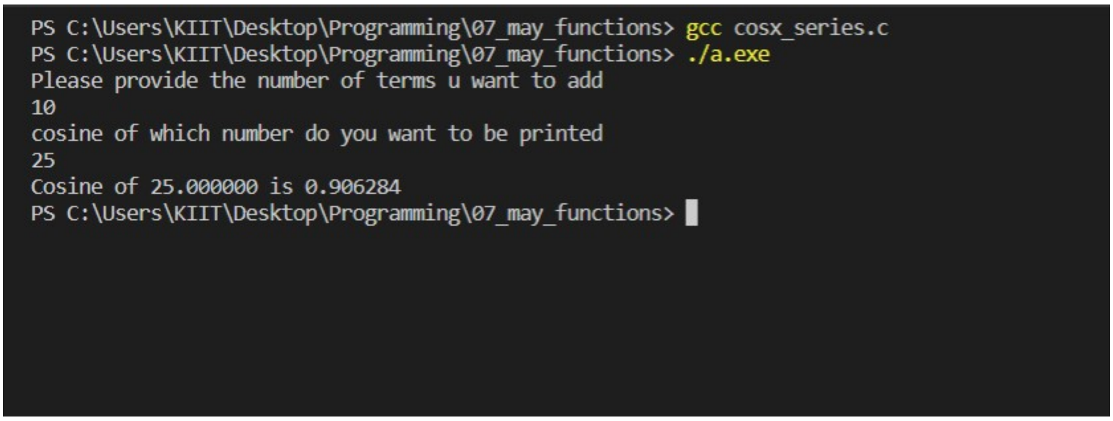
#2. WAP to compute the cosine series using function.

Code:

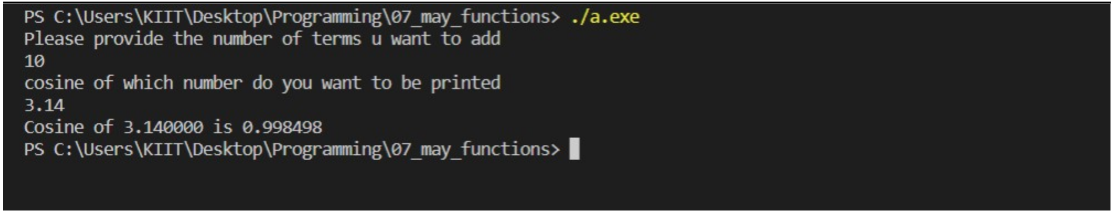


```c
#include <stdio.h>
#include <math.h>
const float PI = 3.142;
float fact(int f)
{
    float i_285,n;
    n=f;
    for(i_285=1;i_285<f;i_285++)
    {
        n=n*(f-i_285);
    }
    return n;
}
float COS(float x, float n)
{
    int i_285,j;
    float sum_285,qoutient_285,a,b,c;
    x = x * (PI / 180.0);
    sum_285=0;
    for(i_285=1;i_285<n;i_285++)
    {
        if((i_285%2)!=0)
        {
            c=i_285*2;
            a=-(pow(x,c));
            b=fact(c);
            qoutient_285=a/b;
        }
        else if((i_285%2)==0)
        {
            c=i_285*2;
            a=pow(x,c);
            b=fact(c);
            qoutient_285=a/b;
        }
        sum_285=sum_285+qoutient_285;
    }
    return sum_285;
}
int main()
{
    float num,n;
    float result;
    printf("Please provide the number of terms u want to add\n");
    scanf("%f",&n);
    printf("cosine of which number do you want to be printed\n");
    scanf("%f",&num);
    result=1+COS(num,n);
    printf("Cosine of %f is %f\n",num,result);
    return 0;
}
```

Output:

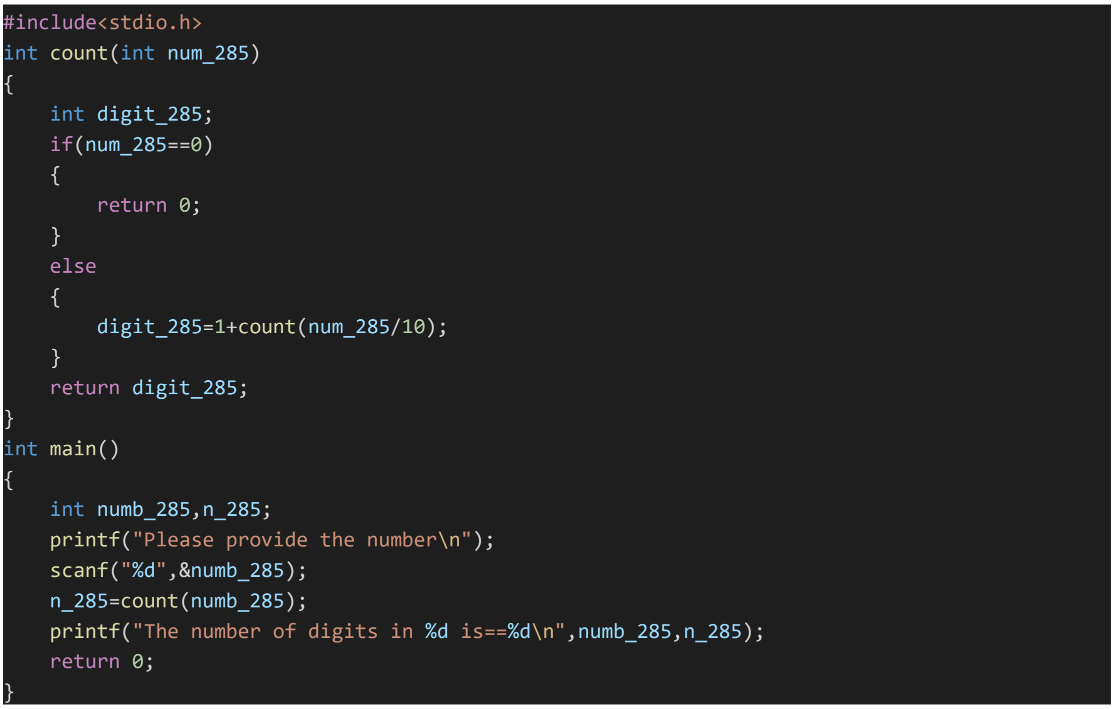


```
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> gcc cosx_series.c
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> ./a.exe
Please provide the number of terms u want to add
10
cosine of which number do you want to be printed
25
Cosine of 25.000000 is 0.906284
PS C:\Users\KIIT\Desktop\Programming\07_may_functions>
```



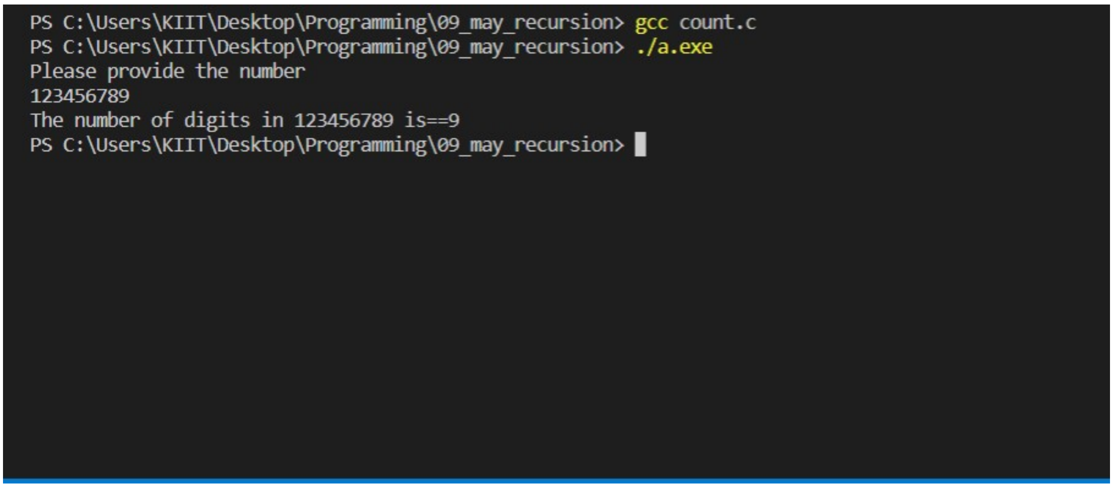
```
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> ./a.exe
Please provide the number of terms u want to add
10
cosine of which number do you want to be printed
3.14
Cosine of 3.140000 is 0.998498
PS C:\Users\KIIT\Desktop\Programming\07_may_functions> 
```

#3. WAP to count number of digits of a positive integer n by using recursion.
Code:



```
#include<stdio.h>
int count(int num_285)
{
    int digit_285;
    if(num_285==0)
    {
        return 0;
    }
    else
    {
        digit_285=1+count(num_285/10);
    }
    return digit_285;
}
int main()
{
    int numb_285,n_285;
    printf("Please provide the number\n");
    scanf("%d",&numb_285);
    n_285=count(numb_285);
    printf("The number of digits in %d is==%d\n",numb_285,n_285);
    return 0;
}
```

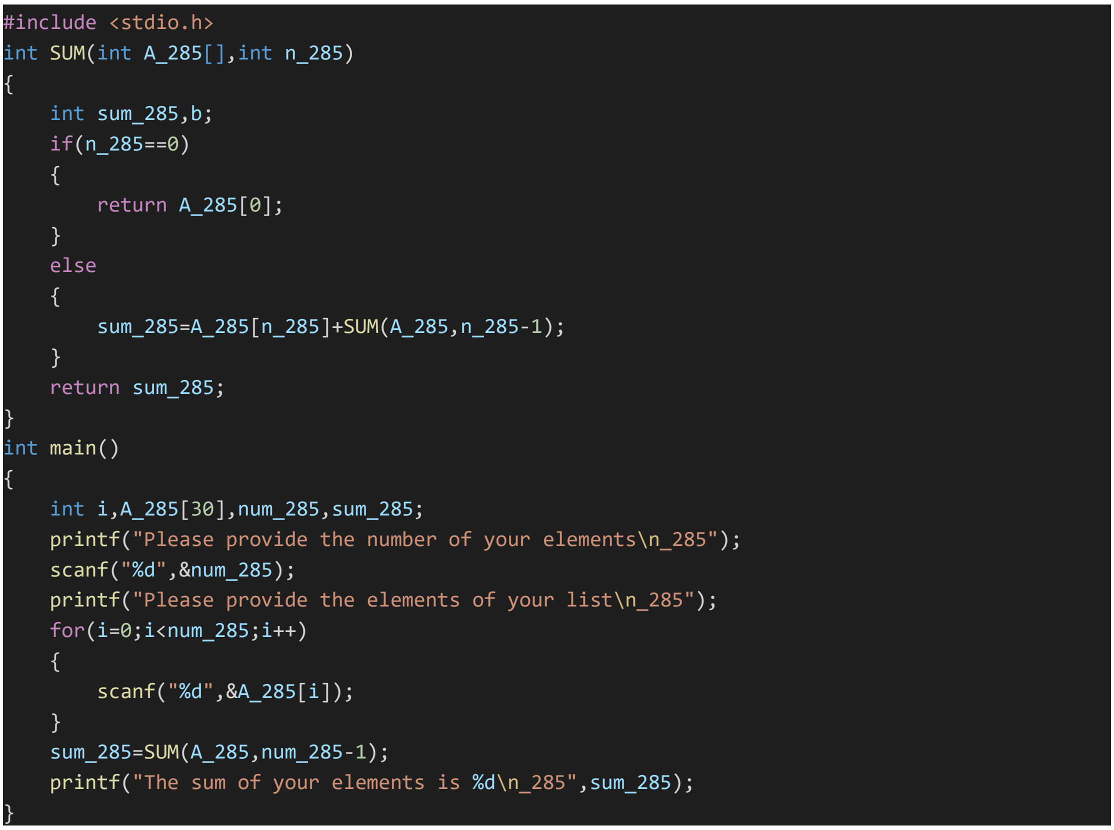
Output:



```
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> gcc count.c
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> ./a.exe
Please provide the number
123456789
The number of digits in 123456789 is==9
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> 
```

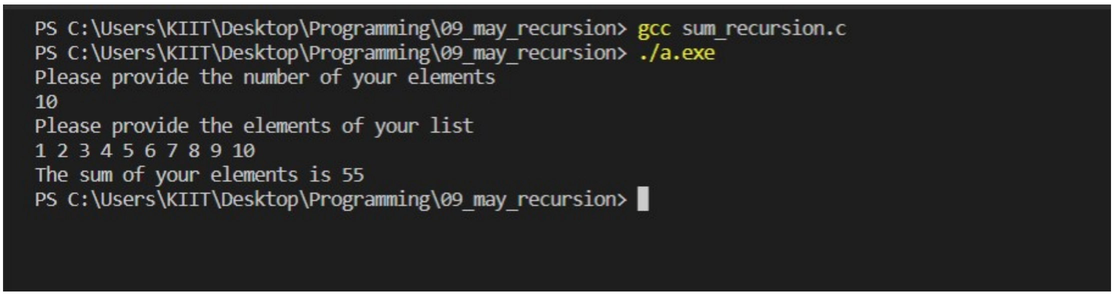
#4. WAP to find out the sum of n elements of an integer array a[] by using recursion.

Code:

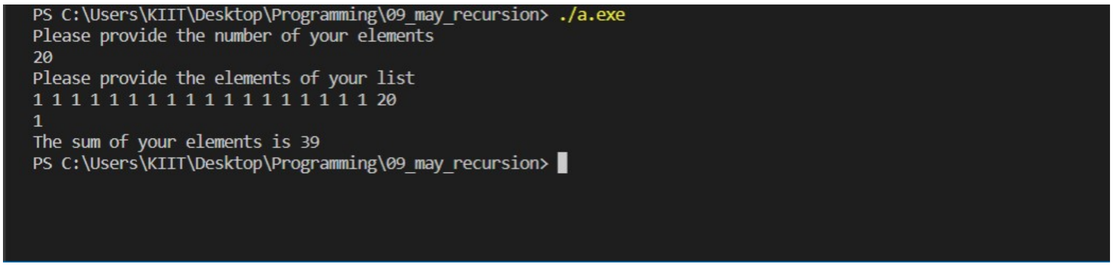


```
#include <stdio.h>
int SUM(int A_285[],int n_285)
{
    int sum_285,b;
    if(n_285==0)
    {
        return A_285[0];
    }
    else
    {
        sum_285=A_285[n_285]+SUM(A_285,n_285-1);
    }
    return sum_285;
}
int main()
{
    int i,A_285[30],num_285,sum_285;
    printf("Please provide the number of your elements\n_285");
    scanf("%d",&num_285);
    printf("Please provide the elements of your list\n_285");
    for(i=0;i<num_285;i++)
    {
        scanf("%d",&A_285[i]);
    }
    sum_285=SUM(A_285,num_285-1);
    printf("The sum of your elements is %d\n_285",sum_285);
}
```

Output:

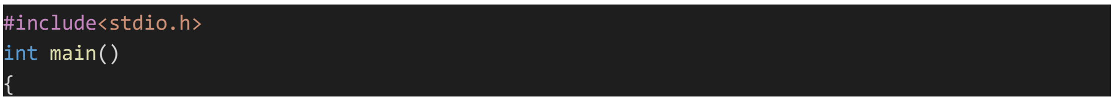


```
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> gcc sum_recursion.c
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> ./a.exe
Please provide the number of your elements
10
Please provide the elements of your list
1 2 3 4 5 6 7 8 9 10
The sum of your elements is 55
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> 
```



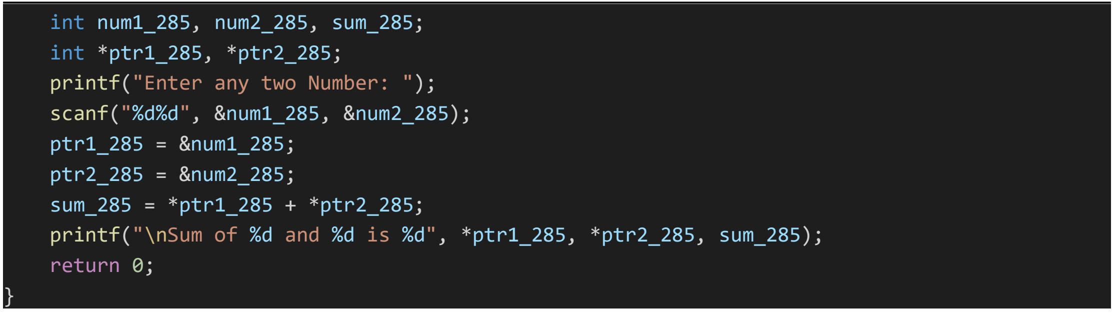
```
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> ./a.exe
Please provide the number of your elements
20
Please provide the elements of your list
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 20
1
The sum of your elements is 39
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> 
```

#5. WAP to add two numbers using call by reference.
Code:



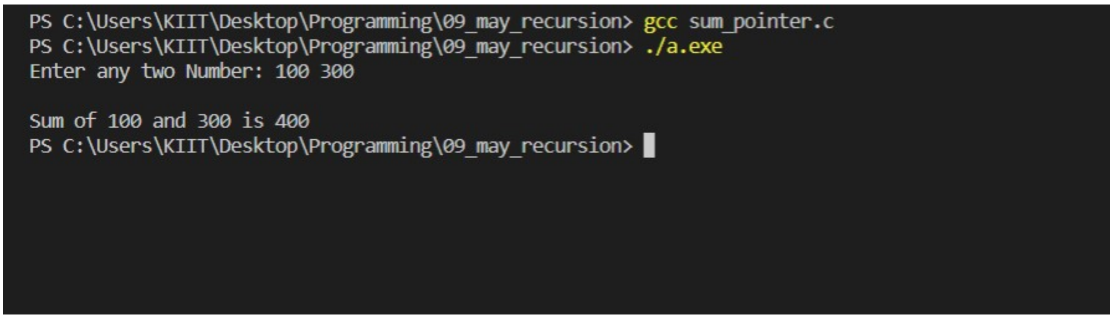
```
#include<stdio.h>
int main()
{
```

KIDUS ABEBE MEKONEN
Roll No- 2106285
Section A11
IT department



```c
    int num1_285, num2_285, sum_285;
    int *ptr1_285, *ptr2_285;
    printf("Enter any two Number: ");
    scanf("%d%d", &num1_285, &num2_285);
    ptr1_285 = &num1_285;
    ptr2_285 = &num2_285;
    sum_285 = *ptr1_285 + *ptr2_285;
    printf("\nSum of %d and %d is %d", *ptr1_285, *ptr2_285, sum_285);
    return 0;
}
```

Output:



```
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> gcc sum_pointer.c
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> ./a.exe
Enter any two Number: 100 300

Sum of 100 and 300 is 400
PS C:\Users\KIIT\Desktop\Programming\09_may_recursion> 
```