



**Programación en Ingeniera.**

**Practica 1.**

**Determinantes de series de constantes y de Taylor.**

**Alumnos:**

Michel A. Ramos Soto  
Ricardo Romero Vega  
Naim Tejeda Díaz

**NUA:**

768935  
389765  
145154

**Correo:**

michel.ramos@ugto.mx

**Maestro:**

Dr. Mario Alberto Ibarra Manzano

**Correo:**

mibarram@gmail.com

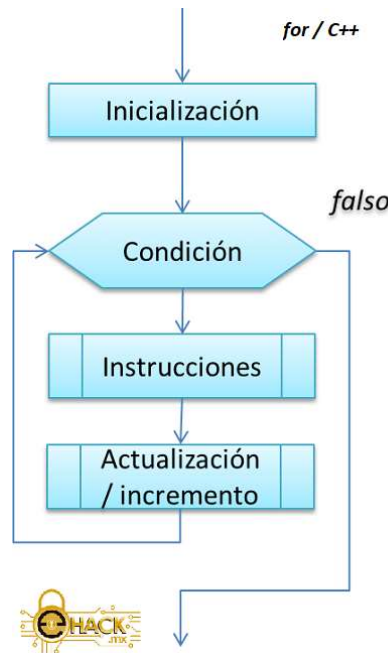
**Fecha de entrega:**

10/10/18

## Introducción

En esta primera practica aplicaremos los conocimientos adquiridos en clase para emplearlos en los ejercicios planteados en los ejercicios en la práctica, hacerlos lo más eficientes posibles y lo más limpio posible, el lenguaje C es un lenguaje de programación originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de Sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas.



El bucle `for` es una estructura de control en programación en la que se puede indicar de antemano el número mínimo de interacciones. Está disponible en casi todos los lenguajes de programación imperativos.

## Objetivos

- Determinar los valores de las constantes.
- Emplear lo aprendido en clase.
- Hacer los ejercicios lo más eficaces y eficientes posibles.
- Hacer las simulaciones necesarias para llegar al resultado más exacto.
- Trabajar en equipo.

## Programa 1

$$\ln(2) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \dots$$

Ciclos:

Recurrencia:

```
#include <stdio.h>
int main()
{
    int i,x,sig;
    float ln=0,lnf=0;
    do{
        printf("Obtener el resultado del logartimo natural de ln (2)\n");
        printf("\nIngresa el numero de iteracciones: ");
        scanf("%d",&x);
    }while(x<0);
    for(i=0;i<x;i++)
    {
        sig = (1-((2)*(i%2)));
        ln =((sig)*(1.0/(i+1)));
        printf("%.3f ",ln);
        lnf +=ln;
    }
    printf("\nEl ln(2) es: %.3f\n",lnf);

    return 0;
}
```



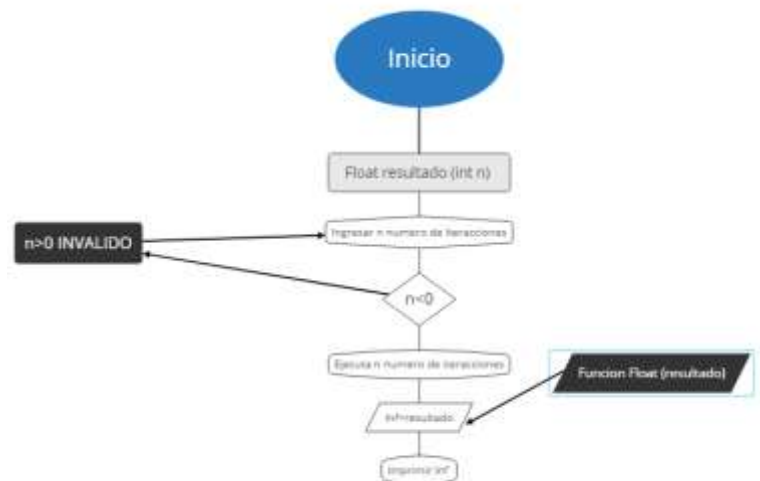
```
#include <stdio.h>

float resultado(int n);

int main()
{
    int n;
    float lnf=0;
    do{
        printf("Obtener el resultado del logartimo natural de (2) mediante iteracciones:\n");
        printf("\nIngresa el numero de iteracciones: ");
        scanf("%d",&n);
    }while(n<0);
    lnf=resultado(n);
    printf("\nEl ln(2) es: %.3f\n",lnf);

    return 0;
}

float resultado(int n)
{
    float ln;
    int i,den,sig;
    for(i=0;i<n;i++)
    {
        sig=(1-((2)*(i%2)));
        den=(i+1);
        ln+=(sig*1.0/den);
    }
    return ln;
}
```



## Programa 1

Programa 1:     $n=x$                        $f_n(x)$                        $F_n(x)-f_{n-1}(x)$                       Calculadora  
 **$\ln(2)$**   $n$

1	2	1	-0.50	1
2	2	0.50	-0.25	0.5
4	2	0.58		
8	2	0.63	-0.13	0.58
16	2	0.66	-0.07	0.63
32	2	0.68	-0.03	0.66
64	2	0.69	-0.01	0.68
128	2	0.69	-0.01	0.69
				0.6

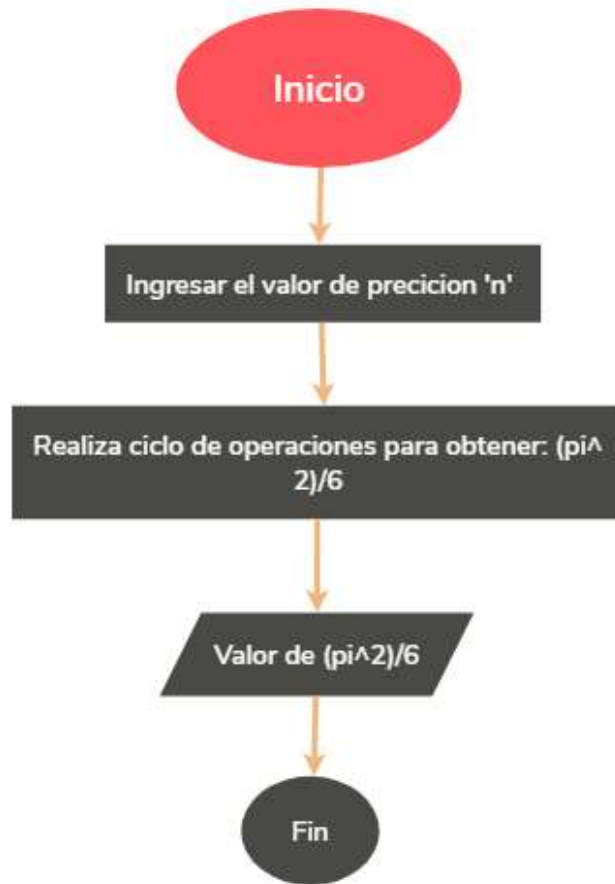


### Programa 3

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots \quad (3)$$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float pi2;
6      int i,n;
7      //Ecuacion 3 (Pi^2)/6
8      printf("Series de Constantes: Calculo de (PI^2)/6");
9      printf("\n\nIngrese la presicion o el numero de iteraciones: ");
10     scanf("%d",&n);
11
12     for(i=1;i<n+1;i++)
13     {
14         pi2+=(1.0/((i)*(i)));
15     }
16     printf("\n\n(Pi^2)/6 = %f.\n\n",pi2);
17     return 0;
18 }
```

$n$	$x$	$f_n(x)$
1	$\pi^2/6$	1.000000
2	$\pi^2/6$	1.250000
4	$\pi^2/6$	1.423611
8	$\pi^2/6$	1.527422
16	$\pi^2/6$	1.584347
32	$\pi^2/6$	1.614167
64	$\pi^2/6$	1.629431
128	$\pi^2/6$	1.637152



## Programa 4

$$\frac{\pi^2}{8} = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots \quad (4)$$

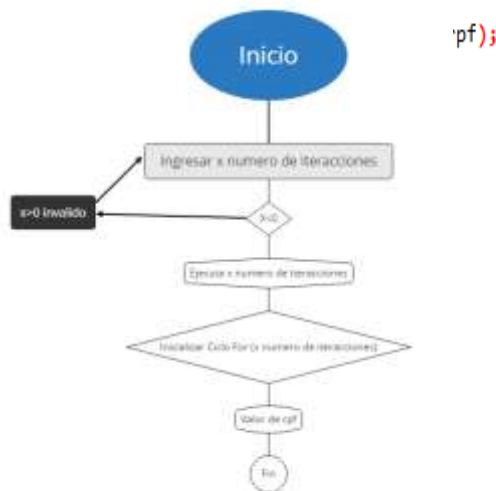
Ciclos:

Recurrencia:

```
#include<stdio.h>

int main()
{
    int i,x,den;

    float r=0, rpf=0;
    do{
        printf("Obtener el resultado de PI^2/8 mediante iteracciones\n");
        printf("\nNumero de iteracciones: ");
        scanf("%d",&x);
    }while(x<0);
    for(i=0,den=0;i<x;i++)
    {
        den=((2*(i)+1)*(2*(i)+1));
        r=(1.0/den);
        printf("%+.6f",r);
        rpf+=r;
    }
```



```
#include<stdio.h>

float resultado(int x);

int main()
{
    int x;
    float rpf=0;
    do{
        printf("Obtener el resultado de PI^2/8:\n");
        printf("\nNumero de iteracciones: ");
        scanf("%d",&x);
    }while(x
    <0);
    rpf=resultado(x);
    printf("\nEl resultado de Pi^2/8 es: %.3f\n",rpf);
float resultado(int x)
} {
    int i,den;
    float rp=0;
    for(i=0,den=0;i<x;i++)
    {
        den=((2*(i)+1)*(2*(i)+1));
        rp+=(1.0/den);
    }
```

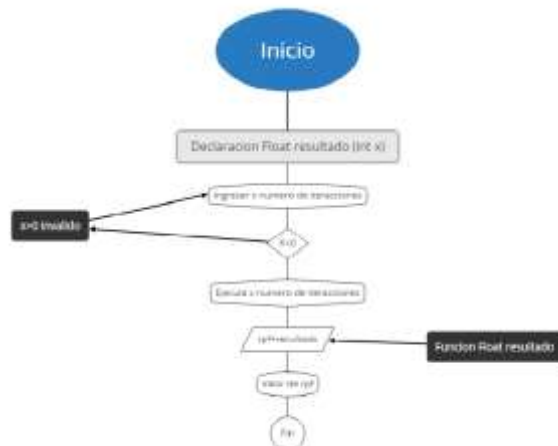




Tabla de programa 4:

n	$n=x$	$f_n(x)$	$F_n(x)-f_{n-1}(x)$	$E_n$
1	$\pi^2/8$	1		1
2	$\pi^2/8$	1.1111	0.1111	1.1111
4	$\pi^2/8$	1.171519	0.020408	1.1715
8	$\pi^2/8$	1.202491	$4.444 \times 10^{-3}$	1.20
16	$\pi^2/8$	1.218081	$1.041 \times 10^{-3}$	1.218081
32	$\pi^2/8$	1.225889	$2.52 \times 10^{-4}$	1.225889
64	$\pi^2/8$	1.229794	1.000062	1.229794
128	$\pi^2/8$	1.231748	$1.6 \times 10^{-5}$	1.231748

## Programa 5

$$\frac{1}{2} = \frac{1}{1 \times 3} + \frac{1}{3 \times 5} + \frac{1}{5 \times 7} + \frac{1}{7 \times 9} + \dots$$

(5)

### Ciclos

```
#include<stdio.h>

int main()
{
    int i,n,den;
    float r=0,med=0;
    do{
        printf("1/2 mediante n iteracciones.\nIngrese el valor de n: ");
        scanf("%d",&n);
    }while(n<0);
    for(i=0,den=0;i<n;i++)
    {
        den=((2*(i)+1)*(2*(i)+3));
        r=(1.0/den);
        med+=r;
    }
    printf("\n1/2 con %d iteracciones es: %f\n",n,med);
    return 0;
}
```



n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	0.5000	0.3333	0.6667	0.5000
2	0.5000	0.4000	0.6000	0.5000
4	0.5000	0.4444	0.5556	0.5000
8	0.5000	0.4706	0.5294	0.5000
16	0.5000	0.4848	0.5152	0.5000
32	0.5000	0.4923	0.5077	0.5000
64	0.5000	0.4961	0.5039	0.5000
128	0.5000	0.4981	0.5019	0.5000

### Recursivos

```
#include<stdio.h>

float resultado(int n);

int main()
{
    int n;
    float med=0;
    do{
        printf("1/2 mediante n iteracciones.\nIngrese el valor de n: ");
        scanf("%d",&n);
    }while(n<0);
    med=resultado(n);
    printf("\n1/2 con %d iteracciones es: %f\n",n,med);
    return 0;
}

float resultado(int n)
{
    int i,den;
    float r=0;
    for(i=0,den=0;i<n;i++)
    {
        den=((2*(i)+1)*(2*(i)+3));
        r+=(1.0/den);
    }
    return r;
}
```



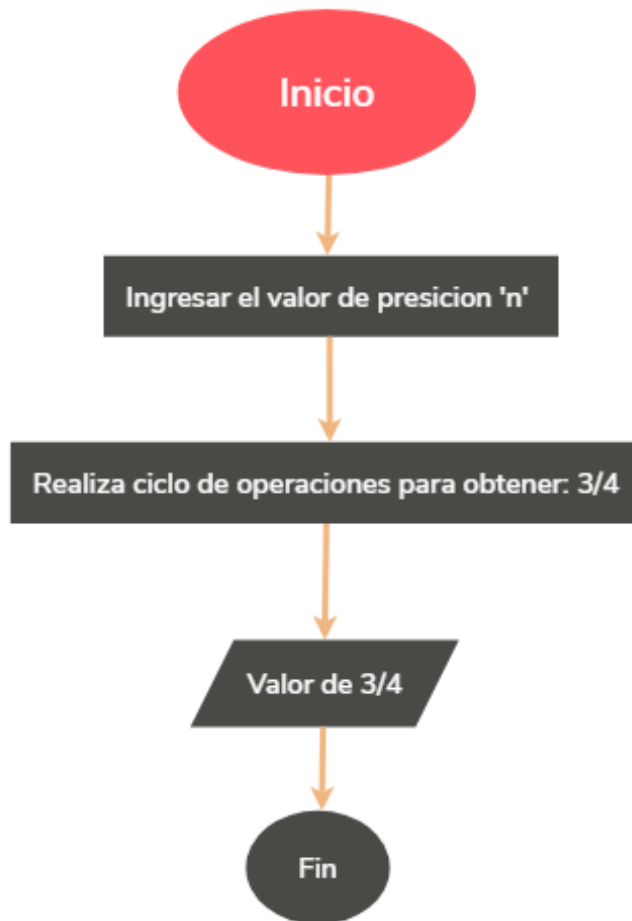
## Programa 6

$$\frac{3}{4} = \frac{1}{1 \times 3} + \frac{1}{2 \times 4} + \frac{1}{3 \times 5} + \frac{1}{4 \times 6} + \dots \quad (6)$$

```

1  #include <stdio.h>
2
3  int main()
4  {
5      float ec;
6      int i,c=1,n;
7      //Ecuacion 6 (3/4)
8      printf("Series de Constantes: Calculo de 3/4");
9      printf("\n\nIngrese la presicion o numero de iteraciones: ");
10     scanf("%d",&n);
11
12     for(i=0;i<=n;i++,c+=1)
13     {
14         ec+=(1.0)/(c*(c+2));
15     }
16     printf("\n\n3/4 = %f.\n\n",ec);
17     return 0;
18 }
```

$n$	$x$	$f_n(x)$
1	$\frac{3}{4}$	0.458333
2	$\frac{3}{4}$	0.525000
4	$\frac{3}{4}$	0.595238
8	$\frac{3}{4}$	0.654545
16	$\frac{3}{4}$	0.695906
32	$\frac{3}{4}$	0.721008
64	$\frac{3}{4}$	0.734962
128	$\frac{3}{4}$	0.742337



## Programa 7

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \quad (7)$$

### Ciclos

```
#include <stdio.h>

int main()
{
    int p,j,n,den,sig;
    float rex=0, rexf=0,x,numero;
    do{
        printf("\nObtener el resultado de e^x mediante iteraciones\n");
        printf("\nNumero de iteraciones: ");
        scanf("%d",&n);
        printf("Ingrese el valor de x: ");
        scanf("%f",&x);
    }while(n<0);
    for(p=0,sig=1;p<n;p++)
    {
        for(j=0,den=1,numero=1;j<p;j++)
        {
            numero*=x;
            den*=j+1;
        }
        rex=(sig*numero)/den;
        printf("%.3f",rex);
        rexf+=rex;
    }
    printf("\nEl resultado de e^%.2f es: %.3f\n",x,rexf);
    return 0;
}
```

### Recurrencia

```
#include <stdio.h>

float potencia(float x,int n);

int main()
{
    int i,j,n,de,sig;
    float re=0, rexf=0,x,numero;
    do{
        printf("\nObtener el resultado de e^x\n");
        printf("\nIngrese el numero de iteraciones: ");
        scanf("%d",&n);
        printf("Ingrese el valor de x: ");
        scanf("%f",&x);
    }while(n<0);
    numero=potencia(x,n);
    for(i=0,sig=1;i<n;i++)
    {
        for(j=0,de=1;j<i;j++)
        {
            de*=j+1;
        }
        re=(sig*numero)/de;
        printf("%.3f",re);
        rexf+=re;
    }

    printf("\nEl resultado de e^%.2f es: %.3f\n",x,rexf);
    return 0;
}

float potencia(float x,int n)
{
    int i;
    float numero;
    for(i=0,numero=1.0;i<n;i++)
    {
        numero*=x;
    }
    return numero;
}
```

## Programa 7

n	x	fn(x)	F <sub>n</sub> (x)-f <sub>n-1</sub> (x)	Calculadora
1	2	1	-	1
2	2	3	2	3
4	2	6.3333	1.3333	6.33333
8	2	7.380953	0.025397	6.72063
16	2	7.389057	0	7.389057
32	2	7.389057	0	7.389057
64	2	7.389057	0	7.389057
128	2	7.389057	0	7.389057
1	5	1	-	1
2	5	6	5	6
4	5	39.333332	20.833332	39.333
8	5	128.619049	15.500999	138.3071677
16	5	148.402924	-0.276669	148.402924
32	5	148.413177	0	148.413177
64	5	-1	-2	-1
128	5	-1	-2	-1
1	3	1	-	1
2	3	4	3	4
4	3	13	4.5	13
8	3	19.846430	0	19.846430
16	3	20.085537	43393	20.085537
32	3	20.085539	0	20.085539
64	3	20.085539	0	20.085539
128	3	-1	0	-1

## Programa 8

$$xe^x = x + \frac{2x^2}{2!} + \frac{3x^3}{3!} + \frac{4x^4}{4!} + \dots$$

(8)

### Ciclos

```
#include <stdio.h>

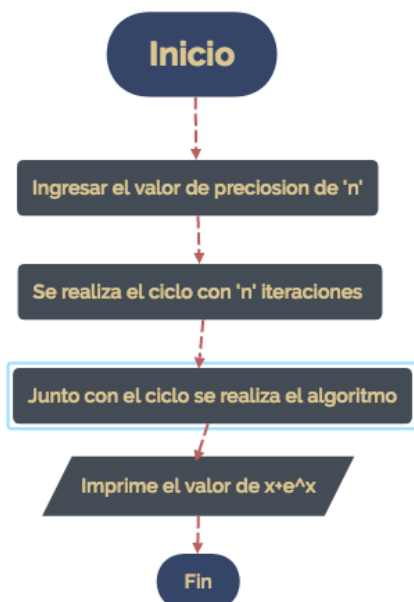
int main(int argc, char *argv[]){

    double fact, x, ex, res;
    long int i, n;

    printf("x*e^x mediante iteraciones\nIngr");
    scanf("%lg %ld", &x, &n);

    for(i=0, ex=0, fact=1; i<n; i++){
        ex+=fact;
        fact*=(x/(i+1));

        res=ex*x;
    }
    printf("exp^%lg*%lg = %lg\n", x, x, res);
    return 0;
}
```



### Recursivos

```
#include <stdio.h>

float resultado(long int n, double x);

int main(int argc, char *argv[]){

    double x;
    long int n;
    float res;

    printf("x*e^x mediante iteraciones\nIngr");
    scanf("%lg %ld", &x, &n);

    res=resultado(n, x);

    printf("exp^%lg*%lg = %f\n", x, x, res);
    return 0;
}

float resultado(long int n, double x){
    double fact, ex, r = 0.0;
    long int i;
    for(i=0, ex=0, fact=1; i<n; i++){
        ex+=fact;
        fact*=(x/(i+1));

        r=ex*x;
    }
    return r;
}
```



## Programa 8

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	6.0000	6.0000	36.0000	2420.5700
2	6.0000	42.0000	324.0000	2420.5700
4	6.0000	366.0000	1434.8600	2420.5700
8	6.0000	1800.8600	618.4800	2420.5700
16	6.0000	2419.3400	1.2300	2420.5700
32	6.0000	2420.5700	0.0000	2420.5700
64	6.0000	2420.5700	0.0000	2420.5700
128	6.0000	2420.5700	-2420.5700	2420.5700

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	2.0000	2.0000	4.0000	14.7781
2	2.0000	6.0000	6.6667	14.7781
4	2.0000	12.6667	2.0952	14.7781
8	2.0000	14.7619	0.0162	14.7781
16	2.0000	14.7781	0.0000	14.7781
32	2.0000	14.7781	0.0000	14.7781
64	2.0000	14.7781	0.0000	14.7781
128	2.0000	14.7781	-14.7781	14.7781

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	4.0000	4.0000	16.0000	14.7781
2	4.0000	20.0000	74.6667	14.7781
4	4.0000	94.6667	112.5583	14.7781
8	4.0000	207.2250	11.1670	14.7781
16	4.0000	218.3920	0.0010	14.7781
32	4.0000	218.3930	0.0000	14.7781
64	4.0000	218.3930	0.0000	14.7781
128	4.0000	218.3930	-218.3930	14.7781



## Programa 9

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \quad -1 < x \leq 1 \quad (9)$$

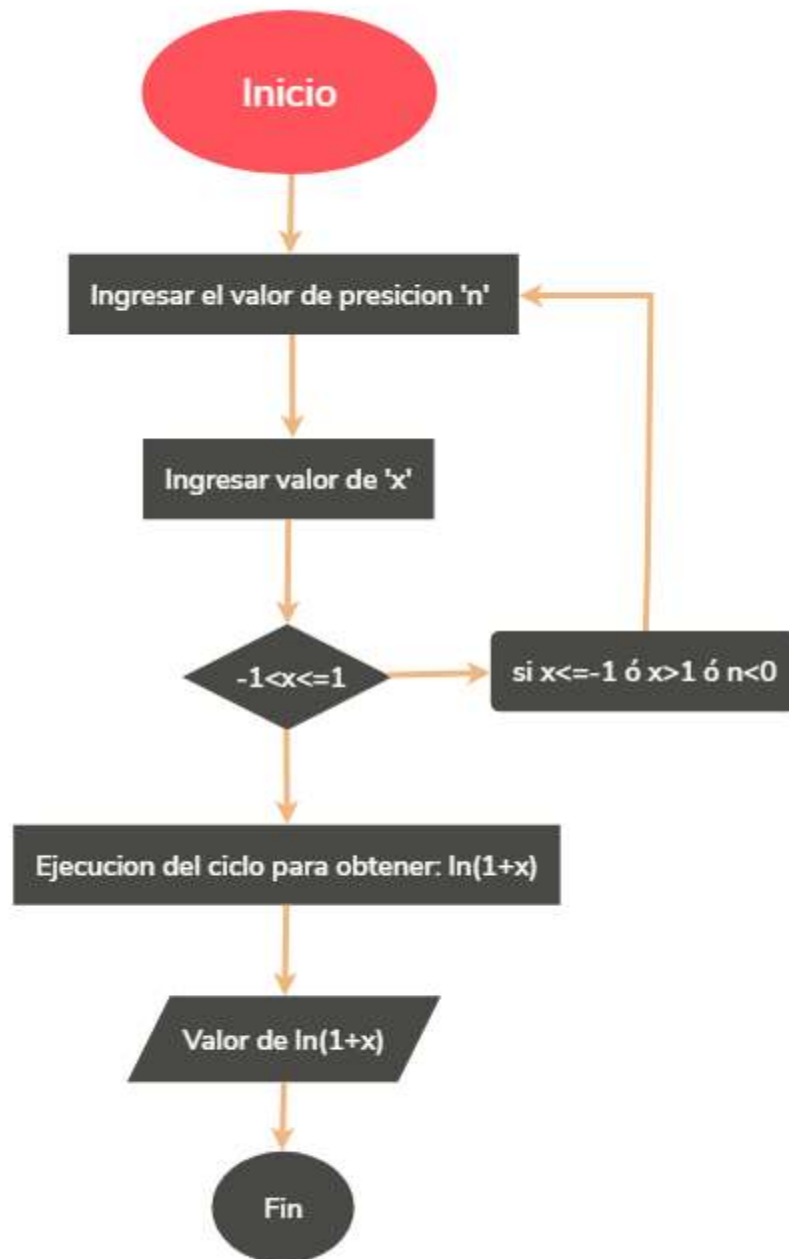
```

1  #include <stdio.h>
2
3  int main()
4  {
5      float x, ln;
6      int i, n, s=1;
7      //Ecuacion 9 ln (1+x)
8      printf("Funciones Exponenciales y Logaritmicas: Calculo de logaritmo natural de 1 + x");
9      do{
10         printf("\n\nIngrese el valor de x: ");
11         scanf("%f",&x);
12         printf("\n\nIndique la precision o numero de iteraciones: ");
13         scanf("%d",&n);
14         }while(x>1||x<-1||n<0);
15
16         for(i=1;i<n+1;i++)
17         {
18             ln+=(x/i);
19             x*=x;
20         }
21         printf("\n\nln = %f\n\n",ln);
22         return 0;
23     }

```

$n$	$x(-1 < x \leq 1)$	$f_n(x)$
1	-0.9876	-0.987600
2	-0.9876	-0.499923
4	-0.9876	0.043432
8	-0.9876	0.408630
16	-0.9876	0.413354
32	-0.9876	0.413354
64	-0.9876	0.413354
128	-0.9876	0.413354
1	0.1	0.100000
2	0.1	0.105000
4	0.1	0.105033
8	0.1	0.105033
16	0.1	0.105033

<i>32</i>	0.1	0.105033
<i>64</i>	0.1	0.105033
<i>128</i>	0.1	0.105033
<i>1</i>	0.9753	0.975300
<i>2</i>	0.9753	1.450905
<i>4</i>	0.9753	1.957171
<i>8</i>	0.9753	2.199990
<i>16</i>	0.9753	2.200174
<i>32</i>	0.9753	2.200174
<i>64</i>	0.9753	2.200174
<i>128</i>	0.9753	2.200174



## Programa 10

$$\frac{1}{2} \ln \left( \frac{1+x}{1-x} \right) = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots \quad -1 < x < 1 \quad (10)$$

Ciclos

Recurrencia

```
#include <stdio.h>
#include <math.h>

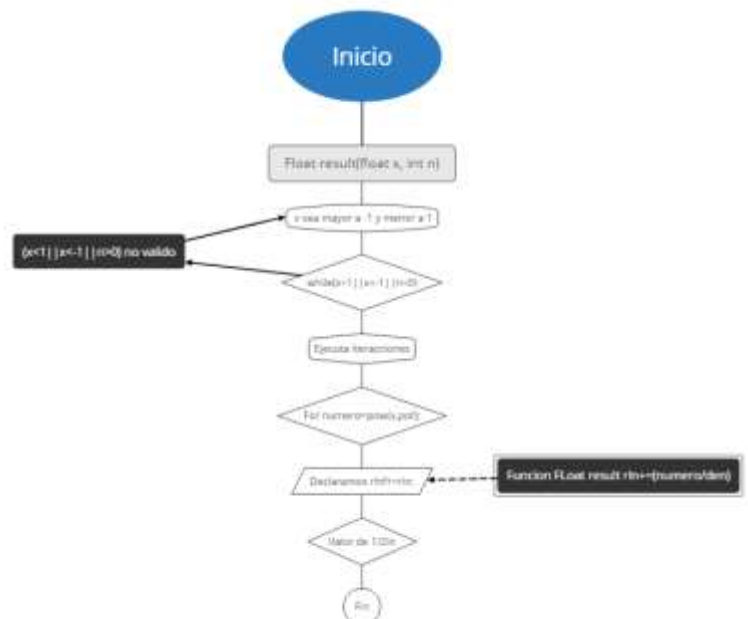
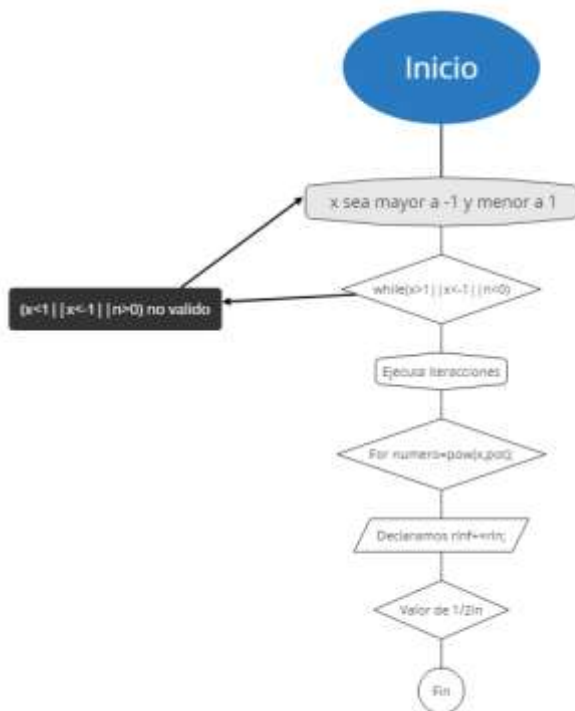
int main()
{
    int i, j, den, n;
    float x, k, l, numero, pot, rln=0, rlnf=0;
    do{
        printf("Obtener el resultado de 1/2 del logaritmo natural de (1+x/1-x) mediante iteraciones... \nSiempre y cuando x sea mayor a -1 y menor a 1\n");
        printf("Ingrese el valor de x: ");
        scanf("%f",&x);
        printf("Numero de iteraciones: ");
        scanf("%d",&n);
        while(x<1 || x>-1 || n<0){
            printf("No es valido\n");
            continue;
        }
        for(j=0; numero=1; pot=(j*(j+1)); j++)
        {
            pot=(2*(j)+1);
            numero=pow(x,pot);
        }
        den=(2*(j)+1);
        rln=(numero/den);
        printf("R^n=","rln");
        rlnf+=rln;
    } while(x<1 || x>-1 || n<0);
    printf("\nEl resultado de 1/2 del ln de (%.2f/%.2f) es: %.3f\n",x,1,x,rlnf);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>

float result(float x,int n);

int main()
{
    int n;
    float x,k,l,rlnf=0;
    do{
        printf("Obtener el resultado de 1/2 del logaritmo natural de (1+x/1-x) \nSiempre y cuando x sea mayor a -1 y menor a 1\n");
        printf("Ingrese el valor de x: ");
        scanf("%f",&x);
        printf("Numero de iteraciones: ");
        scanf("%d",&n);
        while(x>1 || x<-1 || n<0){
            printf("No es valido\n");
            continue;
        }
        k=1+x;
        l=1-x;
        printf("\nEl resultado de 1/2 del ln de (%.2f/%.2f) es: %.3f\n",k,l,rlnf);
    } while(x>1 || x<-1 || n<0);
    return 0;
}

float result(float x,int n)
{
    int i,j,den;
    float numero,pot,rln=0;
    for(i=0;i<n;i++)
    {
        for(j=0; numero=1; pot=(j*(j+1)); j++)
        {
            pot=(2*(j)+1);
            numero=pow(x,pot);
        }
        den=(2*(j)+1);
        rln+=(numero/den);
    }
    return rln;
}
```



## Programa 10

n	x	fn(x)	Calculadora
1	2	2	2
2	2	4.666667	4.666667
4	2	29.352381	29.352381
8	2	3087.110352	3087.110352
16	2	94602640.00	94602640.00
32	2	197387007477940220.00	197387007477940220.00
64	2	179589090500445940000 0000000000000000.000	179589090500445940000 0000000000000000.000
128	2	1	1
1	3	3	3
2	3	12	12
4	3	373.028564	373.028564
8	3	1097898.375000	1097898.375000
16	3	22612497399808.000	22612497399808.000
32	3	205230741688976770000 00000000.0000	205230741688976770000 00000000.0000
64	3	1	1
128	3	1	1
1	4	4	4
2	4	25.333334	25.333334
4	4	2570.704834	2570.704834
8	4	77158008.000	77158008.000
16	4	159418568868364290	159418568868364290
32	4	144351129262891660000 0000000000000000.000	144351129262891660000 0000000000000000.000
64	4	1	1
128	4	1	1

## Programa 11

$$\ln(x) = 2 \left\{ \left( \frac{x-1}{x+1} \right) + \frac{1}{3} \left( \frac{x-1}{x+1} \right)^3 + \frac{1}{5} \left( \frac{x-1}{x+1} \right)^5 + \dots \right\} \quad x > 0 \quad (11)$$

### Ciclos

```
#include <stdio.h>

int main(int argc, char *argv[]){

    float x,y,lnx, num;
    int n, i, den;

    do{
        printf("ln(x) mediante iteraciones.\nIngresa el
scanf("%f",&x);
}while(x<0);
printf("Ingresa el valor de n: ");
scanf("%d",&n);

for(i=0,y=(x-1)/(x+1),num=y,den=1,lnx=0; i<n; i++){
    lnx+=(num/den);
    num*=(y*y);
    den+=2;
}

lnx*=2;

printf("ln(%f): %f\n",x,lnx);

return 0;
}
```



### Recursividad

```
#include <stdio.h>

Float resultado(float x,int n);

int main()
{
    int n;
    float x,k,lnf=0;
    do{
        printf("ln(x) mediante iteraciones.\nIngresa el
scanf("%f",&x);
        printf("Ingresa el numero de iteraciones: ");
        scanf("%d",&n);
    }while(x>=1||x<-1||n<0);
    lnf=resultado(x,n);
    k = 1+x;
    printf("\nEl resultado del ln(%f) es: %f\n",k,lnf);

    return 0;
}

Float resultado(float x,int n)
{
    int den,sig,i,j;
    float k,px,num,ln=0.0;
    for(i=0,ln=0; i<n; i++)
    {
        sig=(1-((2)*(i%2)));
        for(j=0,px=1;j<(i+1);j++)
        {
            px*=x;
        }
        num=px;
        den=(i+1);
        ln+=(sig*num)/(den);
    }
    return ln;
}
```



# Programa 11

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	2.0000	0.6667	0.0247	0.6931
2	2.0000	0.6914	0.0018	0.6931
4	2.0000	0.6931	0.0000	0.6931
8	2.0000	0.6931	0.0000	0.6931
16	2.0000	0.6931	0.0000	0.6931
32	2.0000	0.6931	0.0000	0.6931
64	2.0000	0.6931	0.0000	0.6931
128	2.0000	0.6931	-0.6931	0.6931

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	4.0000	1.2000	0.1440	1.3863
2	4.0000	1.3440	0.0391	1.3863
4	4.0000	1.3831	0.0032	1.3863
8	4.0000	1.3863	0.0000	1.3863
16	4.0000	1.3863	0.0000	1.3863
32	4.0000	1.3863	0.0000	1.3863
64	4.0000	1.3863	0.0000	1.3863
128	4.0000	1.3863	-1.3863	1.3863

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	6.0000	1.4286	0.2430	1.7918
2	6.0000	1.6715	0.1015	1.7918
4	6.0000	1.7730	0.0180	1.7918
8	6.0000	1.7910	0.0007	1.7918
16	6.0000	1.7918	0.0000	1.7918
32	6.0000	1.7918	0.0000	1.7918
64	6.0000	1.7918	0.0000	1.7918
128	6.0000	1.7918	-1.7918	1.7918

## Programa 12

$$\ln(x) = \left(\frac{x-1}{x}\right) + \frac{1}{2} \left(\frac{x-1}{x}\right)^2 + \frac{1}{3} \left(\frac{x-1}{x}\right)^3 + \dots \quad x \geq \frac{1}{2} \quad (12)$$

Tabla y Diagrama Programa 12 -  $\ln(x)$        $x \geq \frac{1}{2}$

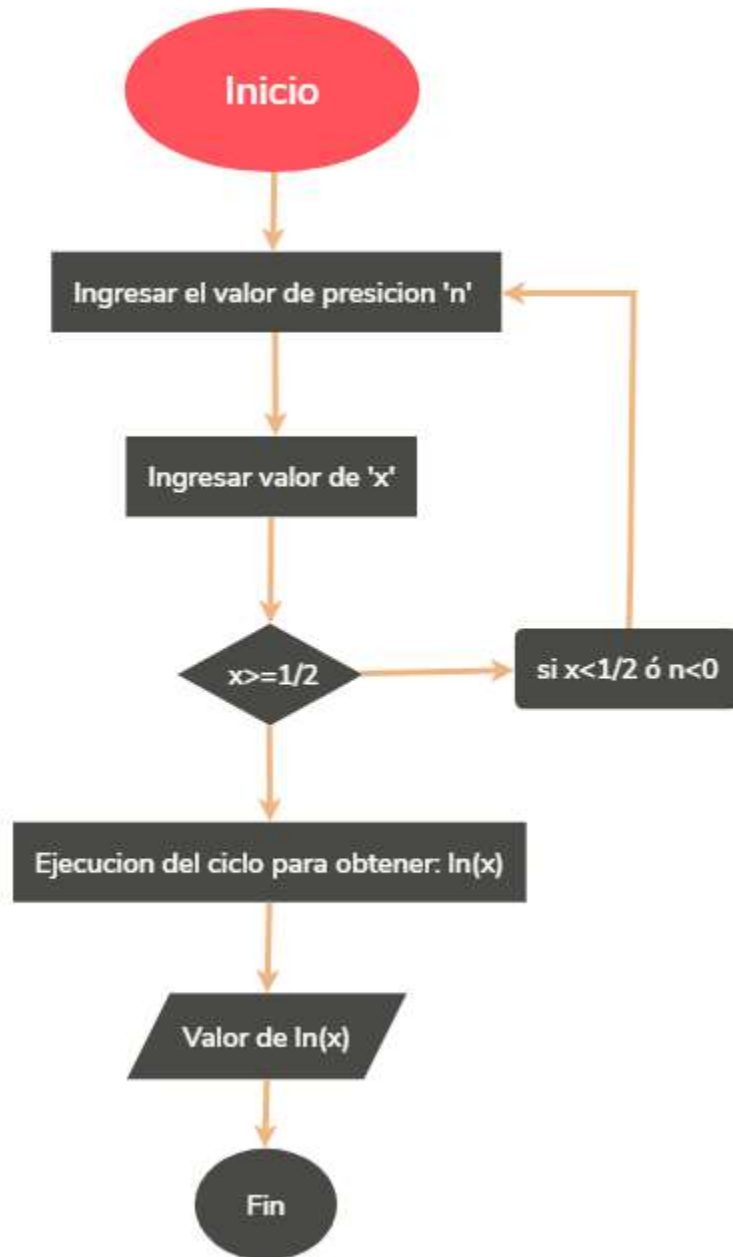
```

1  #include <stdio.h>
2
3  int main()
4  {
5      float ln,aux=1,aux2=0,x;
6      int i,n;
7      //Ecuacion 12 Ln (x)
8      do{
9          printf("Funciones Exponenciales y Logaritmicas: Calculo de logaritmo natural de x");
10         printf("\n\nIngrese el valor de x: ");
11         scanf("%f",&x);
12         printf("\n\nIndique la precision o numero de iteraciones: ");
13         scanf("%d",&n);
14     }while (x<0.5 || n<8);
15
16     ln=x-1;
17     for(i=1;i<n+1;i++)
18     {
19         aux*=(ln/x);
20         aux2+=aux*1/i;
21     }
22     printf("\n\nln(%f) = %f\n\n",x,aux2);
23     return 0;
24 }
```

$n$	$x$	$f_n(x)$
1	0.5050	-0.980198
2	0.5050	-0.499804
4	0.5050	-0.582946
8	0.5050	-0.633770
16	0.5050	-0.661439
32	0.5050	-0.675169
64	0.5050	-0.681063



<i>128</i>	0.5050	−0.682899
<i>1</i>	48.963	0.979576
<i>2</i>	48.963	1.459361
<i>4</i>	48.963	2.002879
<i>8</i>	48.963	2.560152
<i>16</i>	48.963	3.077465
<i>32</i>	48.963	3.495733
<i>64</i>	48.963	3.761936
<i>128</i>	48.963	3.870638
<i>1</i>	377.531	0.997351
<i>2</i>	377.531	1.494706
<i>4</i>	377.531	2.072759
<i>8</i>	377.531	2.696765
<i>16</i>	377.531	3.338776
<i>32</i>	377.531	3.975444
<i>64</i>	377.531	4.581191
<i>128</i>	377.531	5.120628



## Programa 14

$$a^x = e^{x \ln a} = 1 + x \ln a + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \dots \quad (14)$$

### Ciclos

```
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[])
{
    int i,j,n,den,denln;
    float x,a,k,h,pot,num=1.0,numln,rln=0.0,rlnf=0.0,ra=0.0,raf=0.0,rafi=0.0;
    do{
        printf("a^x mediante iteraciones\n");
        printf("\nIngrese el valor a: ");
        scanf("%f",&a);
        printf("Ingrese el valor x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0);
    for(i=0;i<n;i++)
    {
        for(j=0,numln=1.0,pot=0.0,den=1,denln=1;j<i+1;j++)
        {
            pot=((2*(j))+1);
            h=(a-1.0)/(a+1.0);
            numln=pow(h,pot);
            den*=(j+1);
        }
        denln=(2*(i)+1);
        rln+=(numln/denln);
        rlnf=2.0*rln;
        k=x*rlnf;
        num*=k;
        ra=(num/den);
        raf+=ra;
        rafi=1+raf;
    }
    printf("\nEl resultado de %.2f%.2f es: %.4f\n",a,x,rafi);

    return 0;
}
```

### Recursividad

```
#include <stdio.h>
#include <math.h>

float resultado(float x,float a,int n);

int main()
{
    int n;
    float x,a,rafi=0.0;
    do{
        printf("a^x mediante iteraciones\n");
        printf("\nIngrese el valor a: ");
        scanf("%f",&a);
        printf("Ingrese el valor x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0);
    rafi=resultado(x,a,n);
    printf("\nEl resultado de %.2f%.2f es: %.4f\n",a,x,rafi);

    return 0;
}

float resultado(float x,float a,int n)
{
    int i,j,den,denln;
    float k,h,pot,num=1.0,numln,rln=0.0,rlnf=0.0,ra=0.0,raf=0.0,rafi=0.0;
    for(i=0;i<n;i++)
    {
        for(j=0,numln=1.0,pot=0.0,den=1,denln=1;j<i+1;j++)
        {
            pot=((2*(j))+1);
            h=(a-1.0)/(a+1.0);
            numln=pow(h,pot);
            den*=(j+1);
        }
        denln=(2*(i)+1);
    }
}
```



#### Programa 14

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	2,2	2.3330	0.9221	3.8813
2	2,2	3.2551	0.5735	3.8813
4	2,2	3.8286	0.0526	3.8813
8	2,2	3.8812	0.0001	3.8813
16	2,2	3.8813	0.0000	3.8813
32	2,2	3.8813	0.0000	3.8813
64	2,2	3.8813	0.0000	3.8813
128	2,2	3.8813	-3.8813	3.8813

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	4,4	5.8000	12.9024	621.4116
2	4,4	18.7024	56.3751	621.4116
4	4,4	75.0775	114.7036	621.4116
8	4,4	189.7811	431.6305	621.4116
16	4,4	621.4116	0.0000	621.4116
32	4,4	621.4116	0.0000	621.4116
64	4,4	621.4116	0.0000	621.4116
128	4,4	621.4116	-621.4116	621.4116

n	x a	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	6,6	9.5714	42.9821	8477.6738
2	6,6	52.5535	549.2364	8477.6738
4	6,6	601.7899	7875.8839	8477.6738
8	6,6	8477.6738	0.0000	8477.6738
16	6,6	8477.6738	0.0000	8477.6738
32	6,6	8477.6738	0.0000	8477.6738
64	6,6	8477.6738	0.0000	8477.6738
128	6,6	8477.6738	-8477.6738	8477.6738

## Programa 17

$$E_{2k} = i \sum_{m=1}^{2k+1} \sum_{j=0}^m \binom{m}{j} \frac{(-1)^j (m-2j)^{2k+1}}{2^m i^m m} \quad (17)$$
$$i^2 = -1$$

## Ciclos

```
#include <stdio.h>

int main()
{
    int it, j, sig=-1, nden=3;
    float div=0, pi=1, i, num, den=1, rest, fact=1, pot=2;
    printf("Ingrese K ");
    scanf("%d", &it);
    printf("Ingrese n");
    scanf("%f", &num);
    num*=2;
    rest=num;
    for(i=1;i<=num;i++)
    {
        fact*=rest;
        rest-=1;
        pi*=3.1416;
        pot*=2;
    }
    pi*=3.1416;
    pot*=2;
    fact/=pi;
    for (i=1;i<=it;i++)
    {
        for(j=0;j<=num;j++)
            den*=nden;
        div+=(sig*(1/den));
        den=1;
        sig*=-1;
        nden+=2;
    }
    div+=1;
    fact*=div;
    printf("%f", fact);
    return 0;
}
```



# Programa 17

n	k	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	2.00	1.0021	3.9981	2.7183
2	2.00	5.0002	1379.9708	2.7183
4	2.00	1384.9709	#VALUE!	2.7183
8	2.00	inf	#VALUE!	2.7183
16	2.00	inf	#VALUE!	2.7183
32	2.00	inf	#VALUE!	2.7183
64	2.00	inf	#VALUE!	2.7183
128	2.00	inf	#VALUE!	2.7183

n	k	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	6.00	1.0006	4.0003	2.7183
2	6.00	5.0009	2299.2443	2.7183
4	6.00	2304.2452	#VALUE!	2.7183
8	6.00	inf	#VALUE!	2.7183
16	6.00	inf	#VALUE!	2.7183
32	6.00	inf	#VALUE!	2.7183
64	6.00	inf	#VALUE!	2.7183
128	6.00	inf	#VALUE!	2.7183

n	k	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	4.00	1.0005	3.9998	2.7183
2	4.00	5.0003	1297.3240	2.7183
4	4.00	1302.3243	#VALUE!	2.7183
8	4.00	inf	#VALUE!	2.7183
16	4.00	inf	#VALUE!	2.7183
32	4.00	inf	#VALUE!	2.7183
64	4.00	inf	#VALUE!	2.7183
128	4.00	inf	#VALUE!	2.7183

## Programa 18

$$\text{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (18)$$

Tabla y Diagrama Programa 18.

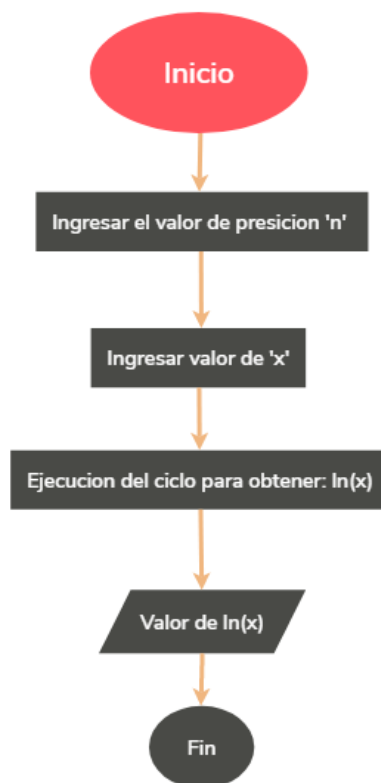
```

1 //funcion: seno.c
2
3 int main()
4 {
5     int i,n,s=1,fac=1;
6     float x,aux,r,sen=0;
7     //Ecuacion 18 sen (x)
8     printf("Funciones Trigonometricas: Calculo de sen (x)");
9     //La precision varia mucho, dependiendo del valor de 'x' y del numero de iteraciones que el usuario ingrese
10    printf("\nIngrese el valor de x: ");
11    scanf("%f",&x);
12    printf("\nIngrese la precision o numero de iteraciones: ");
13    scanf("%d",&n);
14
15    aux=x;
16    for(i=1;i<=n;i++)
17    {
18        fac*=i;
19        if(i%2==0)
20        {
21            sen+=(aux*s)/fac;
22            aux*=(x*x);
23            s*=-1;
24        }
25    }
26    r=sen;
27    printf("\n\nsen (%f) = %f",x,r);
28    return 0;
29 }

```

$n$	$x$	$f_n(x)$
1	-2	-2.000000
2	-2	-2.000000
4	-2	-0.666667
8	-2	-0.907936
16	-2	-0.909284
32	-2	2.523191
64	-2	-1.#IND00
128	-2	-1.#IND00
1	0.4137	0.413700
2	0.4137	0.413700
4	0.4137	0.401899
8	0.4137	0.402000
16	0.4137	0.402000

32	0.4137	0.402000
64	0.4137	-1.#IND00
128	0.4137	-1.#IND00
1	19	19.000000
2	19	19.000000
4	19	-1124.166626
8	19	-157845.500000
16	19	-7554661888.000000
32	19	-1.#IND00
64	19	-1.#IND00
128	19	-1.#IND00





## Programa 19

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (19)$$

### Ciclos

```
#include <stdio.h>

int main()
{
    int n,i;
    float cos,fact,x;
    do{
        printf("Obtener el resultado de cos(x)\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0||x<0);
    for(i=0,fact=1,cos=0.0;i<n;i++)
    {
        cos+=fact;
        fact*=(-1.0)*((x/(4*i+3))*(x/(4*i+4)));
    }
    printf("\nEl resultado de cos(%.3f) es: %.5f\n",x,cos);
    return 0;
}
```

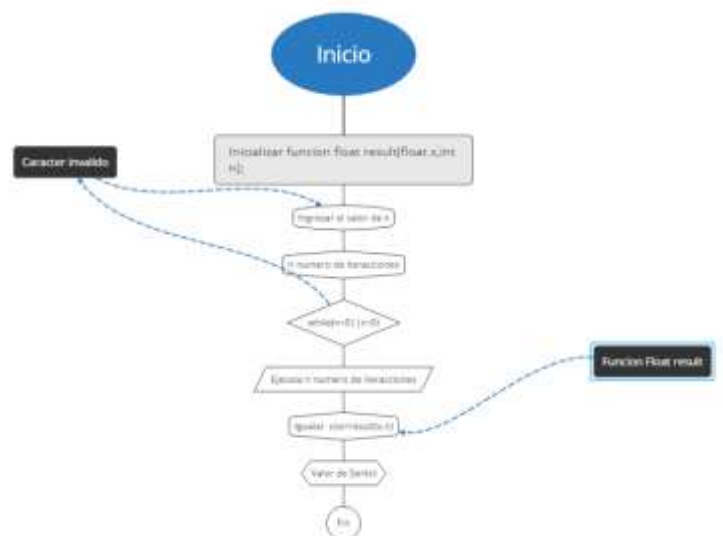
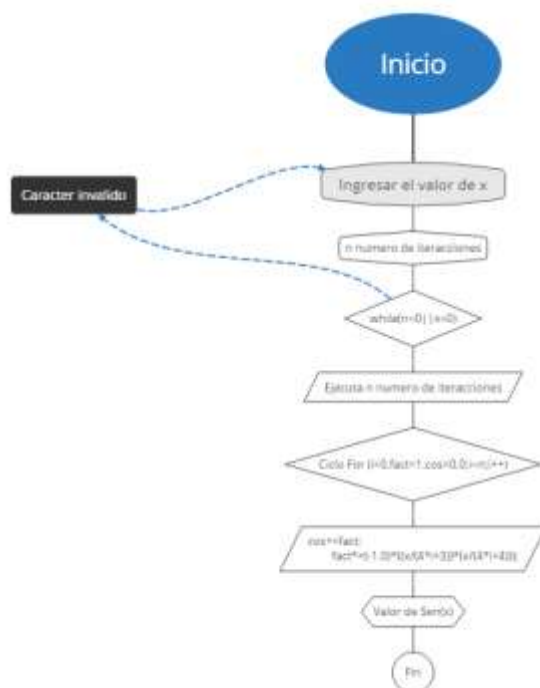
### Recurrencia

```
#include <stdio.h>

float result(float x,int n);

int main()
{
    int n;
    float cos,x;
    do{
        printf("Obtener el resultado de cos(x) mediante iteracciones\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Numero de iteracciones: ");
        scanf("%d",&n);
    }while(n<0||x<0);
    cos=result(x,n);
    printf("\nEl resultado de cos(%.3f) es: %.5f\n",x,cos);
    return 0;
}

float result(float x,int n)
{
    int i;
    float rcos,fact;
    for(i=0,fact=1,rcos=0.0;i<n;i++)
    {
        rcos+=fact;
        fact*=(-1.0)*((x/(4*i+3))*(x/(4*i+4)));
    }
    return rcos;
}
```



$\alpha$	0	$\pi/6$	$\pi/4$	$\pi/3$	$\pi/2$	$\pi$	$3\pi/2$	$2\pi$
$\cos \alpha$	1	$\sqrt{3}/2$	$\sqrt{2}/2$	1/2	0	-1	0	1

Programa 20

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \dots + \frac{B_{2n}(-4)^n(1-4^n)}{(2n)!}x^{2n-1} + \dots \quad |x| < \frac{\pi}{2}$$

(20)

```
#include <stdio.h>

int main(void)
{
    int i, n;
    float sx, x, nu,tan,cx;
    do{
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d", &n);
    } while (n<1);
    printf("Ingrese el valor de x: ");
    scanf("%f", &x);
    for (i=0, sx=0, nu=x; i<n; i++, nu*=(-1*x*x/((2*i)*(2*i+1))))
        sx += nu;
    for (i=0, cx=0, nu=1; i<n; i++, nu*=(-1*x*x/((2*i-1)*(2*i))))
        cx += nu;
    tan=sx/cx;
    printf("tan(%f) = %f\n", x, tan);
    return 0;
}
```



## Programa 20

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	2.00	2.0000	-2.6667	-2.1850
2	2.00	-0.6667	-1.4837	-2.1850
4	2.00	-2.1504	-0.0347	-2.1850
8	2.00	-2.1850	0.0000	-2.1850
16	2.00	-2.1850	0.0000	-2.1850
32	2.00	-2.1850	0.0000	-2.1850
64	2.00	-2.1850	0.0000	-2.1850
128	2.00	-2.1850	2.1850	-2.1850

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	4.00	4.0000	-3.0476	1.1575
2	4.00	0.9524	-0.2679	1.1575
4	4.00	0.6845	0.4731	1.1575
8	4.00	1.1575	0.0000	1.1575
16	4.00	1.1575	0.0000	1.1575
32	4.00	1.1575	0.0000	1.1575
64	4.00	1.1575	0.0000	1.1575
128	4.00	1.1575	-1.1575	1.1575

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	6.00	1.5574	0.2073	-0.2910
2	6.00	1.7647	-1.0186	-0.2910
4	6.00	0.7461	-1.1300	-0.2910
8	6.00	-0.3839	0.0929	-0.2910
16	6.00	-0.2910	0.0000	-0.2910
32	6.00	-0.2910	0.0000	-0.2910
64	6.00	-0.2910	0.0000	-0.2910
128	6.00	-0.2910	0.2910	-0.2910

## Programa 22

$$\csc(x) = \frac{1}{x} + \frac{x}{6} + \frac{7x^3}{360} + \frac{31x^5}{15,120} +$$

Ciclos

Recurrencia

```
#include <stdio.h>
#include <stdlib.h>

float potencia(float, float);
float factorial(float);
float sen(float);

int main(int argc, char** argv) {

    float x, seno, csc;

    printf("Funcion trigonometrica\n");
    printf("Cosecante\n");
    printf("Dame un numero\n");
    scanf("%f", &x);
    csc = 1/sen(x);
    printf("La cosecante de %.2f es %.2f", x,csc);
    return 0;

}

float factorial(float k) {
    float z = 1;
    int i;

    for (i = 1; i <= k; i++) {
        z *= i;
    }

    float potencia(float x, float k) {
        float z = 1;
        int i;

        if (k == 0) {
            return (1);
        } else {
            for (i = 0; i < k; i++) {
                z *= x;
            }
        }
    }

    return (z);
}

float sen(float x) {
    float suma = 0, ax, error = 0.001;
    int k;

    do {
        ax = (((potencia(-1, k) * potencia(x, 2 * k)) + 1) / factorial((2 * k) + 1));
        suma += ax;
        k++;
        if (ax < 0) {
            ax = -ax;
        }
    } while (ax > error);
    return (suma);
}
```

```
#include <stdio.h>
#include <stdlib.h>

float potencia(float, float);
float factorial(float);
float exponencial(float);
float sen(float);

float potencia(float x, float k) {
    float z = 1;
    int i;

    if (k == 0) {
        return (1);
    } else {
        for (i = 0; i < k; i++) {
            z *= x;
        }
    }

    return (z);
}

float factorial(float k) {
    float z = 1;
    int i;

    for (i = 1; i <= k; i++) {
        z *= i;
    }

    return (z);
}

float exponencial(float x) {
    float suma = 0, ax, error = 0.0001;
    int k = 0;

    do {
        ax = (potencia(x, k) / factorial(k));
        suma += ax;
        k++;
        if (ax < 0) {
            ax = -ax;
        }
    } while (ax > error);
    return (suma);
}

float sen(float x) {
    float suma = 0, ax, error = 0.001;
    int k;

    do {
        ax = (((potencia(-1, k) * potencia(x, 2 * k)) + 1) / factorial((2 * k) + 1));
        suma += ax;
        k++;
        if (ax < 0) {
            ax = -ax;
        }
    } while (ax > error);
    return (suma);
}
```

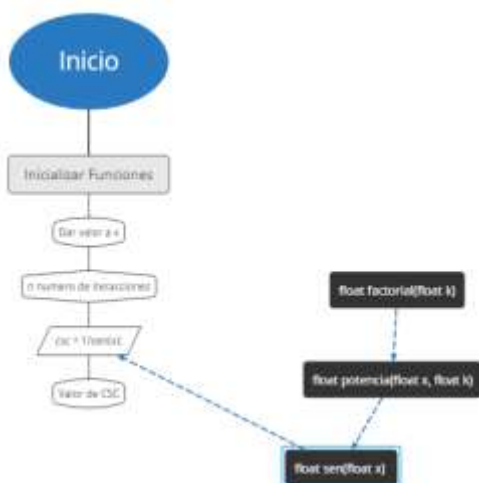


Tabla Cosecante:

Grados	0	30	45	60	90	135	180	225	270	315
Radianes	0	$1/6\pi$	$1/4\pi$	$1/3\pi$	$1/2\pi$	$3/4\pi$	$\pi$	$5/4\pi$	$3/2\pi$	$7/4\pi$
CSC	$+\text{inf}$	2	$\sqrt{2}$	$\frac{2\sqrt{3}}{3}$	1	$\sqrt{2}$	$+\text{inf}$	$-\sqrt{2}$	-1	$-\sqrt{2}$

## Programa 23

$$\text{sen}^{-1}(x) = x + \frac{1}{2} \frac{x^3}{3} + \frac{1}{2} \frac{3}{4} \frac{x^5}{5} + \frac{1}{2} \frac{3}{4} \frac{5}{6} \frac{x^7}{7}$$

```
#include <stdio.h>
#include <math.h>

int main()
{
    int i,j,den = 0,denk = 0,n;
    float x,pot,num = 0.0,numk = 0.0,pf,k=1.0,rsen=0.0,rsenf=0.0;
    do{
        printf("Sen^-1(x) mediante iteracciones... \nSiempre y cua
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteracciones: ");
        scanf("%d",&n);
    }while(x>1 || n<0);
    for(i=0;i<n;i++)
    {
        for(j=0;j<i+1;j++)
        {
            numk=((2*(j)+1));
            denk=((2*(j)+2));
            pot=((2*(j)+1));
            num=pow(x,pot);
            den=((2*(j)+1));
        }
        k*=(numk/denk);
        pf=(num/den);
        rsen=(k*pf);
        rsenf+=rsen;
    }
    printf("El resultado de sen^-1(%.3f) es: %.6f ",x,rsenf);
    return 0;
}
```



### Programa 23

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	-1.00	-0.5000	-0.1250	-1.5797
2	-1.00	-0.6250	-0.1016	-1.5797
4	-1.00	-0.7266	-0.0771	-1.5797
8	-1.00	-0.8036	-0.0564	-1.5797
16	-1.00	-0.8601	-0.0406	-1.5797
32	-1.00	-0.9007	-0.0495	-1.5797
64	-1.00	-0.9502	-0.6295	-1.5797
128	-1.00	-1.5797	1.5797	-1.5797

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	-3.00	-1.5000	-3.3750	null
2	-3.00	-4.8750	-100.6172	null
4	-3.00	-105.4922	#VALUE!	null
8	-3.00	inf	#VALUE!	null
16	-3.00	inf	#VALUE!	null
32	-3.00	inf	#VALUE!	null
64	-3.00	inf	#VALUE!	null
128	-3.00	inf	#VALUE!	null

n	x	F <sub>n</sub> (x)	F <sub>n</sub> (x) - F <sub>n-1</sub> (x)	Calculadora
1	-6.00	-2.7500	-5.8150	null
2	-6.00	-8.5650	-201.9272	null
4	-6.00	-210.4922	#VALUE!	null
8	-6.00	inf	#VALUE!	null
16	-6.00	inf	#VALUE!	null
32	-6.00	inf	#VALUE!	null
64	-6.00	inf	#VALUE!	null
128	-6.00	inf	#VALUE!	null

## Programa 24

$$\cos^{-1}(x) = \frac{\pi}{2} - \sin^{-1}(x) = \frac{\pi}{2} - \left( x + \frac{1}{2} \frac{x^3}{3} + \frac{1}{2} \frac{3}{4} \frac{x^5}{5} + \dots \right) \quad |x| < 1 \quad (24)$$

Tabla y Diagrama Programa 24 ~  $\cos^{-1}(x)$ .

```

1  #include <stdio.h>
2  #define PI 3.141592
3  int main()
4  {
5      int i,n;
6      float x,aux,cos=0,r,u=1,v=1,w=2,p;
7      //Ecuacion 24 cos^-1 (x)
8      printf("Funciones Trigonometricas: Calculo de cos^-1 (x)");
9      printf("\n\nIngrese el valor de x (|x| < 1): ");
10     scanf("%f",&x);
11     printf("\nIngrese la precision o numero de iteraciones: ");
12     scanf("%d",&n);
13
14     aux=x;
15     for(i=1;i<n+1;i++)
16     {
17         if(i%2!=0)
18         {
19             cos+=(aux/i)*u;
20             aux*=(aux*aux);
21             u*=(v/w);
22             v+=2;
23             w+=2;
24         }
25     }
26     p=PI/2;
27     r=p-cos;
28     printf("\n\ncos^-1 (%f) = %f",x,r);
29     return 0;

```

$n$	$x$	$f_n(x)$
1	0.13487	1.435926
2	0.13487	1.435926
4	0.13487	1.435517
8	0.13487	1.435517
16	0.13487	1.435517
32	0.13487	1.435517
64	0.13487	1.435517
128	0.13487	1.435517

1	0.9831	0.587696
2	0.9831	0.587696
4	0.9831	0.429337
8	0.9831	0.336827
16	0.9831	0.328832
32	0.9831	0.328832
64	0.9831	0.328832
128	0.9831	0.328832
1	0.5555	1.015296
2	0.5555	1.015296
4	0.5555	0.986727
8	0.5555	0.986349
16	0.5555	0.986349
32	0.5555	0.986349
64	0.5555	0.986349
128	0.5555	0.986349





## Programa 25

$$\tan^{-1}(x) = \begin{cases} x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots & |x| < 1 \\ \pm \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots & [+ \text{ si } x \geq 1, - \text{ si } x \leq -1] \end{cases} \quad (25)$$

Ciclos:

Recurrencia:

```

#include<stdio.h>
#include<math.h>

int main()
{
    int i,j,deno,r,signo;
    float x,numero,pot,rtan=0.0,rtangf=0.0;
    do{
        printf("Tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea menor a 1\n");
        printf("Ingrese el valor de x: ");
        scanf("%f",&x);
        printf("Numero de iteraciones: ");
        scanf("%d",&n);
        while(x>1||x<-1){
            for(i=0;i<n;i++){
                signo=((2)*(i%2))-1;
                for(j=0,num=1,pot=0;j<(i+1);j++){
                    pot=(2*(j)+1);
                    num=pow(x,pot);
                }
                deno=(2*(i)+1);
                rtan=(signo*num/deno);
                printf("%f",rtan);
                rtangf+=rtan;
            }
        }
        printf("\nEl resultado de tan^-1 de (%.2f) es: %.3f\n",x,rtangf);

        int k,l,e,signo;
        float y,deno,pote,rtang=0.0,rtangf=0.0,rtangfi=0.0;
        do{
            printf("Resultado de la tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea mayor o igual a 1\n");
            printf("Ingrese el valor de x: ");
            scanf("%f",&y);
            printf("Numero de iteraciones: ");
            scanf("%d",&m);
            while(y<1||y>1){
                signo=((2)*(k%2))-1;
                for(l=0,pote=0;l<(k+1);l++){
                    pote=(2*(l)+1);
                    deno=(2*(l)+1)*(pow(y,pote));
                }
                rtang=(signo*1.0/deno);
                printf("%f",rtang);
                rtangf+=rtang;
                rtangfi=(rtang+1.570796327);
            }
            printf("\nEl resultado de tan^-1 de (%.2f) es: %.3f\n",y,rtangfi);

            int r,s,e,signo;
            float z,denom,poten,rtange=0.0,rtangef=0.0,rtangefi=0.0;
            do{
                printf("Resultado de la tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea menor o igual a 1\n");
                printf("Ingrese el valor de x: ");
                scanf("%f",&z);
                printf("Numero de iteraciones: ");
                scanf("%d",&o);
                while(z>1||z<-1){
                    for(r=0,rco=r+1){
                        signo=((2)*(r%2))-1;
                        for(s=0,poten=0;s<(r+1);s++){
                            poten=(2*(s)+1);
                            denom=(2*(s)+1)*(pow(z,poten));
                        }
                        rtange=(signo*1.0/denom);
                        printf("%f",rtange);
                        rtangef+=rtange;
                        rtangefi=(rtange-1.570796327);
                    }
                }
                printf("\nEl resultado de tan^-1 de (%.2f) es: %.3f\n",z,rtangefi);
                return 0;
            }
        }
    }
}

```

```

#include<stdio.h>
#include<math.h>

float resultado(float x,int n)
{
    float resultado(float x,int n);
    float resultado(float y,int m);
    float resultado(float z,int o);

    int main()
    {
        int n;
        float x,rtangf;
        do{
            printf("Obtener el resultado de la tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea menor a 1\n");
            printf("Ingrese el valor de x: ");
            scanf("%f",&x);
            printf("Ingrese el numero de iteraciones: ");
            scanf("%d",&n);
            while(x>1||x<-1){
                rtangf=resultado(x,n);
                printf("\nEl resultado de tan^-1 de (%.2f) es: %.3f\n",x,rtangf);
            }

            int m;
            float y,rtangfi;
            do{
                printf("Obtener el resultado de la tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea mayor o igual a 1\n");
                printf("Ingrese el valor de x: ");
                scanf("%f",&y);
                printf("Ingrese el numero de iteraciones: ");
                scanf("%d",&m);
                while(y<1||y>1){
                    rtangfi=resultado2(y,m);
                }
            }

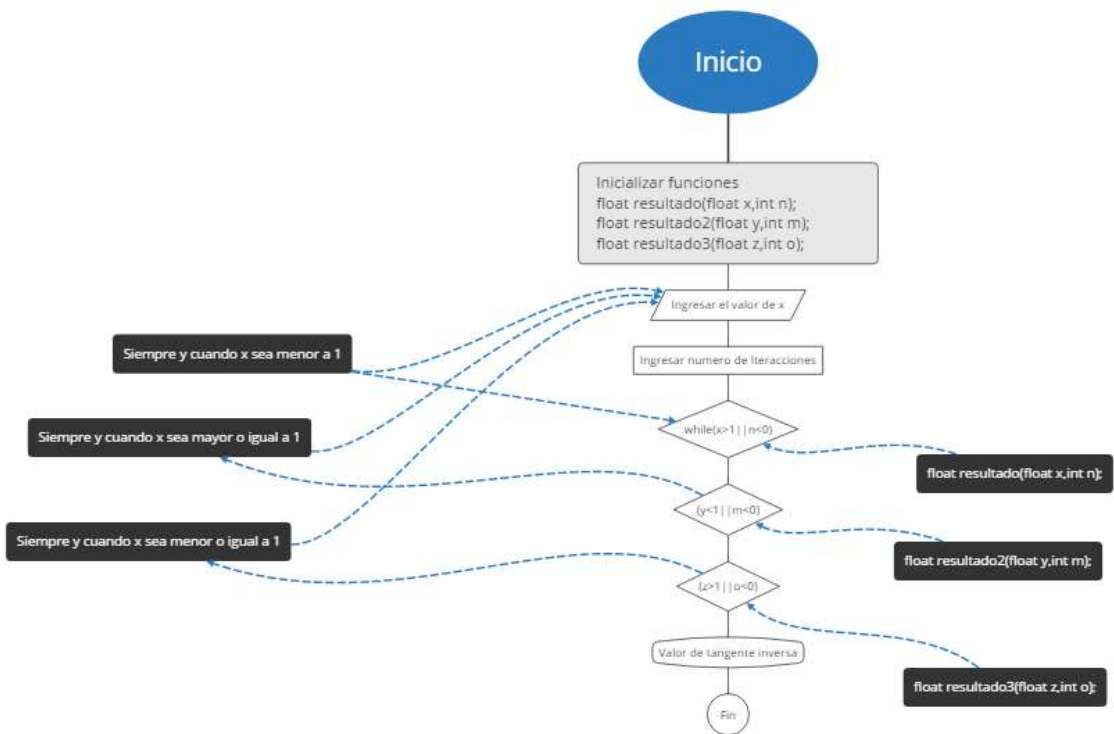
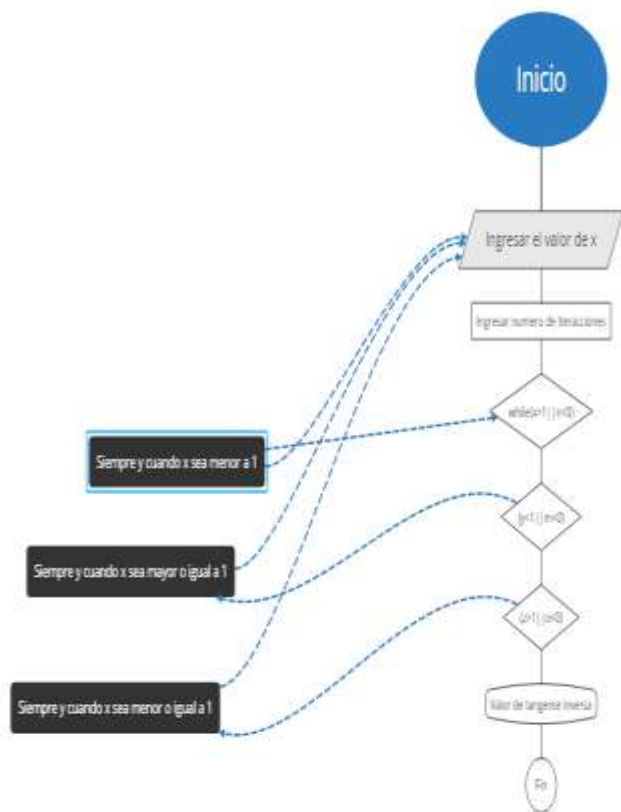
            float resultado2(float y,int m)
            {
                int k,l,signo;
                float deno,pote,rtang=0.0,rtangf=0.0;
                for(k=0;k<m;k++){
                    signo=((2)*(k%2))-1;
                    for(l=0,pote=0;l<(k+1);l++){
                        pote=(2*(l)+1);
                        deno=(2*(l)+1)*(pow(y,pote));
                    }
                    rtang+=(signo*1.0/deno);
                    rtangf=(rtang+1.570796327);
                }
                return rtangf;
            }

            float resultado3(float z,int o)
            {
                int r,s,signo;
                float denom,poten,rtange=0.0,rtangef=0.0,rtangefi=0.0;
                for(r=0;r<o;r++){
                    signo=((2)*(r%2))-1;
                    for(s=0,poten=0;s<(r+1);s++){
                        poten=(2*(s)+1);
                        denom=(2*(s)+1)*(pow(z,poten));
                    }
                    rtange+=(signo*1.0/denom);
                    rtangef=(rtange-1.570796327);
                }
                return rtangef;
            }

            int o;
            float z,rtangefi;
            do{
                printf("Obtener el resultado de la tan^-1 de (x) mediante iteraciones... \nSiempre y cuando x sea menor o igual a 1\n");
                printf("Ingrese el valor de x: ");
                scanf("%f",&z);
                printf("Ingrese el numero de iteraciones: ");
                scanf("%d",&o);
                while(z>1||z<-1){
                    rtangefi=resultado3(z,o);
                    printf("\nEl resultado de tan^-1 de (%.2f) es: %.3f\n",z,rtangefi);
                    return 0;
                }
            }

            float resultado(float x,int n)
            {
                int i,j,den,signo;
                float num,pot,rtan=0.0;
                for(i=0;i<n;i++){
                    signo=((2)*(i%2))-1;
                    for(j=0,num=1,pot=0;j<(i+1);j++){
                        pot=(2*(j)+1);
                        num=pow(x,pot);
                    }
                }
            }
        }
    }
}

```



## Programa 26

$$\sinh(x) = x + \frac{1}{3!}x^3 + \frac{1}{5!}x^5 + \frac{1}{7!}x^7 + \dots + \frac{1}{(2n+1)!}x^{2n+1} \quad (26)$$

### Ciclo

```
#include <stdio.h>

int main()
{
    int n,i;
    float rsen,fact,x;
    do{
        printf("Sinh(x) mediante iteracciones.\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteracciones: ");
        scanf("%d",&n);
    }while(n<0||x<0);
    for(i=0,fact=x,rsen=0.0;i<n;i++)
    {
        rsen+=fact;
        fact*=((x/(2*i+2))*(x/(2*i+3)));
    }
    printf("\nEl resultado de sinh(%.3f) es: %.5f\n",x,rsen);
    return 0;
}
```



### Recursiva

```
#include <stdio.h>

float resultado(float x,int n);

int main()
{
    int n;
    float rseno,x;
    do{
        printf("Sinh(x) mediante iteracciones\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteracciones: ");
        scanf("%d",&n);
    }while(n<0||x<0);
    rseno=resultado(x,n);
    printf("\nEl resultado de sinh(%.3f) es: %.5f\n",x,rseno);
    return 0;
}

float resultado(float x,int n)
{
    int i;
    float rsen,fact;
    for(i=0,fact=x,rsen=0.0;i<n;i++)
    {
        rsen+=fact;
        fact*=((x/(2*i+2))*(x/(2*i+3)));
    }
    return rsen;
}
```



## Programa 26

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	2.00	2.0000	1.3333	3.6268
2	2.00	3.3333	0.2921	3.6268
4	2.00	3.6254	0.0014	3.6268
8	2.00	3.6268	0.0000	3.6268
16	2.00	3.6268	0.0000	3.6268
32	2.00	3.6268	0.0000	3.6268
64	2.00	3.6268	0.0000	3.6268
128	2.00	3.6268	-3.6268	3.6268

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	4.00	4.0000	10.6667	27.2899
2	4.00	14.6667	11.7841	27.2899
4	4.00	26.4508	0.8391	27.2899
8	4.00	27.2899	0.0000	27.2899
16	4.00	27.2899	0.0000	27.2899
32	4.00	27.2899	0.0000	27.2899
64	4.00	27.2899	0.0000	27.2899
128	4.00	27.2899	-27.2899	27.2899

n	x	$F_n(x)$	$F_n(x) - F_{n-1}(x)$	Calculadora
1	6.00	6.0000	36.0000	201.7131
2	6.00	42.0000	120.3429	201.7131
4	6.00	162.3429	39.3173	201.7131
8	6.00	201.6601	0.0530	201.7131
16	6.00	201.7131	0.0000	201.7131
32	6.00	201.7131	0.0000	201.7131
64	6.00	201.7131	0.0000	201.7131
128	6.00	201.7131	-201.7131	201.7131

## Programa 27

$$\cosh(x) = 1 + \frac{1}{2!}x^2 + \frac{1}{4!}x^4 + \frac{1}{6!}x^6 + \dots + \frac{1}{(2n)!}x^{2n} \quad (27)$$

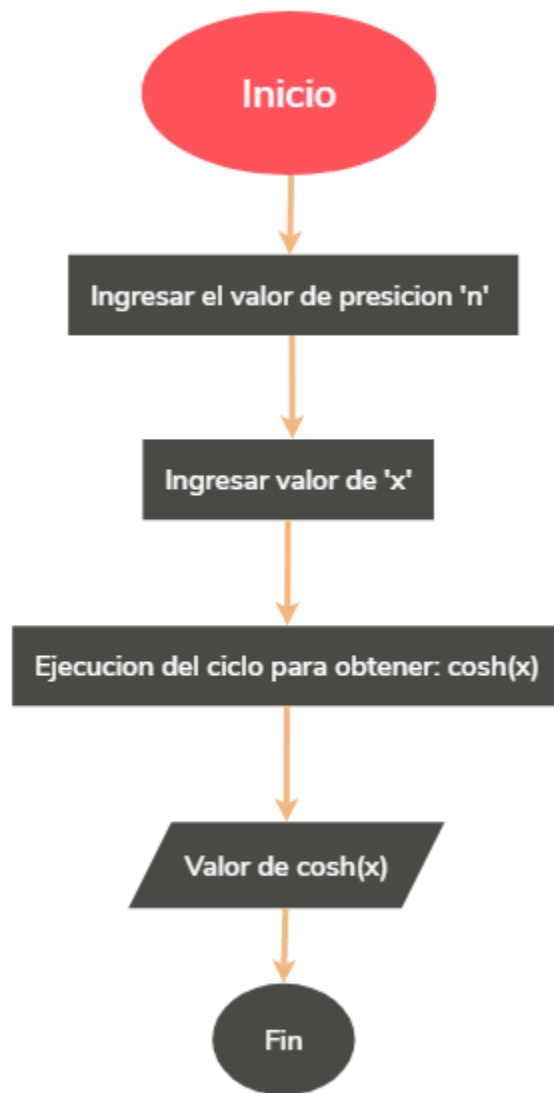
Tabla y Diagrama Programa 27 - cosh(x).

```

1 //Programa 27.cosh(x)
2
3 int main()
4 {
5     int i=1,n;
6     float cos=1,aux2=1,aux=1,x;
7     //Ecuacion 27 cosh (x)
8     printf("Funciones Hiperbolicas: Calculo de cosh (x)");
9     //ADVERTENCIA! Esta funcion tiende a infinito, entre mas grande sea el valor de x mayor sera el resultado
10    //se recomienda utilizar 'x' con valores entre -15 y 15 con iteraciones menores a 50, dependiendo del valor de x
11    printf("\n\nIngrese el valor de x: ");
12    scanf("%f",&x);
13    printf("\n\nIndique la precision o numero de iteraciones: ");
14    scanf("%d",&n);
15
16    while(i<=n){
17        aux*=x;
18        aux2*=i;
19        if(i%2==0){
20            cos+=aux/aux2;
21            i++;
22        }
23    }
24    printf("\n\ncosh (%f) = %f\n\n",x,cos);
25    return 0;
26 }

```

$n$	$x$	$f_n(x)$
1	3.56	1.000000
2	3.56	7.336800
4	3.56	14.029305
8	3.56	18.022564
16	3.56	17.595818
32	3.56	17.595819
64	3.56	17.595819
128	3.56	-1.#IND00
1	0.7225	1.000000
2	0.7225	1.261003
4	0.7225	1.272357
8	0.7225	1.272556
16	0.7225	1.272556
32	0.7225	1.272556
64	0.7225	1.272556
128	0.7225	1.272556
1	-8.3315	1.000000
2	-8.3315	35.706947
4	-8.3315	236.468964
8	-8.3315	1276.782715
16	-8.3315	2069.227051
32	-8.3315	2076.321045
64	-8.3315	-1.#IND00
128	-8.3315	-1.#IND00



## Programa 28

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \frac{17}{315}x^7 + \dots + \frac{B_{2n}4^n(4^n-1)}{(2n)!}x^{2n-1} \quad |x| < \frac{\pi}{2} \quad (28)$$

### Ciclos

```
#include <stdio.h>

int main()
{
    int n,i;
    float rsen,rcos,fact,x,tanh;
    do{
        printf("Obtener el resultado de senh(x) mediante iteraciones\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0 || x<0);
    for(i=0,fact=x,rsen=0.0;i<n;i++)
    {
        rsen+=fact;
        fact*=((x/(2*i+2))*(x/(2*i+3)));
    }
    printf("\nEl resultado de senh(%.3f) es: %.5f\n",x,rsen);

    do{
        printf("Obtener el resultado de cosh(x) mediante iteraciones\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0 || x<0);
    for(i=0,fact=1,rcos=0.0;i<n;i++)
    {
        rcos+=fact;
        fact*=((x/(4*i+3))*(x/(4*i+4)));
    }
    printf("\nEl resultado de cosh(%.3f) es: %.5f\n",x,rcos);
    return 0;

    do{
        printf("Obtener el valor de tanh(x)\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);

    }while(n<0 || x<0);
    for(i=0,fact=0 ,tanh=0.0;i<n;i++)
    {
        tanh=rsen/rcos;
    }
    printf("\nEl resultado de tanh(%.3f) es: %.5f\n",x,tanh);
    return 0;
}
```

### Recursiva

```
#include <stdio.h>

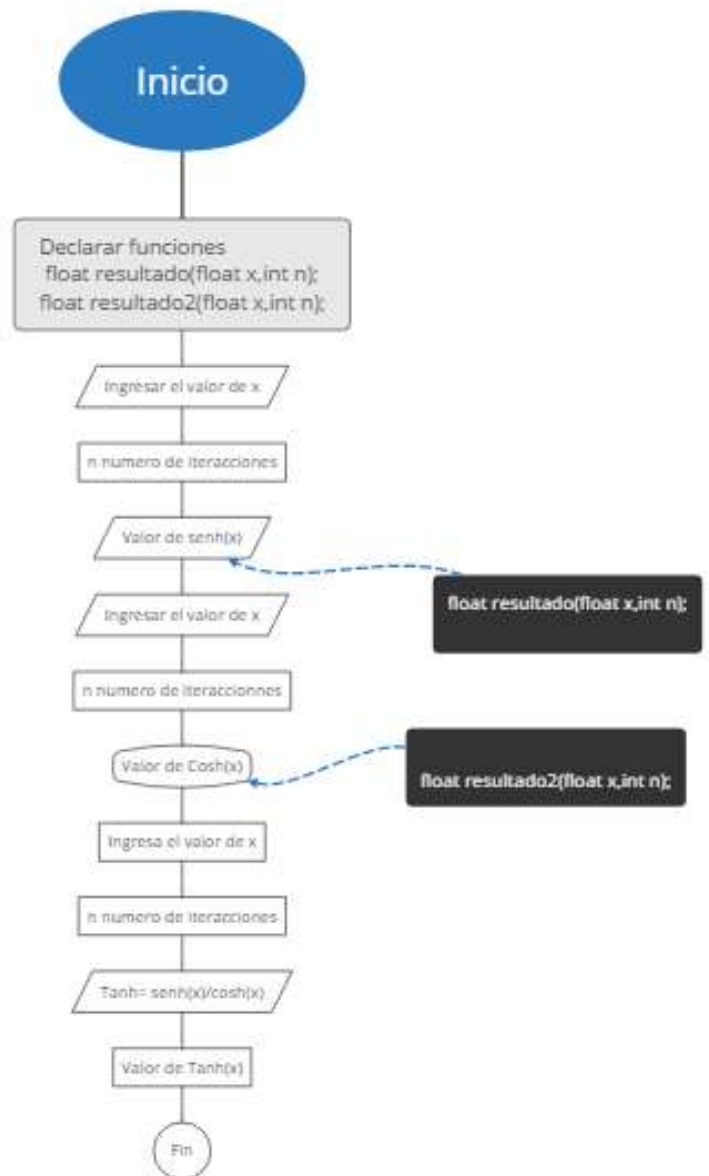
float resultado(float x,int n)
{
    float rsen,rcos,fact,x,tanh;
    int i;
    do{
        printf("Obtener el resultado de senh(x) mediante iteraciones\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);
    }while(n<0 || x<0);
    rsen=resultado2(x,n);
    rcos=resultado2(x,n);
    printf("\nEl resultado de cosh(%.3f) es: %.5f\n",x,rcos);

    do{
        printf("Obtener el valor de tanh(x)\n");
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteraciones: ");
        scanf("%d",&n);

    }while(n<0 || x<0);
    for(i=0, tanh=0.0; i<n;i++)
    {
        tanh=rsen/rcos;
    }
    printf("\nEl resultado de tanh(%.3f) es: %.5f\n",x,tanh);
    return 0;
}

float resultado2(float x,int n)
{
    int i;
    float rsen,fact;
    for(i=0,fact=x,rsen=0.0;i<n;i++)
    {
        rsen+=fact;
        fact*=((x/(2*i+2))*(x/(2*i+3)));
    }
    return rsen;
}

float resultado2(float x,int n)
{
    int i;
    float rcos,fact;
    for(i=0,fact=1,rcos=0.0;i<n;i++)
    {
        rcos+=fact;
        fact*=((x/(4*i+3))*(x/(4*i+4)));
    }
    return rcos;
}
```





x	x1	Diferencia, %
2	0.96403	51.79862
0.96403	0.74607	22.60927
0.74607	0.63280	15.18235
0.63280	0.55998	11.50794
0.55998	0.50796	9.28902
0.50796	0.46835	7.79701
0.46835	0.43687	6.72259
0.43687	0.41105	5.91095
0.41105	0.38936	5.27567

## Programa 29

$$\sinh^{-1}(x) = x - \frac{1}{6}x^3 + \frac{3}{40}x^5 - \frac{5}{112}x^7 + \dots + \frac{(-1)^n(2n)!}{4^n(n!)^2(2n+1)}x^{2n+1} \quad |x| < 1 \quad (29)$$

### Ciclos

```
#include <stdio.h>
#include <math.h>

int main()
{
    int i,j,den,denk,n,sig;
    float x,pot,num,numk,pf,k=1.0,rsen=0.0,rsenf=0.0;
    do{
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteracciones: ");
        scanf("%d",&n);
    }while(x>1||n<0);
    for(i=0;i<n;i++)
    {
        sig=(1-((2)*(i%2)));
        for(j=0;j<i+1;j++)
        {
            numk=((2*(j)+1));
            denk=((2*(j)+2));
            pot=((2*(j)+1));
            num=pow(x,pot);
            den=((2*(j)+1));
        }
        k*=(numk/denk);
        pf=(num/den);
        rsen=((sig)*(k*pf));
        rsenf+=rsen;
    }
    printf("El resultado de sinh^-1(%.3f) es: %.6f ",x,rsenf);
    return 0;
}
```

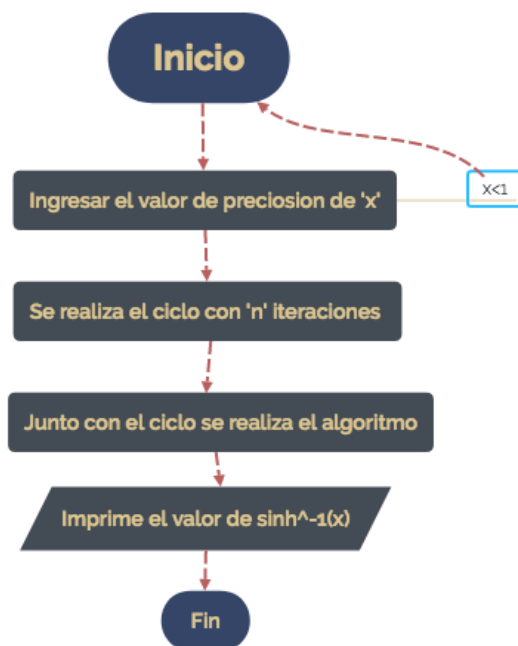
### Recursiva

```
#include <math.h>

float resultado(float x,int n);

int main()
{
    int n;
    float x,rsenf;
    do{
        printf("\nIngrese el valor de x: ");
        scanf("%f",&x);
        printf("Ingrese el numero de iteracciones: ");
        scanf("%d",&n);
    }while(x>1||n<0);
    rsenf=resultado(x,n);
    printf("El resultado de sinh^-1(%.3f) es: %.6f ",x,rsenf);
    return 0;
}

float resultado(float x,int n)
{
    int i,j,den,denk,sig;
    float pot,num,numk,pf,k=1.0,rsen=0.0,rsenf=0.0;
    for(i=0;i<n;i++)
    {
        sig=(1-((2)*(i%2)));
        for(j=0;j<i+1;j++)
        {
            numk=((2*(j)+1));
            denk=((2*(j)+2));
            pot=((2*(j)+1));
            num=pow(x,pot);
            den=((2*(j)+1));
        }
        k*=(numk/denk);
        pf=(num/den);
        rsen+=((sig)*(k*pf));
    }
    return rsenf;
}
```



# Programa 29

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	1.00	0.5000	-0.1250	0.4083
2	1.00	0.3750	0.0333	0.4083
4	1.00	0.4083	0.0000	0.4083
8	1.00	0.4083	0.0000	0.4083
16	1.00	0.4083	0.0000	0.4083
32	1.00	0.4083	0.0000	0.4083
64	1.00	0.4083	0.0000	0.4083
128	1.00	0.4083	-0.4083	0.4083

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	-1.00	-0.5000	0.1250	-0.4099
2	-1.00	-0.3750	-0.0349	-0.4099
4	-1.00	-0.4099	0.0000	-0.4099
8	-1.00	-0.4099	0.0000	-0.4099
16	-1.00	-0.4099	0.0000	-0.4099
32	-1.00	-0.4099	0.0000	-0.4099
64	-1.00	-0.4099	0.0000	-0.4099
128	-1.00	-0.4099	0.4099	-0.4099

n	x	Fn (x)	Fn (x) - Fn-1 (x)	Calculadora
1	-2.00	-1.0000	1.0000	inf
2	-2.00	0.0000	3.0000	inf
4	-2.00	3.0000	165330.5156	inf
8	-2.00	165333.5156	7435709.4844	inf
16	-2.00	#####	#VALUE!	inf
32	-2.00	inf	#VALUE!	inf
64	-2.00	inf	#VALUE!	inf
128	-2.00	inf	#VALUE!	inf

### Programa 30

$$\tanh^{-1}(x) = x + \frac{1}{3}x^3 + \frac{1}{5}x^5 + \frac{1}{7}x^7 + \dots + \frac{1}{2n+1}x^{2n+1} \quad |x| < 1 \quad (30)$$

Tabla y Diagrama Programa 30 ~  $\tanh^{-1}(x)$ .

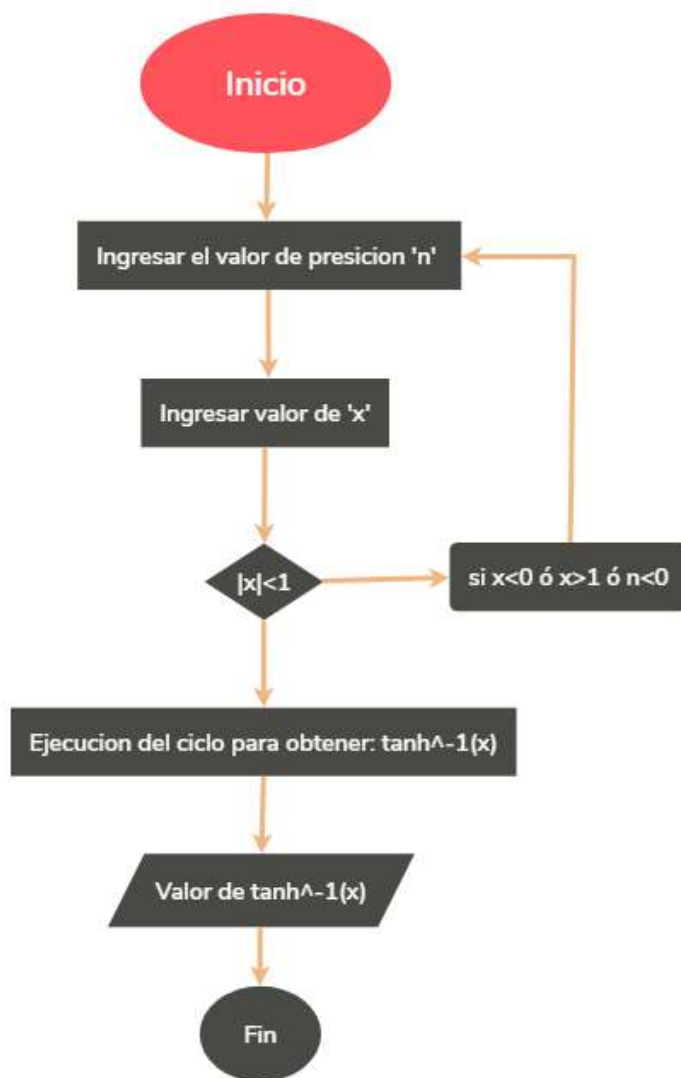
```

1  #include <stdio.h>
2
3  int main()
4  {
5      int i,n;
6      float x,aux=1,th=0;
7      //Ecuacion 30 tanh^-1 (x)
8      printf("Funciones Trigonometricas: Calculo de tanh^-1 (x)");
9      do{
10         printf("\n\nIngrese el valor de x (valores entre 0 y 1): ");
11         scanf("%f",&x);
12         printf("\n\nIndique la precision o numero de iteraciones: ");
13         scanf("%d",&n);
14     }while (x>1 || x<0 || n<0);
15
16     for(i=1;i<n+1;i++)
17     {
18         aux*=x; //Potencia de x
19         if(i%2!=0)
20         {
21             th+=aux/i;
22         }
23     }
24     printf("\n\ntanh^-1 (%f) = %f \n\n",x,th);
25     return 0;
26 }

```

$n$	$x$	$f_n(x)$
1	0.0907	0.090700
2	0.0907	0.090700
4	0.0907	0.090949
8	0.0907	0.090950
16	0.0907	0.090950
32	0.0907	0.090950
64	0.0907	0.090950
128	0.0907	0.090950
1	0.5011	0.501100
2	0.5011	0.501100
4	0.5011	0.543042
8	0.5011	0.550495
16	0.5011	0.550773
32	0.5011	0.550774
64	0.5011	0.550774
128	0.5011	0.550774

1	0.91979	0.919790
2	0.91979	0.919790
4	0.91979	1.179175
8	0.91979	1.390406
16	0.91979	1.523964
32	0.91979	1.577781
64	0.91979	1.587281
128	0.91979	1.587661



## Programa 31

$$\frac{\ln(1+x)}{1+x} = x - \left(1 + \frac{1}{2}\right) x^2 + \left(1 + \frac{1}{2} + \frac{1}{3}\right) x^3 - \dots \quad |x| < 1 \quad (31)$$

### Programa 31:

```
#include<stdio.h>

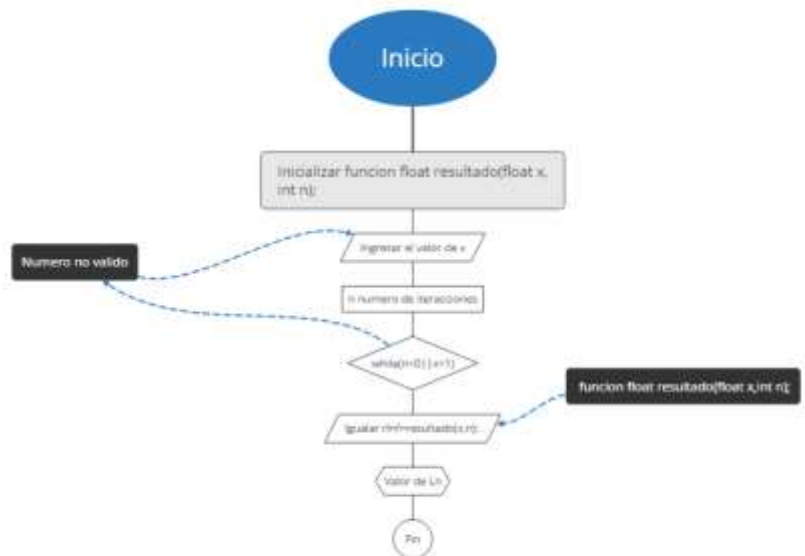
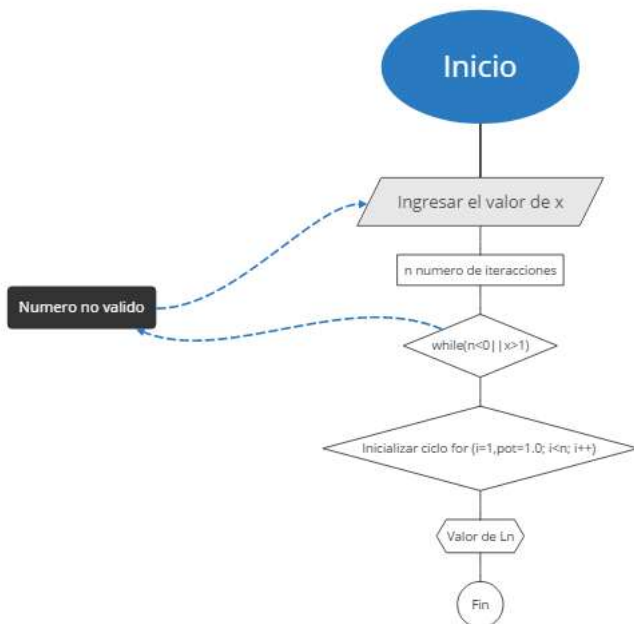
int main()
{
    int n,i,sig;
    float x,k,pot,rln=0.0;
    do
    {
        printf("Obtener el resultado del logaritmo natural de (1+x)/(1-x) \nSiempre y cuando x sea menor a 1 \n");
        printf("\nIngrese el valor de x: ");
        scanf("%f", &x); fflush(stdin);
        printf("Numero de iteraciones: ");
        scanf("%d", &n); fflush(stdin);
    }while(n<0||x>1);
    for(i=1,pot=1.0; i<n; i++)
    {
        sig=(1-((2)*(i%2)));
        pot*=x;
        k+=(sig*1.0/i+1.0);
        rln+=k*pot;
    }
    printf("\nEl resultado de (ln(%f))/%.2f es: %.4f\n", 1+x,1-x,rln);
    return 0;
}
```

```
#include<stdio.h>

float resultado(float x,int n);

int main()
{
    int n;
    float x,rlnf;
    do
    {
        printf("Obtener el resultado del logaritmo natural de (1+x)/(1-x) \nSiempre y cuando x sea menor a 1 \n");
        printf("\nIngrese el valor de x: ");
        scanf("%f", &x); fflush(stdin);
        printf("Numero de iteraciones: ");
        scanf("%d", &n); fflush(stdin);
    }while(n<0||x>1);
    rlnf=resultado(x,n);
    printf("\nEl resultado de (ln(%f))/%.2f es: %.4f\n", 1+x,1-x,rlnf);
    return 0;
}

float resultado(float x,int n)
{
    int i,sig;
    float k,pot,rln=0.0;
    for(i=1,pot=1.0; i<n; i++)
    {
        sig=(1-((2)*(i%2)));
        pot*=x;
        k+=(sig*1.0/i+1.0);
        rln+=k*pot;
    }
    return rln;
}
```



**Tabla de programa 31:**

x	x1	Diferencia, %
2	0.36620	81.68980
0.36620	0.22840	37.63138
0.22840	0.16746	26.67928
0.16746	0.13262	20.80419
0.13262	0.10995	17.09296
0.10995	0.09398	14.52368
0.09398	0.08211	12.63503

## **Conclusión.**

En esta práctica cumplimos los objetivos anteriormente establecidos, trabajamos en equipo y terminamos los códigos de la manera más eficiente, estos códigos pueden ser utilizados para futuros proyectos y es muy importante guardarlos ya que son muy exactos y pueden de ser de gran utilidad en un futuro, en mi opinión personal fue un trabajo bastante duro, pero todos estamos satisfechos, muchas gracias.

## **Bibliografía.**

- 1.- <https://sume.ugto.mx>
- 2.- [https://es.wikipedia.org/wiki/C\\_\(lenguaje\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n))
- 3.- [https://es.wikipedia.org/wiki/Bucle\\_for](https://es.wikipedia.org/wiki/Bucle_for)