**May 7\_9 Lab questions**

#1. WAP to find the LCM of two numbers a and b by using a suitable function

(say LCM) for this.

Code:

#include <stdio.h>

void LCM(int a\_285,int b\_285)

{

    int i\_285,result\_285;

    for(i\_285=1;i\_285<5000;i\_285++)

    {

        if(i\_285%a\_285==0 && i\_285%b\_285==0)

        {

            printf("LCM of %d and %d is %d\n",a\_285,b\_285,i\_285);

            break;

        }

    }

}

int main()

{

    int a\_285,b\_285;

    printf("Please provide the two numbers\n");

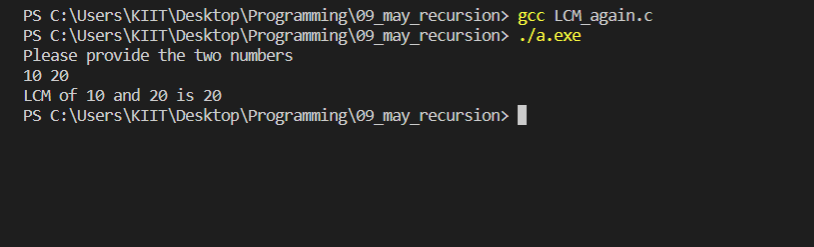
    scanf("%d%d",&a\_285,&b\_285);

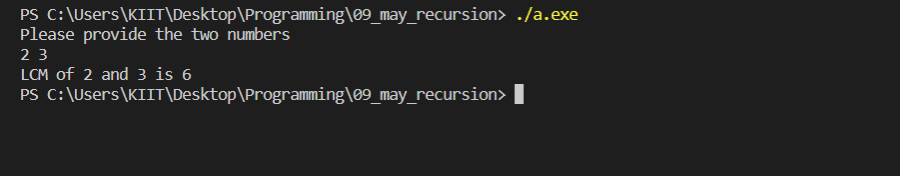
    LCM(a\_285,b\_285);

    return 0;

}

Output:





#2. 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]);

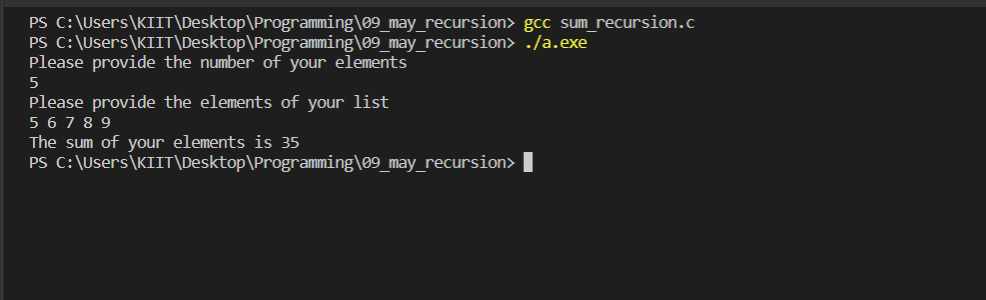
    }

    sum\_285=SUM(A\_285,num\_285-1);

    printf("The sum of your elements is %d\n\_285",sum\_285);

}

Output:



#3. WAP by designing a recursive function to calculate the sum of all even

digits of any given integer.

Code:

#include <stdio.h>

int SUM(int count\_285, int arr\_285[])

{

    int i,total;

    if(count\_285<=0)

        return 0;

    total= arr\_285[count\_285-1]+SUM(count\_285-1,arr\_285);

    return total;

}

int main()

{

    int num\_285,sum,arr\_285[20],i,rem\_285,count\_285;

    printf("Please provide an integer\n");

    scanf("%d",&num\_285);

    for(i=0;num\_285!=0;)

    {

        rem\_285=num\_285%10;

        if(rem\_285%2==0)

        {

            arr\_285[i]=rem\_285;

            count\_285++;

            i++;

        }

        num\_285=num\_285/10;

    }

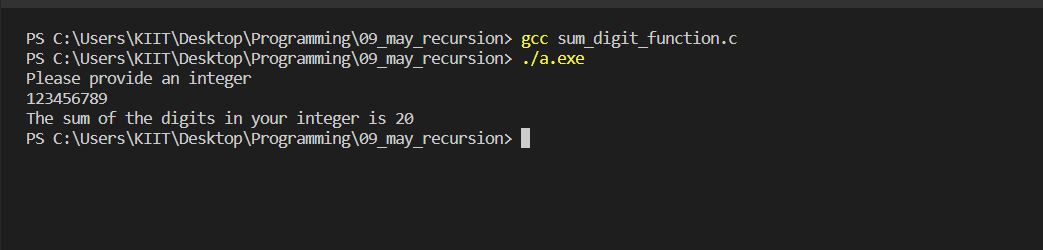
    sum=SUM(count\_285,arr\_285);

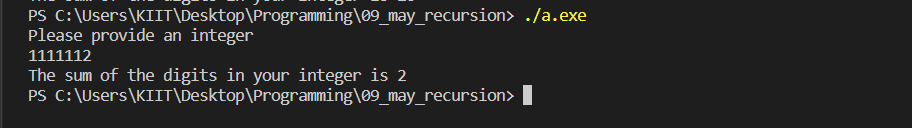
    printf("The sum of the digits in your integer is %d\n",sum);

    return 0;

}

Output:





#4. WAP to generate the first n terms of the sequence by writing a suitable

user defined function (say fib) to be used to get nth term Fibonacci value.

Code:

#include <stdio.h>

void fib(int n)

{

    int term1\_285,term2\_285,term3\_285,i,fibn[100];

    term1\_285=0;

    fibn[0]=term1\_285;

    term2\_285=1;

    fibn[1]=term2\_285;

    term3\_285=term1\_285+term2\_285;

    for(i=2;i<n;i++)

    {

        fibn[i]=fibn[i-1]+fibn[i-2];

    }

    for(i=0;i<n;i++)

    {

        printf("%d ",fibn[i]);

    }

}

int main()

{

    int n;

    printf("Provide the number of fibonacci terms u want=\n");

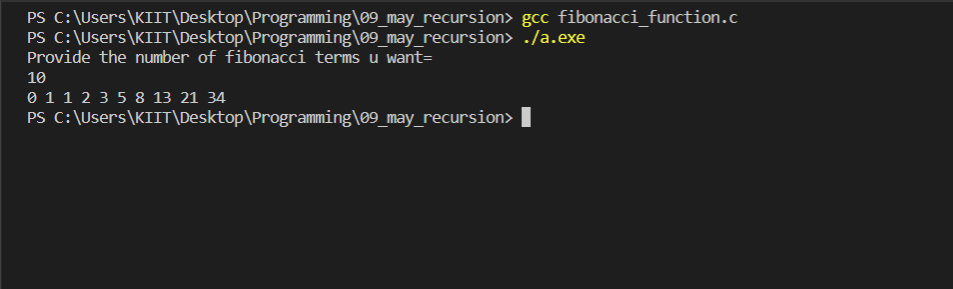
    scanf("%d",&n);

    fib(n);

    return 0;

}

Output:



#5. WAP to change the value of constant integer using pointers.

Code:

#include <stdio.h>

int main()

{

    const float pi\_285=3.14;

    float \*ptr\_285;

    ptr\_285=&pi\_285;

    printf("The value of pi is %f\n",\*ptr\_285);

    printf("But its okay, give your own desired value to pi\n");

    scanf("%f",ptr\_285);

    printf("=-=-==-=-=-=-=-=-=-==-=-=-=-=-=-\n");

    printf("The new value you have given is = %f\n",\*ptr\_285);

    printf("=-=-==-=-=-=-=-=-=-==-=-=-=-=-=-\n");

    return 0;

}

Output:

