**Implementar ecuación de segundo grado y la serie de Taylor**

1.- Ecuación de 2° grado

Versión 1:

#include <stdio.h>

#include <math.h>

int main() {

double a, b, c;

printf("Ingrese los coeficientes (a, b, c) de la ecuacion cuadratica:\n");

scanf("%lf %lf %lf", &a, &b, &c);

if (a == 0) {

printf("El coeficiente 'a' debe ser diferente de 0.\n");

return 1; // Termina el programa con error

}

// Calcula el discriminante

double discriminante = b \* b - 4 \* a \* c;

if (discriminante > 0) {

// Dos raices reales y diferentes

double raiz1 = (-b + sqrt(discriminante)) / (2 \* a);

double raiz2 = (-b - sqrt(discriminante)) / (2 \* a);

printf("Raices reales y diferentes:\n");

printf("Raiz 1 = %.2lf\n", raiz1);

printf("Raiz 2 = %.2lf\n", raiz2);

} else if (discriminante == 0) {

// Raices reales e iguales

double raiz = -b / (2 \* a);

printf("Raices reales e iguales:\n");

printf("Raiz = %.2lf\n", raiz);

} else {

// Raices complejas

double parteReal = -b / (2 \* a);

double parteImaginaria = sqrt(-discriminante) / (2 \* a);

printf("Raices complejas:\n");

printf("Raiz 1 = %.2lf + %.2lfi\n", parteReal, parteImaginaria);

printf("Raiz 2 = %.2lf - %.2lfi\n", parteReal, parteImaginaria);

}

return 0;

}

Version 2:

#include <stdio.h>

#include <math.h>

int main() {

float a, b, c;

printf("Ingrese el coeficiente 'a' de la ecuacion cuadratica:\n");

scanf("%f", &a);

if (a != 0) {

printf("Ingrese los coeficientes 'b' y 'c' de la ecuacion cuadratica:\n");

scanf("%f %f", &b, &c);

// Calcula el discriminante

float discriminante = b \* b - 4 \* a \* c;

if (discriminante > 0) {

// Dos raices reales y diferentes

float raiz1 = (-b + sqrtf(discriminante)) / (2 \* a);

float raiz2 = (-b - sqrtf(discriminante)) / (2 \* a);

printf("Raices reales y diferentes:\n");

printf("Raiz 1 = %.2f\n", raiz1);

printf("Raiz 2 = %.2f\n", raiz2);

} else if (discriminante == 0) {

// Raices reales e iguales

float raiz = -b / (2 \* a);

printf("Raices reales e iguales:\n");

printf("Raiz = %.2f\n", raiz);

} else {

// Raices complejas

float parteReal = -b / (2 \* a);

float parteImaginaria = sqrtf(-discriminante) / (2 \* a);

printf("Raices complejas:\n");

printf("Raiz 1 = %.2f + %.2fi\n", parteReal, parteImaginaria);

printf("Raiz 2 = %.2f - %.2fi\n", parteReal, parteImaginaria);

}

} else {

printf("El coeficiente 'a' debe ser diferente de cero.\n");

return 1; // Termina el programa con error

}

return 0;

}

**2.- Serie de Taylor**

**2.1.- Potencia:**

#include <stdio.h>

int main() {

double x;

int n;

printf("Ingrese el valor de x: ");

scanf("%lf", &x);

printf("Ingrese el valor de n: ");

scanf("%d", &n);

double result = 1.0;

for ( i = 0; i < n; i++) {

result \*= x;

}

printf("%.2lf elevado a %d es igual a %.2lf\n", x, n, result);

return 0;

}

**2.2.- Factorial**

**a.- Solución general**

#include <stdio.h>

int main() {

int n;

printf("Ingrese un número para calcular su factorial: ");

scanf("%d", &n);

if (n < 0) {

printf("No se puede calcular el factorial de un número negativo.\n");

return 1; // Termina el programa con error

}

int factorial = 1;

for (int i = 1; i <= n; ++i) {

factorial \*= i;

}

printf("El factorial de %d es %d\n", n, factorial);

return 0;

}

**b.- Solución recursiva**

#include <stdio.h>

int factorial(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

int main() {

int n;

printf("Ingrese un número para calcular su factorial: ");

scanf("%d", &n);

if (n < 0) {

printf("No se puede calcular el factorial de un número negativo.\n");

return 1; // Termina el programa con error

}

int result = factorial(n);

printf("El factorial de %d es %d\n", n, result);

return 0;

}

**2.3.- Serie de Taylor para la función exponencial**

#include <stdio.h>

double power(double x, int n) {

double result = 1.0;

for (int i = 0; i < n; ++i) {

result \*= x;

}

return result;

}

int factorial(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

double exponential(double x) {

double result = 0.0;

int i = 0;

while (power(x, i) / factorial(i) > 0.001) {

result += power(x, i) / factorial(i);

i++;

}

return result;

}

int main() {

double x;

printf("Ingrese el valor de x: ");

scanf("%lf", &x);

double exp\_x = exponential(x);

printf("e^%.2lf utilizando la serie de Taylor es aproximadamente %.6lf\n", x, exp\_x);

return 0;

}

**Versión con float en vez de double**.

#include <stdio.h>

float power(float x, int n) {

float result = 1.0;

for (int i = 0; i < n; ++i) {

result \*= x;

}

return result;

}

int factorial(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

float exponential(float x) {

float result = 0.0;

float termino = 1;

int i = 0;

while (termino > 0.001) {

termino = power(x, i) / factorial(i);

result = result + termino;

i++;

}

return result;

}

int main() {

float x;

printf("Ingrese el valor de x: ");

scanf("%f", &x);

float exp\_x = exponential(x);

printf("e^%.2f utilizando la serie de Taylor es aproximadamente %.6f\n", x, exp\_x);

return 0;

}