



**Universidad Tecnológica
de Pereira**

Facultad de Ingenierías

Programa de Ingeniería de Sistemas y Computación

Título de la tarea: Programación 2 - Código IS284

Nombre del Estudiante: Esteban Vásquez Castañeda.Ing©

Profesor: Dr. Ricardo Moreno Laverde

Fecha de Entrega: 14 de Marzo del 2024

1.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de fibonacci
menores a 10000
#include <stdio.h>
void Fibonacci (int , int);
int main(){
    printf ("Este programa presenta la serie de Fibonacci como la serie que
comienza con los dígitos 1 y 0 y va\n");
    printf("sumando progresivamente los dos últimos elementos de la serie,
así: 0 1 1 2 3 5 8 13 21 34.....\n");
    printf("Para este programa, se presentará la serie de Fibonacci hasta
llegar sin sobrepasar el número 10,000.\n");
    Fibonacci(1, 1);
    return 0;
}
//Calcula serie
void Fibonacci (int n, int counter){
    if (counter<10000){
        printf ("%i, ", n);
        Fibonacci(counter, n+counter);
    }
    else
        printf ("%i.", n);
}
```

Diseño de pantalla:

```
Este programa presenta la serie de Fibonacci como la serie que comienza con los dígitos 1 y 0 y va
sumando progresivamente los dos últimos elementos de la serie, así: 0 1 1 2 3 5 8 13 21 34.....
Para este programa, se presentará la serie de Fibonacci hasta llegar sin sobrepasar el número 10,000.
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765.
```

2.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Casta  da.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programaci  n II
//Descripcion del programa: Este programa Calcula la serie de fibonacci
menores a 100 y hace la suma de estos.
#include <stdio.h>
void Fibonacci (int, int, int);
int main(){
    printf ("Este programa presenta la suma de los elementos de la serie de
Fibonacci entre 0 y 100. Los n  meros a sumar son:\n");
    Fibonacci (1, 1, 0);

    return 0;
}
//Calcula serie
void Fibonacci (int n, int counter, int add){
    if (counter<100){
        printf ("%i, ", n);
        Fibonacci(counter, n+counter, n+add);
    }
    else
        printf ("%i. y la su suma es de %i", n, add+n);
}
```

Dise  o de pantalla:

```
Este programa presenta la suma de los elementos de la serie de Fibonacci entre 0 y 100. Los n  meros a sumar son:
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. y la su suma es de 232
```

3.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Casta  da.Ing c
//Lenguaje C
//Version del lenguaje: C18
```

```

//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Lucas con de
la cantidad de terminos dada por el usuario
#include <stdio.h>
void SerieLucas (int, int, int, int);
int main(){
    int num=0;//Cantidad de terminos
    printf ("Ingrese el numero de terminos que desea en la serie de lucas:
");
    scanf("%i", &num);
    SerieLucas (2, 1, 0, num);

    return 0;
}
//Calcula la serie
void SerieLucas (int nMenos1, int n, int counter, int end){
    if (counter<end) {
        printf ("%i, ", nMenos1);
        SerieLucas(n, nMenos1+n, counter+1, end);
    }
    else
        printf ("%i.", nMenos1);
}

```

Diseño de pantalla:

```

Ingrese el numero de terminos que desea en la serie de lucas: 9
2, 1, 3, 4, 7, 11, 18, 29, 47, 76.

```

```

Ingrese el numero de terminos que desea en la serie de lucas: 16
2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207.

```

```

Ingrese el numero de terminos que desea en la serie de lucas: 5
2, 1, 3, 4, 7, 11.

```

4.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castaño. Ing c

```

```

//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Pell con de
la cantidad de terminos dada por el usuario
#include <stdio.h>
void SeriePell (int, int, int, int);
int main(){
    int num=0;//Cantidad de terminos
    printf ("Ingrese el numero de terminos que desea en la serie de Pell:
");
    scanf("%i", &num);
    SeriePell (0, 1, 0, num);

    return 0;
}
//Calcula la serie
void SeriePell (int nMenos1, int n, int counter, int end){
    if (counter<end) {
        printf ("%i, ", nMenos1);
        SeriePell(n, nMenos1+2*n, counter+1, end);
    }
    else
        printf ("%i.", nMenos1);
}

```

Diseño de pantalla:

```

Ingrese el numero de terminos que desea en la serie de Pell: 5
0, 1, 2, 5, 12, 29.

```

```

Ingrese el numero de terminos que desea en la serie de Pell: 9
0, 1, 2, 5, 12, 29, 70, 169, 408, 985.

```

```

Ingrese el numero de terminos que desea en la serie de Pell: 20
0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860, 33461, 80782, 195025, 470832, 1136689, 2744210, 6625109, 15994428.

```

5.

```

//Fecha: 11/03/2024

```

```

//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Perrin con de
la cantidad de terminos dada por el usuario
#include <stdio.h>
void SeriePerrin (int, int, int, int, int);
int main(){
    int num=0;//Cantidad de terminos
    printf ("Ingrese el numero de terminos que desea en la serie de Perrin:
");
    scanf("%i", &num);
    SeriePerrin (3, 0, 2, 0, num);

    return 0;
}
//Calcula serie
void SeriePerrin (int nMenos2, int nMenos1, int n, int counter, int end){
    if(counter<end){
        printf("%i, ", nMenos2);
        SeriePerrin (nMenos1, n, nMenos2+nMenos1, counter+1, end);
    }
    else
        printf("%i, ", nMenos2);
}

```

Diseño de pantalla:

```

Ingrese el numero de terminos que desea en la serie de Perrin: 9
3, 0, 2, 3, 2, 5, 5, 7, 10, 12,

```

```

Ingrese el numero de terminos que desea en la serie de Perrin: 16
3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39, 51, 68, 90,

```

```

Ingrese el numero de terminos que desea en la serie de Perrin: 20
3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39, 51, 68, 90, 119, 158, 209, 277,

```

6.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Padovan con
de la cantidad de terminos dada por el usuario
#include <stdio.h>
void SeriePadovan (int, int, int, int, int);
int main(){
    int num=0;//Cantidad terminos
    printf ("Ingrese el numero de terminos que desea en la serie de Padovan:
");
    scanf("%i", &num);
    SeriePadovan (1, 0, 0, 0, num);

    return 0;
}
//Calcular Serie
void SeriePadovan (int nMenos2, int nMenos1, int n, int counter, int end){
    if(counter<end){
        printf("%i, ", nMenos2);
        SeriePadovan(nMenos1, n, nMenos2+nMenos1, counter+1, end);
    }
    else
        printf("%i.", nMenos2);
}
```

Diseños de pantalla:

```
Ingrese el numero de terminos que desea en la serie de Padovan: 7
1, 0, 0, 1, 0, 1, 1, 1.
```

Ingrese el numero de terminos que desea en la serie de Padovan: 10
1, 0, 0, 1, 0, 1, 1, 1, 2, 2, 3.

Ingrese el numero de terminos que desea en la serie de Padovan: 16
1, 0, 0, 1, 0, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16.

7.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Narayana con
de la cantidad de terminos dada por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>
int SerieNarayana (int, int, int, int, int);
int main(){
    int num=0;//Cantidad Parametros
    printf ("Ingrese el numero de terminos que desea en la serie de
Narayana: ");
    scanf("%i", &num);
    SerieNarayana (1, 1, 1, 0, num);

    return 0;
}
//Calcular Serie
int SerieNarayana (int start, int data1, int data2, int counter, int end){
    if(counter<end){
        printf("%i, ", start);
        SerieNarayana (data1, data2, start+data2, counter+1, end);
    }
    else
        printf("%i.", start);

    return 0;
}
```


Diseños de pantalla:

```
Ingrese el numero de terminos que desea en la serie de Narayana: 7
1, 1, 1, 2, 3, 4, 6, 9.
```

```
Ingrese el numero de terminos que desea en la serie de Narayana: 14
1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129.
```

```
Ingrese el numero de terminos que desea en la serie de Narayana: 19
1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129, 189, 277, 406, 595, 872.
```

8.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Catalan con
de la cantidad de terminos dada por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>

//Calcula el factorial de un numero
long long int DoFactorial (int number){
    if (number==0)
        return 1;
    else
        return number*DoFactorial(number-1);
}

//Calcula el Termino catalan correspondiente
int CalculateTerm (long int term){
    if (term==0)
        return 1;
    else
        return DoFactorial(2*term)/(DoFactorial(term)*DoFactorial(term+1));
}
```

```

}

//Imprime la Serie
void PrintCatalan( int term, int quantity){
    if ( term<quantity){
        printf("%i, ", CalculateTerm(term));
        PrintCatalan(term+1, quantity);
    }
    else
        printf("%i.", CalculateTerm(term));
}

//Funcion Principal
int main(){
    int quantityTerms = 0;//Cantidad de terminos

    printf("Ingrese el numero de terminos que desea conocer de la serie de Catalan: ");
    scanf("%i", &quantityTerms);
    PrintCatalan(0, quantityTerms-1);
    return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Catalan: 3
1, 1, 2.

```

```

Ingrese el numero de terminos que desea conocer de la serie de Catalan: 7
1, 1, 2, 5, 14, 42, 132.

```

```

Ingrese el numero de terminos que desea conocer de la serie de Catalan: 11
1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796.

```

9.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde

```

```

//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Bell con de
la cantidad de terminos dada por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>
int CalculateTerm(int);

//Calcula el factorial
long long int DoFactorial (int number){
    if (number==0)
        return 1;
    else
        return number*DoFactorial(number-1);
}

//Calcula la sumatoria de un termino
int SummationStirling (int n, int i) {
    return (DoFactorial(n-1)/(DoFactorial(i)*DoFactorial((n-1)-i)));
}

//Calcula la sumatoria
int SummationBell( int n, int i, int accumulator ){
    if (i > n-1) { // i<n-1
        return accumulator;
    } else {
        return SummationBell(n, i+1, accumulator + SummationStirling(n,
i)*CalculateTerm(i));
    }
}

//Funcion principal de la sumatoria de bell
int CalculateTerm(int term){
    if(term==0 || term==1)
        return 1;
    else
        return SummationBell(term, 0, 0);
}

//Imprime los valores termino a termino
void PrintBell( int term, int quantity){
    if ( term<quantity){
        printf("%i, ", CalculateTerm(term));
        PrintBell(term+1, quantity);
    }
}

```

```

    }
    else
        printf("%i.", CalculateTerm(term));
}

//Funcion Principal
int main(){
    int quantityTerms = 0; //Cantidad terminos

    printf("Ingrese el numero de terminos que desea conocer de la serie de Bell: ");
    scanf("%i", &quantityTerms);
    PrintBell(0, quantityTerms-1);
    return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Bell: 5
1, 1, 2, 5, 15.

```

```

Ingrese el numero de terminos que desea conocer de la serie de Bell: 9
1, 1, 2, 5, 15, 52, 203, 877, 4140.

```

```

Ingrese el numero de terminos que desea conocer de la serie de Bell: 16
1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, 27644437, 190899322, 1382958545.

```

10.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la Serie de Motzkin con
de la cantidad de terminos dada por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>

```

```

//Hace las operaciones de la serie
int motzkin(int n) {
    // Casos base
    if (n == 0 || n == 1)
        return 1;
    if (n == 2)
        return 2;

    // Llamadas recursivas y sumas parciales
    return ((2 * n + 1) * motzkin(n - 1) + (3 * n - 3) * motzkin(n - 2)) /
(n + 2);
}

//Imprime los terminos
void print_motzkin_series_helper(int n, int i) {
    if (i > n) return;
    printf("%d ", motzkin(i)); // Imprime el término actual
    print_motzkin_series_helper(n, i + 1); // Llama recursivamente para el
siguiente término
}

//Funcion Principal
int main() {
    int n=0;//Cantidad de terminos
    printf("Ingrese el valor de n para calcular la serie de Motzkin: ");
    scanf("%d", &n);

    print_motzkin_series_helper(n, 0);

    return 0;
}

```

Diseños de pantalla:

```

Ingrese el valor de n para calcular la serie de Motzkin: 6
1 1 2 4 9 21 51

```

```

Ingrese el valor de n para calcular la serie de Motzkin: 12
1 1 2 4 9 21 51 127 323 835 2188 5798 15511

```

```

Ingrese el valor de n para calcular la serie de Motzkin: 19
1 1 2 4 9 21 51 127 323 835 2188 5798 15511 41835 113634 310572 853467 2356779 6536382 18199284

```

11.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Calcula la serie de Triangular
con de la cantidad de terminos dada por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>

//Calcula terminos
int TriangularSeries (int number){
    if (number==0)
        return 0;
    else
        return number+TriangularSeries(number-1);
}

//Imprime termino a termino
void imprimir_serie(int n, int termino_actual) {
    if (n <= 0) {
        return;
    } else {
        printf("%d", TriangularSeries(termino_actual));
        if (n > 1) {
            printf(", ");
        }
        imprimir_serie(n - 1, termino_actual + 1);
    }
}

//Funcion principal
int main() {
    int n=0;//Cantidad de terminos
```

```

    printf("Ingrese el número de términos deseados en la serie triangular:
");
    scanf("%d", &n);

    if (n < 1) {
        printf("El número de términos debe ser al menos 1.\n");
        return 1;
    }

    printf("Los primeros %d términos de la serie triangular son:\n", n);
    imprimir_serie(n, 1);
    printf("\n");

    return 0;
}

```

Diseños de pantalla:

```

Ingrese el número de términos deseados en la serie triangular: 8
Los primeros 8 términos de la serie triangular son:
1, 3, 6, 10, 15, 21, 28, 36

```

```

Ingrese el número de términos deseados en la serie triangular: 16
Los primeros 16 términos de la serie triangular son:
1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136

```

```

Ingrese el número de términos deseados en la serie triangular: 25
Los primeros 25 términos de la serie triangular son:
1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153, 171, 190, 210, 231, 253, 276, 300, 325

```

12.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Programa de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II

```

```

//Descripcion del programa: Este programa Invierte un numero dado por el
usuario
//Salvedad: Solo toma valores positivos

#include <stdio.h>
void turNum (int);
int main(){
    int num=0; //Recibe el numero dado por el usuario
    printf ("Escriba el numero que quiere reordenar: ");
    scanf("%i", &num);
    turNum (num);

}

//Funcion que invierte el numero
void turNum (int num) {
    if(num>0) {
        printf("%i", num%10);
        turNum(num/10);
    }
}

```

Diseños de pantalla:

```

Escriba el numero que quiere reordenar: 985
589

```

```

Escriba el numero que quiere reordenar: 577
775

```

```

Escriba el numero que quiere reordenar: 658
856

```

13.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira

```



```
//Programa de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripción del programa: Este programa recibe 75 numeros dados por el
usuario y dice cuantos numeros positivos hay, cuantos negativos
//Cual es el numero mayor, cual es el numero menor y cual es el promedio de
los numeros positivos

#include <stdio.h>

// Función para calcular la cantidad de números mayores a 150.
int CalculateMajorTerm(int num) {
    if (num > 150)
        return 1;
    else
        return 0;
}

// Función para calcular la cantidad de números negativos.
int CalculateNumNegative(int num) {
    if (num < 0)
        return 1;
    else
        return 0;
}

// Función para encontrar el número mayor.
int NumMajor(int num1, int num2) {
    if (num1 > num2)
        return num1;
    else
        return num2;
}

// Función para encontrar el número menor.
int NumMin(int num1, int num2) {
    if (num1 < num2)
        return num1;
    else
        return num2;
}

// Función para determinar si un número es positivo.
int NumPositive(int num) {
    if (num > 0)
        return 1;
```

```

        else
            return 0;
    }

// Función para sumar los términos positivos.
float SummationTerms(int num) {
    if (num > 0)
        return num;
    else
        return 0;
}

// Función para solicitar los términos y realizar los cálculos.
void AskForTerms(int term, int amountMayor150, int amountNegative, int
numMayor, int numMinium, float summationPositive, int numPositive) {
    int num = 0; // Valor asignado por el usuario
    if (term <= 75) {
        printf("Escriba el término %i: ", term);
        scanf("%i", &num);
        if (num == 0) { // Confirma que el término sea diferente de cero
(0).
            printf("El número debe ser diferente de cero (0). Intente de
nuevo.\n");
            AskForTerms(term, amountMayor150, amountNegative, numMayor,
numMinium, summationPositive, numPositive);
        } else
            AskForTerms(term + 1, amountMayor150 + CalculateMajorTerm(num),
amountNegative + CalculateNumNegative(num), NumMajor(numMayor, num),
NumMin(numMinium, num), summationPositive + SummationTerms(num), numPositive
+ NumPositive(num));
        } else {
            printf("Hay %i números mayores a 150.\n", amountMayor150);
            printf("Hay %i números negativos.\n", amountNegative);
            printf("El número mayor de los términos es %i y el más pequeño es
%i.\n", numMayor, numMinium);
            if (numPositive != 0) // Verifica si hay números positivos para
evitar división por cero.
                printf("El promedio de los números positivos es %.2f.\n",
summationPositive / numPositive);
            else
                printf("No hay números positivos para calcular el promedio.\n");
        }
    }
}

// Función principal

```

```
int main() {
    printf("Este programa recibe 75 términos y muestra algunas conclusiones sobre ellos:\n");
    AskForTerms(1, 0, 0, 0, 0, 0, 0);
    return 0;
}
```

Diseños de pantalla(Como es un ejemplo y son demasiados números, el diseño de pantalla solo hará el ejemplo con 5 términos por diseño de pantalla):

```
Este programa recibe 75 términos y muestra algunas conclusiones sobre ellos:
Escriba el término 1: 654
Escriba el término 2: 152
Escriba el término 3: -546
Escriba el término 4: 8
Escriba el término 5: -8
Hay 2 números mayores a 150.
Hay 2 números negativos.
El número mayor de los términos es 654 y el más pequeño es -546.
El promedio de los números positivos es 271.33.
```

```
Este programa recibe 75 términos y muestra algunas conclusiones sobre ellos:
Escriba el término 1: 845
Escriba el término 2: 564
Escriba el término 3: -85
Escriba el término 4: 5
Escriba el término 5: -98
Hay 2 números mayores a 150.
Hay 2 números negativos.
El número mayor de los términos es 845 y el más pequeño es -98.
El promedio de los números positivos es 471.33.
```

```
Este programa recibe 75 términos y muestra algunas conclusiones sobre ellos:
Escriba el término 1: 65
Escriba el término 2: 564
Escriba el término 3: -89
Escriba el término 4: 655
Escriba el término 5: 0
El número debe ser diferente de cero (0). Intente de nuevo.
Escriba el término 5: -1
Hay 2 números mayores a 150.
Hay 2 números negativos.
El número mayor de los términos es 655 y el más pequeño es -89.
El promedio de los números positivos es 428.00.
```

14.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
```

```
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Muestra las multiplicaciones de
la tabla del uno hasta la tabla del 10
//Salvedad: Solo toma valores positivos
#include <stdio.h>
void multiplier(int, int);

int main(){
    multiplier(1, 1);//Inicia en la multiplicacion 1x1
}

//Funcion para mostrar las tablas de multiplicar
void multiplier(int num1, int num2){
    if(num1<=10){
        if(num2<=10){
            printf("%i x %i = %i\n", num1, num2, num1*num2);
            multiplier(num1, num2+1);
        }
        else
            multiplier(num1+1, 1);
    }
}
```

Diseños de pantalla:

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

15.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Imprime la serie de factoriales
desde el termino cero hasta el termino dado por el usuario
//Salvedad: Solo toma valores positivos
#include <stdio.h>
int DoFactorial(int);
void PrintFactorial(int, int);

int main(){
    int num=0;// Cantidad terminos
    printf("Escriba el numero para hayarle una factorial: ");
    scanf("%i", &num);
    PrintFactorial(0, num);

}

//Calcula el factorial
int DoFactorial (int number){
    if (number==0)
        return 1;
    else
        return number*DoFactorial(number-1);
}

//Imprime termino a termino
void PrintFactorial( int term, int quantity){//Term es para el termino y el
que varia, quantity es el termino final
    if ( term<quantity){
        printf("%i, ", DoFactorial(term));
        PrintFactorial(term+1, quantity);
    }
    else
```

```
    printf("%i.", DoFactorial(term));
}
```

Diseños de pantalla:

```
Escriba el numero para hayarle una factorial: 7
1, 1, 2, 6, 24, 120, 720, 5040.
```

```
Escriba el numero para hayarle una factorial: 9
1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880.
```

```
Escriba el numero para hayarle una factorial: 16
1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, 39916800, 479001600, 1932053504, 1278945280, 2004310016, 2004189184.
```

16.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Muestra los terminos del
factorial y los suma
//Salvedad: Solo toma valores positivos
#include <stdio.h>
int DoFactorial(int);
void PrintFactorial(int, int, int);

int main(){
    int num=0;// Cantidad terminos
    printf("Escriba el numero para hayarle una factorial: ");
    scanf("%i", &num);
    if(num>=0){
        PrintFactorial(0, num, 0);
    }
    else{
        printf("El numero no es positivo. intente de nuevo.");
        main();
    }
}
```

```

    }

//Calcula el factorial
int DoFactorial (int number){
    if (number==0)
        return 1;
    else
        return number*DoFactorial(number-1);
}

//Imprime los factoriales
void PrintFactorial ( int term, int quantity, int counter){
    if ( term<quantity){
        printf("%i + ", DoFactorial(term));
        PrintFactorial(term+1, quantity, counter+DoFactorial(term));
    }
    else{
        printf("%i = ", DoFactorial(term));
        printf ("%i", counter+DoFactorial(term));
    }
}
}

```

Diseños de pantalla:

```

Escriba el numero para hayarle una factorial: 5
1 + 1 + 2 + 6 + 24 + 120 = 154

```

```

Escriba el numero para hayarle una factorial: 9
1 + 1 + 2 + 6 + 24 + 120 + 720 + 5040 + 40320 + 362880 = 409114

```

```

Escriba el numero para hayarle una factorial: 12
1 + 1 + 2 + 6 + 24 + 120 + 720 + 5040 + 40320 + 362880 + 3628800 + 39916800 + 479001600 = 522956314

```

17.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira

```



```

//Programa de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Algunas Parejas de numeros
enteros siguiendo un patron
#include <stdio.h>
void NumericalPairs(int, int, int);

int main(){
    NumericalPairs(0, 1, 0);
}

void NumericalPairs(int num1, int num2, int counter){
    if(num1<10){
        if(counter%2==0){
            printf("%i %i\n", num1, num2);
            NumericalPairs(num1+1, num2, counter+1);
        }
        else{
            printf("%i %i\n", num1, num2);
            NumericalPairs(num1+1, num2+1, counter+1);
        }
    }
}

```

Diseños de pantalla:

```

0 1
1 1
2 2
3 2
4 3
5 3
6 4
7 4
8 5
9 5

```

18.

```

//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0

```

```

//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Algunas Ternas de numeros
enteros siguiendo un patron
#include <stdio.h>
void NumericalPairs(int, int, int, int);

int main(){
    NumericalPairs(1, 1, 1, 1);
}

//Calcula las ternas
void NumericalPairs(int num1, int num2, int num3, int counter){
    if(num1<10){
        if(counter%3==0){
            if(num3<3){
                printf("%i %i %i\n", num1, num2, num3);
                NumericalPairs(num1+1, num2+1, num3+1, counter+1);
            }
            else{
                printf("%i %i %i\n", num1, num2, num3);
                NumericalPairs(num1+1, num2+1, 1, counter+1);
            }
        }
        else{
            printf("%i %i %i\n", num1, num2, num3);
            NumericalPairs(num1+1, num2, num3+1, counter+1);
        }
    }
}
}

```

Diseños de pantalla:

```

1 1 1
2 1 2
3 1 3
4 2 1
5 2 2
6 2 3
7 3 1
8 3 2
9 3 3

```

19.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Hace un efecto de movimientos de
dos A ubicadas en dos extremos

#include <stdio.h>
#include <unistd.h>

// Función para imprimir la animación de los movimientos de las letras A.
void printAnimation(int x1, int x2) { //Parametros de posicion
    if (x1 == 40 && x2 == 41) //En el momento que los dos se muevan 4
    espacios, se detiene el programa
        return;

    printf("\033[H\033[J"); //Funcion para ir borrando el recorrido luego de
    pasarlo
    printf("%*sA%*sA\n", x1, "", x2 - x1 - 1, "");

    if (x1 < 40)
        x1++;
    if (x2 > 41)
        x2--;

    usleep(100000); // Ajuste el tiempo de espera a 200,000 microsegundos
    (0.2 segundos)
    printAnimation(x1, x2); //Se vuelve a llamar de forma recursiva
}

int main() {
    printAnimation(0, 80); //Se llama a la funcion con dos A, una en la
    posrpcion uno y otro en la posicion 8
    return 0;
}
```

Diseños de pantalla:

```
  A                                     A
█
```

```
  █                                     A       A
```

```
                                     A A
PS C:\Users\esteb\OneDrive\Documentos\Tareas\Progra 2\Tarea 3\output> █
```

20.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa:  Genera una figura con la letra A

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" ");
        Spaces(amount - 1);
    }
}

// Función para imprimir la letra 'A' con un número dado de repeticiones.
void LetterA(int amount) {
    if(amount > 0) {
        printf("A");
        LetterA(amount - 1);
    } else {
```

```
printf("\n");
}
}

// Función para formar la figura utilizando la letra 'A'.
void FormLetter(int amount) {
    if(amount > 55) {
        Spaces(amount); // Llama a la función Spaces() para imprimir
        espacios iniciales.
        LetterA(80 - amount); // Llama a la función LetterA() para imprimir
        la letra 'A'.
        FormLetter(amount - 1); // Llama recursivamente a FormLetter() con
        un decremento en el número de columnas.
    }
}

int main() {
    FormLetter(79); // Llama a la función FormLetter() con un ancho inicial
    de 79 columnas.
    return 0;
}
```

Diseños de pantalla:

[illegible]

21.

```
//Fecha: 11/03/2024
//Hora: 8:46 Pm
//Version del codigo: 1.0
//Autor: Esteban Vasquez Castañeda.Ing c
//Lenguaje C
//Version del lenguaje: C18
//Version del compilador: 13.2.0
//Nombre del SO: Windows 11
//Version del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnologica De Pereira
//Porgrama de Ingenieria de Sistemas y Computacion
//Asignatura IS284 Programación II
//Descripcion del programa: Este programa Genera una figura con la letra P
#include <stdio.h>
void Spaces(int);
void LettterP(int);
void FormLetter(int, int, int);

int main(){
    FormLetter(7, 0, 0);
    return 0;
}

//Genera una cantidad de espacios
void Spaces(int amount){
    if(amount>0){
        printf(" ");
        Spaces(amount-1);
    }
}

//Genera una de las letras dependiendo de i
void LetterPNLJHFD(int amount, int i){
    char letter [8] = "PNLJHFD";
    if(amount>0){
        printf("%c", letter[i]);
        Spaces(1);
        LetterPNLJHFD(amount-1, i);
    }
    else
        printf("\n");
}
```

```
//Forma la figura
void FormLetter(int amount, int space, int i){
    if (amount>0){
        Spaces(space);
        LetterPNLJHFD(amount, i);
        FormLetter(amount-1, space+1, i+1);
    }
}
```

Diseño de pantalla:

```
P P P P P P P
N N N N N N
L L L L L
J J J J
H H H
F F
D
```

22.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una letra 'P' formada por
la repetición de la letra 'P' en líneas descendentes. Cada línea tiene una
cantidad
//decreciente de 'P's. La cantidad total de 'P's en la letra 'P' y el
espacio entre las columnas se pueden ajustar al modificar los parámetros en
la llamada a la
//función FormLetter() en la función main().

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int);

// Función para imprimir una serie de letras 'P' seguidas de un espacio.
```

```

void LetterP(int);

// Función para imprimir la letra 'P' formada por líneas descendentes.
void FormLetter(int, int);

int main() {
    // Llama a la función FormLetter() para imprimir la letra 'P' con
    // 13 líneas y un espacio inicial de 36 caracteres.
    FormLetter(13, 36);
    return 0;
}

// Función recursiva para imprimir espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" "); // Imprime un espacio.
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una
        // cantidad decreciente de espacios.
    }
}

// Función recursiva para imprimir 'P's con un espacio después de cada 'P'.
void LetterP(int amount) {
    if(amount > 0) {
        printf("P"); // Imprime un 'P'.
        Spaces(1); // Llama a Spaces() para imprimir un espacio.
        LetterP(amount - 1); // Llama recursivamente a LetterP() con una
        // cantidad decreciente de 'P's.
    } else {
        printf("\n"); // Cuando se completa una fila de 'P's, imprime un
        // salto de línea.
    }
}

// Función recursiva para imprimir la letra 'P' formada por líneas
// descendentes.
void FormLetter(int amount, int space) {
    if(amount > 0) {
        Spaces(space); // Llama a Spaces() para imprimir un espacio inicial.
        LetterP(amount - 6); // Llama a LetterP() para imprimir la cantidad
        // apropiada de 'P's en cada línea.
        FormLetter(amount - 1, space + 1); // Llama recursivamente a
        // FormLetter() con una cantidad decreciente de líneas y un aumento en el
        // espacio inicial.
    }
}

```



```
}
```

Diseño de pantalla:

```

P P P P P P P
 P P P P P P
  P P P P P
   P P P P
    P P P
     P P
      P

```

23.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una forma de letra 'P'
usando la letra 'P' en líneas horizontales y verticales, donde cada fila
tiene un número
//decreciente de 'P's. La cantidad total de filas y el espacio entre las
columnas se pueden ajustar modificando los parámetros en la llamada a la
función FormLetter()
//en la función main().

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int);

// Función para imprimir una serie de letras 'P' seguidas de un espacio.
void LetterP(int);

// Función para imprimir la forma de letra 'P' usando la letra 'P' en líneas
horizontales y verticales.
void FormLetter(int, int);
```

```

int main() {
    // Llama a la función FormLetter() para imprimir la forma de letra 'P'
    con 7 filas y sin espacio inicial.
    FormLetter(7, 0);
    return 0;
}

// Función recursiva para imprimir espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" "); // Imprime un espacio.
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una
        cantidad decreciente de espacios.
    }
}

// Función recursiva para imprimir 'P's con un espacio después de cada 'P'.
void LetterP(int amount) {
    if(amount > 0) {
        printf("P"); // Imprime un 'P'.
        Spaces(1); // Llama a Spaces() para imprimir un espacio.
        LetterP(amount - 1); // Llama recursivamente a LetterP() con una
        cantidad decreciente de 'P's.
    } else {
        printf("\n"); // Cuando se completa una fila de 'P's, imprime un
        salto de línea.
    }
}

// Función recursiva para imprimir la forma de letra 'P'.
void FormLetter(int amount, int space) {
    if(amount > 0) {
        Spaces(space); // Llama a Spaces() para imprimir espacios iniciales.
        LetterP(amount); // Llama a LetterP() para imprimir la cantidad
        apropiada de 'P's en cada fila.
        FormLetter(amount - 1, space + 1); // Llama recursivamente a
        FormLetter() con una cantidad decreciente de filas y un aumento en el
        espacio inicial.
    }
}

```

Diseño de pantalla:

```
P P P P P P P
P P P P P P
P P P P P
P P P P
P P P
P P P
P P
P
```

24.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una forma de letra 'A'
usando la letra 'A' en líneas horizontales y verticales, donde cada fila
tiene un número
//decreciente de 'A's. La forma de la letra 'A' se ajusta a la cantidad de
líneas y al espacio entre las columnas, que se especifican en la función
main().

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int);

// Función para imprimir una serie de letras 'A'.
void LetterA(int);

// Función para imprimir la forma de letra 'A' usando la letra 'A' en líneas
horizontales y verticales.
void FormLetter(int, int, int);

int main() {
```

```

    // Llama a la función FormLetter() para imprimir la forma de letra 'A'
    con 1 fila inicial, 5 espacios iniciales y el contador inicializado en 1.
    FormLetter(1, 5, 1);
    return 0;
}

// Función recursiva para imprimir espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" "); // Imprime un espacio.
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una
        cantidad decreciente de espacios.
    }
}

// Función recursiva para imprimir 'A's.
void LetterA(int amount) {
    if(amount > 0) {
        printf("A"); // Imprime una 'A'.
        LetterA(amount - 1); // Llama recursivamente a LetterA() con una
        cantidad decreciente de 'A's.
    }
}

//Funcion aux de FormLetter
void AuxFormLetter(int amount, int space){
    Spaces(27); // Imprime espacios para ajustar la posición de la letra
    'A'.
    LetterA(amount); // Llama a LetterA() para imprimir 'A's en la primera
    parte de la forma de letra 'A'.
    Spaces(space); // Imprime espacios entre las columnas.
    LetterA(amount); // Llama a LetterA() para imprimir 'A's en la segunda
    parte de la forma de letra 'A'.
    printf("\n"); // Imprime un salto de línea.
}

// Función recursiva para imprimir la forma de letra 'A'.
void FormLetter(int amount, int space, int counter) {
    if(counter < 8) {
        if(counter < 4) {
            AuxFormLetter(amount, space);
            FormLetter(amount + 1, space - 2, counter + 1); // Llama
            recursivamente a FormLetter() con más 'A's y menos espacio entre columnas.
        } else if(counter == 4) {
            Spaces(27); // Imprime espacios para ajustar la posición de la
            letra 'A'.
        }
    }
}

```

```

        LetterA(amount - 1); // Llama a LetterA() para imprimir 'A's en
la primera parte de la forma de letra 'A'.
        Spaces(space); // Imprime espacios entre las columnas.
        LetterA(amount); // Llama a LetterA() para imprimir 'A's en la
segunda parte de la forma de letra 'A'.
        printf("\n"); // Imprime un salto de línea.
        FormLetter(amount - 1, space + 2, counter + 1); // Llama
recursivamente a FormLetter() con menos 'A's y más espacio entre columnas.
    } else {
        AuxFormLetter(amount, space);
        FormLetter(amount - 1, space + 2, counter + 1); // Llama
recursivamente a FormLetter() con menos 'A's y más espacio entre columnas.
    }
}
}

```

Diseño de pantalla:

```

      A      A
     AA     AA
    AAA    AAA
   AAAAAA
  AAA  AAA
 AA   AA
 A    A

```

25.

```

//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una forma de letra 'Z'
utilizando la letra 'Z' en líneas horizontales, donde cada fila tiene un
número
// decreciente de 'Z's. La cantidad de filas y el espacio entre las columnas
se especifican en la función main().

```

```

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int);

// Función para imprimir una serie de letras 'Z'.
void LetterZ(int);

// Función para imprimir la forma de letra 'Z' utilizando la letra 'Z' en
líneas horizontales.
void FormLetter(int, int, int);

int main() {
    // Llama a la función FormLetter() para imprimir la forma de letra 'Z'
    con 1 fila inicial, 7 espacios iniciales y el contador inicializado en 1.
    FormLetter(1, 7, 1);
    return 0;
}

// Función recursiva para imprimir espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" "); // Imprime un espacio.
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una
        cantidad decreciente de espacios.
    }
}

// Función recursiva para imprimir 'Z's.
void LetterZ(int amount) {
    if(amount > 0) {
        printf("Z"); // Imprime una 'Z'.
        LetterZ(amount - 1); // Llama recursivamente a LetterZ() con una
        cantidad decreciente de 'Z's.
    } else {
        printf("\n"); // Cuando se completa una fila de 'Z's, imprime un
        salto de línea.
    }
}

// Función recursiva para imprimir la forma de letra 'Z'.
void FormLetter(int amount, int space, int counter) {
    if(counter <= 8) {
        if(counter <= 3) {

```

```

        Spaces(space); // Llama a Spaces() para imprimir espacios
        iniciales.
        LetterZ(amount); // Llama a LetterZ() para imprimir 'Z's en la
        parte ascendente de la forma de letra 'Z'.
        FormLetter(amount + 2, space - 1, counter + 1); // Llama
        recursivamente a FormLetter() con más 'Z's y menos espacio entre columnas.
    } else {
        Spaces(space); // Llama a Spaces() para imprimir espacios
        iniciales.
        LetterZ(amount); // Llama a LetterZ() para imprimir 'Z's en la
        parte descendente de la forma de letra 'Z'.
        FormLetter(amount - 2, space + 1, counter + 1); // Llama
        recursivamente a FormLetter() con menos 'Z's y más espacio entre columnas.
    }
}
}

```

Diseño de pantalla:

```

    Z
   ZZZ
  ZZZZZ
 ZZZZZZZ
  ZZZZZ
   ZZZ
    Z

```

26.

```

//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una forma de letra 'Z'
utilizando la letra 'Z' en líneas horizontales, donde cada fila tiene un
número decreciente

```

//de 'Z's. El espacio entre las columnas se ajusta en cada iteración de la función recursiva.

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

```
// Función para imprimir un número específico de espacios.  
void Spaces(int);
```

```
// Función para imprimir una serie de letras 'Z'.  
void LetterZ(int);
```

```
// Función para imprimir la forma de letra 'Z' utilizando la letra 'Z' en  
líneas horizontales.  
void FormLetter(int, int);
```

```
int main() {  
    // Llama a la función FormLetter() para imprimir la forma de letra 'Z'  
    con 17 espacios iniciales y el contador inicializado en 0.  
    FormLetter(17, 0);  
    return 0;  
}
```

```
// Función recursiva para imprimir espacios.  
void Spaces(int amount) {  
    if(amount > 0) {  
        printf(" "); // Imprime un espacio.  
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una  
cantidad decreciente de espacios.  
    }  
}
```

```
// Función recursiva para imprimir 'Z's.  
void LetterZ(int amount) {  
    if(amount > 0) {  
        printf("Z"); // Imprime una 'Z'.  
        LetterZ(amount - 1); // Llama recursivamente a LetterZ() con una  
cantidad decreciente de 'Z's'.  
    }  
}
```

```
// Función recursiva para imprimir la forma de letra 'Z'.  
void FormLetter(int space, int counter) {  
    if(counter < 9) {  
        Spaces(counter); // Llama a Spaces() para imprimir espacios  
iniciales.
```



```

        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
superior de la forma de letra 'Z'.
        Spaces(space); // Imprime espacios entre las columnas.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
inferior de la forma de letra 'Z'.
        printf("\n"); // Imprime un salto de línea.
        FormLetter(space - 2, counter + 1); // Llama recursivamente a
FormLetter() con menos espacio entre columnas y un contador incrementado.
    } else {
        Spaces(counter); // Llama a Spaces() para imprimir espacios
iniciales.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
final de la forma de letra 'Z'.
        printf("\n"); // Imprime un salto de línea.
    }
}
}

```

Diseño de pantalla:

27.

```

//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II

```

//Descripción del programa: Este programa imprime una forma de letra 'Z' utilizando la letra 'Z' en líneas horizontales, donde cada fila tiene un número decreciente //de 'Z's. El espacio entre las columnas se ajusta en cada iteración de la función recursiva.

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

```
// Función para imprimir un número específico de espacios.
```

```
void Spaces(int);
```

```
// Función para imprimir una serie de letras 'Z'.
```

```
void LetterZ(int);
```

```
// Función para imprimir la forma de letra 'Z' utilizando la letra 'Z' en líneas horizontales.
```

```
void FormLetter(int, int);
```

```
int main() {
```

```
    // Llama a la función FormLetter() para imprimir la forma de letra 'Z' con 0 espacios iniciales y el contador inicializado en 9.
```

```
    FormLetter(0, 9);
```

```
    return 0;
```

```
}
```

```
// Función recursiva para imprimir espacios.
```

```
void Spaces(int amount) {
```

```
    if(amount > 0) {
```

```
        printf(" "); // Imprime un espacio.
```

```
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una cantidad decreciente de espacios.
```

```
    }
```

```
}
```

```
// Función recursiva para imprimir 'Z's.
```

```
void LetterZ(int amount) {
```

```
    if(amount > 0) {
```

```
        printf("Z"); // Imprime una 'Z'.
```

```
        LetterZ(amount - 1); // Llama recursivamente a LetterZ() con una cantidad decreciente de 'Z's.
```

```
    }
```

```
}
```

```
// Función recursiva para imprimir la forma de letra 'Z'.
```

```
void FormLetter(int space, int counter) {
```

```

    if(counter == 9) {
        Spaces(counter); // Llama a Spaces() para imprimir espacios
        iniciales.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la primera
        fila.
        printf("\n"); // Imprime un salto de línea.
        FormLetter(space + 1, counter - 1); // Llama recursivamente a
        FormLetter() con más espacio entre columnas y un contador decreciente.
    } else if(counter > 0) {
        Spaces(counter); // Llama a Spaces() para imprimir espacios
        iniciales.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
        superior de la forma de letra 'Z'.
        Spaces(space); // Imprime espacios entre las columnas.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
        inferior de la forma de letra 'Z'.
        printf("\n"); // Imprime un salto de línea.
        FormLetter(space + 2, counter - 1); // Llama recursivamente a
        FormLetter() con más espacio entre columnas y un contador decreciente.
    } else {
        Spaces(counter); // Llama a Spaces() para imprimir espacios
        iniciales.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
        final de la forma de letra 'Z'.
        Spaces(space); // Imprime espacios entre las columnas.
        LetterZ(1); // Llama a LetterZ() para imprimir una 'Z' en la parte
        final de la forma de letra 'Z'.
    }
}

```

Diseño de pantalla:

```

      Z
     Z Z
    Z  Z
   Z   Z
  Z    Z
 Z     Z
Z      Z
Z       Z
Z        Z

```

28.

//Fecha: 11/03/2024

```

//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa imprime una forma de letra 'A'
utilizando la letra 'A' en líneas horizontales, donde cada fila tiene un
número decreciente
//de 'A's. El espacio entre las columnas se ajusta en cada iteración de la
función recursiva.

#include <stdio.h>

// Función para imprimir un número específico de espacios.
void Spaces(int);

// Función para imprimir una serie de letras 'A'.
void LetterA(int);

// Función para imprimir la forma de letra 'A' utilizando la letra 'A' en
líneas horizontales.
void FormLetter(int, int, int);

int main() {
    // Llama a la función FormLetter() para imprimir la forma de letra 'A'
    con 1 fila inicial, 39 espacios iniciales y el contador inicializado en 0.
    FormLetter(1, 39, 0);
    return 0;
}

// Función recursiva para imprimir espacios.
void Spaces(int amount) {
    if(amount > 0) {
        printf(" "); // Imprime un espacio.
        Spaces(amount - 1); // Llama recursivamente a Spaces() con una
        cantidad decreciente de espacios.
    }
}

```

```

// Función recursiva para imprimir 'A's.
void LetterA(int amount) {
    if(amount > 0) {
        printf("A"); // Imprime una 'A'.
        LetterA(amount - 1); // Llama recursivamente a LetterA() con una
cantidad decreciente de 'A's.
    } else {
        printf("\n"); // Cuando se completa una fila de 'A's, imprime un
salto de línea.
    }
}

// Función recursiva para imprimir la forma de letra 'A'.
void FormLetter(int amount, int space, int counter) {
    if(counter <= 10) {
        if(counter <= 4) {
            Spaces(space); // Llama a Spaces() para imprimir espacios
iniciales.
            LetterA(amount); // Llama a LetterA() para imprimir 'A's en la
parte ascendente de la forma de letra 'A'.
            FormLetter(amount + 1, space - 1, counter + 1); // Llama
recursivamente a FormLetter() con más 'A's y menos espacio entre columnas.
        } else {
            Spaces(space); // Llama a Spaces() para imprimir espacios
iniciales.
            LetterA(amount); // Llama a LetterA() para imprimir 'A's en la
parte descendente de la forma de letra 'A'.
            FormLetter(amount - 1, space + 1, counter + 1); // Llama
recursivamente a FormLetter() con menos 'A's y más espacio entre columnas.
        }
    }
}

```

Diseño de pantalla:

```
A
AA
AAA
AAAA
AAAAA
AAAAAA
AAAAA
AAAA
AAA
AA
A
```

29.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de
la serie de Taylor para  $e^x$  hasta un número dado de términos. El usuario
puede ingresar
// el número de términos y el valor de x.

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

// Función recursiva para calcular el factorial de un número.
long long int DoFactorial(int number) {
    if(number == 0)
        return 1;
    else
        return number * DoFactorial(number - 1);
}

// Función para calcular un término de la serie de Taylor.
float TermTaylor(int i, int x) {
    return pow(x, i) / DoFactorial(i);
}
```

```

// Función recursiva para calcular la suma de los términos de la serie de
Taylor.
float SummationTaylor(int n, int i, int x) {
    if(i > n) {
        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}

// Función recursiva para imprimir los términos de la serie de Taylor y su
suma.
void PrintTaylor(int term, int quantity, int x) {
    if(term < quantity) {
        printf("%.8f + ", TermTaylor(term, x));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("%.8f = ", TermTaylor(term, x));
        printf("%.8f", SummationTaylor(quantity, 0, x));
    }
}

int main() {
    int quantityTerms = 0; // Cantidad terminos
    int x = 0; // Valor asignado por el usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de
Taylor de e^x: ");
    scanf("%i", &quantityTerms);

    printf("Ingrese el valor de la x que desea tomar: ");
    scanf("%i", &x);

    PrintTaylor(0, quantityTerms, x);

    return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de e^x: 5
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 1.00000000 + 0.50000000 + 0.16666667 + 0.04166667 + 0.00833333 = 2.71666670

```

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor de e^x: 9
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 1.00000000 + 0.50000000 + 0.16666667 + 0.04166667 + 0.00833333 + 0.00138889 + 0.00019841 + 0.00002480 + 0.00000276 = 2.71828151
```

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor de e^x: 1
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 1.00000000 = 2.00000000
```

30.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de
la serie de Taylor para cos(x) hasta un número dado de términos.
//El usuario puede ingresar el número de términos y el valor de x.

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

// Función recursiva para calcular el factorial de un número.
long long int DoFactorial(int number) {
    if(number == 0)
        return 1;
    else
        return number * DoFactorial(number - 1);
}

// Función para calcular un término de la serie de Taylor para cos(x).
float TermTaylor(int i, int x) {
    return (pow(-1, i) * pow(x, 2 * i)) / DoFactorial(2 * i);
}

// Función recursiva para calcular la suma de los términos de la serie de
Taylor para cos(x).
float SummationTaylor(int n, int i, int x) {
    if(i > n) {
```



```

        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}

// Función recursiva para imprimir los términos de la serie de Taylor y su
suma.
void PrintTaylor(int term, int quantity, int x) {
    if(term < quantity) {
        printf("%.8f) + ", TermTaylor(term, x));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("%.8f) = ", TermTaylor(term, x));
        printf("%.8f", SummationTaylor(quantity, 0, x));
    }
}

int main() {
    int quantityTerms = 0; // Cantidad terminos
    int x = 0; // Valor asignado por el usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de
Taylor de cos(x): ");
    scanf("%i", &quantityTerms);

    printf("Ingrese el valor de la x que desea tomar: ");
    scanf("%i", &x);

    PrintTaylor(0, quantityTerms, x);

    return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de cos(x): 2
Ingrese el valor de la x que desea tomar: 1
(1.00000000) + (-0.50000000) + (0.04166667) = 0.54166663

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de cos(x): 6
Ingrese el valor de la x que desea tomar: 1
(1.00000000) + (-0.50000000) + (0.04166667) + (-0.00138889) + (0.00002480) + (-0.00000028) + (0.00000000) = 0.54030228

```

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor de cos(x): 9
Ingrese el valor de la x que desea tomar: 1
(1.00000000) + (-0.50000000) + (0.04166667) + (-0.00138889) + (0.00002480) + (-0.00000028) + (0.00000000) + (-0.00000000) + (0.00000000) + (-0.00000000) = 0.54030228
```

31.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de
la serie de Taylor para sinh(x) hasta un número dado de términos.
//El usuario puede ingresar el número de términos y el valor de x.

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

// Función recursiva para calcular el factorial de un número.
long long int DoFactorial(int number) {
    if(number == 0)
        return 1;
    else
        return number * DoFactorial(number - 1);
}

// Función para calcular un término de la serie de Taylor para sinh(x).
float TermTaylor(int i, int x) {
    return pow(x, (2 * i) + 1) / DoFactorial((2 * i) + 1);
}

// Función recursiva para calcular la suma de los términos de la serie de
Taylor para sinh(x).
float SummationTaylor(int n, int i, int x) {
    if(i > n) {
        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}
```

```

    }
}

// Función recursiva para imprimir los términos de la serie de Taylor y su
suma.
void PrintTaylor(int term, int quantity, int x) {
    if(term < quantity) {
        printf("%.8f + ", TermTaylor(term, x));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("%.8f = ", TermTaylor(term, x));
        printf("%.8f", SummationTaylor(quantity, 0, x));
    }
}

int main() {
    int quantityTerms = 0; // Cantidad terminos
    int x = 0; // Valora asignado por usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de
Taylor del sinh(x): ");
    scanf("%i", &quantityTerms);

    printf("Ingrese el valor de la x que desea tomar: ");
    scanf("%i", &x);

    PrintTaylor(0, quantityTerms, x);

    return 0;
}

```

Diseño de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sinh(x): 2
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.16666667 + 0.00833333 = 1.17499995

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sinh(x): 5
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.16666667 + 0.00833333 + 0.00019841 + 0.00000276 + 0.00000003 = 1.17520118

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sinh(x): 9
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.16666667 + 0.00833333 + 0.00019841 + 0.00000276 + 0.00000003 + 0.00000000 + 0.00000000 + 0.00000000 = 1.17520118

```

32.

```
//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de
la serie de Taylor para cosh(x) hasta un número dado de términos.
// El usuario puede ingresar el número de términos y el valor de x.

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

// Función recursiva para calcular el factorial de un número.
long long int DoFactorial(int number) {
    if(number == 0)
        return 1;
    else
        return number * DoFactorial(number - 1);
}

// Función para calcular un término de la serie de Taylor para cosh(x).
float TermTaylor(int i, int x) {
    return pow(x, 2 * i) / DoFactorial(2 * i);
}

// Función recursiva para calcular la suma de los términos de la serie de
Taylor para cosh(x).
float SummationTaylor(int n, int i, int x) {
    if(i > n) {
        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}
```

```
// Función recursiva para imprimir los términos de la serie de Taylor y su
suma.
void PrintTaylor(int term, int quantity, int x) {
    if(term < quantity) {
        printf("%.8f + ", TermTaylor(term, x));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("%.8f = ", TermTaylor(term, x));
        printf("%.8f", SummationTaylor(quantity, 0, x));
    }
}

int main() {
    int quantityTerms = 0; // Cantidad terminos
    int x = 0; // Valora asignado por usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de
Taylor del cosh(x): ");
    scanf("%i", &quantityTerms);

    printf("Ingrese el valor de la x que desea tomar: ");
    scanf("%i", &x);

    PrintTaylor(0, quantityTerms, x);

    return 0;
}
```

Diseños de pantalla:

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor del cosh(x): 2
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.50000000 + 0.04166667 = 1.54166675
```

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor del cosh(x): 5
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.50000000 + 0.04166667 + 0.00138889 + 0.00002480 + 0.00000028 = 1.54308057
```

```
Ingrese el numero de terminos que desea conocer de la serie de Taylor del cosh(x): 9
Ingrese el valor de la x que desea tomar: 1
1.00000000 + 0.50000000 + 0.04166667 + 0.00138889 + 0.00002480 + 0.00000028 + 0.00000000 + 0.00000000 + 0.00000000 + 0.00000000 = 1.54308057
```

33.

//Fecha: 11/03/2024

//Hora: 8:46 PM

```

//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de
la serie de Taylor para  $\ln(x)$  hasta un número dado de términos.
// El usuario puede ingresar el número de términos y el valor de x.

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

// Función para calcular un término de la serie de Taylor para  $\ln(x)$ .
float TermTaylor(int i, float x) {
    return (pow(-1, (i - 1)) * pow(x, i) / i);
}

// Función recursiva para calcular la suma de los términos de la serie de
Taylor para  $\ln(x)$ .
float SummationTaylor(int n, int i, float x) {
    if(i > n) {
        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}

// Función recursiva para imprimir los términos de la serie de Taylor y su
suma.
void PrintTaylor(int term, int quantity, float x) {
    if(term < quantity) {
        printf("(%.8f) + ", TermTaylor(term, x - 1));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("(%.8f) = ", TermTaylor(term, x - 1));
        printf("(%.8f)", SummationTaylor(quantity, 1, x - 1));
    }
}

```

```

int main() {
    int quantityTerms = 0; // Cantidad terminos
    float x = 0; // Valor asignado por usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de Taylor de ln(x): ");
    scanf("%i", &quantityTerms);

    printf("Ingrese el valor de la x que desea tomar: ");
    scanf("%f", &x);

    PrintTaylor(1, quantityTerms+1, x);

    return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de ln(x): 2
Ingrese el valor de la x que desea tomar: 0.5
(-0.50000000) + (-0.12500000) + (-0.04166667) = (-0.66666669)

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de ln(x): 5
Ingrese el valor de la x que desea tomar: 0.5
(-0.50000000) + (-0.12500000) + (-0.04166667) + (-0.01562500) + (-0.00625000) + (-0.00260417) = (-0.69114584)

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor de ln(x): 9
Ingrese el valor de la x que desea tomar: 0.5
(-0.50000000) + (-0.12500000) + (-0.04166667) + (-0.01562500) + (-0.00625000) + (-0.00260417) + (-0.00111607) + (-0.00048828) + (-0.00021701) + (-0.00009766) = (-0.69306487)

```

34.

```

//Fecha: 11/03/2024
//Hora: 8:46 PM
//Versión del código: 1.0
//Autor: Esteban Vásquez Castañeda. Ing. C
//Lenguaje: C
//Versión del lenguaje: C18
//Versión del compilador: 13.2.0
//Nombre del S.O.: Windows 11
//Versión del S.O.: 10.0.22621
//Presentado a: Doctor Ricardo Moreno Laverde
//Universidad Tecnológica De Pereira
//Programa de Ingeniería de Sistemas y Computación
//Asignatura: IS284 Programación II
//Descripción del programa: Este programa calcula y muestra los términos de la serie de Taylor para sen(x) hasta un número dado de términos.
// El usuario puede ingresar el número de términos y el valor de x.

```

```

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

// Función recursiva para calcular el factorial de un número.
long long int DoFactorial(int number) {
    if(number == 0)
        return 1;
    else
        return number * DoFactorial(number - 1);
}

// Función para calcular un término de la serie de Taylor para sen(x).
float TermTaylor(int i, int x) {
    return (pow(-1, i) * pow(x, (2 * i) + 1) / DoFactorial((2 * i) + 1));
}

// Función recursiva para calcular la suma de los términos de la serie de Taylor para sen(x).
float SummationTaylor(int n, int i, int x) {
    if(i > n) {
        return 0;
    } else {
        return TermTaylor(i, x) + SummationTaylor(n, i + 1, x);
    }
}

// Función recursiva para imprimir los términos de la serie de Taylor y su suma.
void PrintTaylor(int term, int quantity, int x) {
    if(term < quantity) {
        printf("%.8f + ", TermTaylor(term, x));
        PrintTaylor(term + 1, quantity, x);
    } else {
        printf("%.8f = ", TermTaylor(term, x));
        printf("%.8f", SummationTaylor(quantity, 0, x));
    }
}

int main() {
    int quantityTerms = 0; // Cantidad terminos
    int x = 0; // Valor asignado por usuario

    printf("Ingrese el numero de terminos que desea conocer de la serie de Taylor del sen(x): ");

```



```

scanf("%i", &quantityTerms);

printf("Ingrese el valor de la x que desea tomar: ");
scanf("%i", &x);

PrintTaylor(0, quantityTerms, x);

return 0;
}

```

Diseños de pantalla:

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sen(x): 2
Ingrese el valor de la x que desea tomar: 1
1.00000000 + -0.16666667 + 0.00833333 = 0.84166670

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sen(x): 5
Ingrese el valor de la x que desea tomar: 1
1.00000000 + -0.16666667 + 0.00833333 + -0.00019841 + 0.00000276 + -0.00000003 = 0.84147096

```

```

Ingrese el numero de terminos que desea conocer de la serie de Taylor del sen(x): 9
Ingrese el valor de la x que desea tomar: 1
1.00000000 + -0.16666667 + 0.00833333 + -0.00019841 + 0.00000276 + -0.00000003 + 0.00000000 + -0.00000000 + 0.00000000 + -0.00000000 = 0.84147096

```

Enlace del canal de YouTube y observar en las listas de reproducción donde se encuentra adjuntas ordenadamente cada punto: <https://www.youtube.com/@IngEstebanVasquezCastaneda>