

Práctica 0. Programa para calcular las raíces de un polinomio de segundo grado

1st José Luis Aguilar
Departamento de aprender a programar
Universidad MdG
Ciudad de México, México
aguilarch.joseluis@gmail.com

Resumen—Este documento presenta la documentación de un programa en C para encontrar las soluciones reales de una ecuación de segundo grado. Se presentan los requisitos del programa, alcances y limitaciones, así como herramientas de diseño de la solución utilizadas.

Index Terms—programación, Clang, introducción, ecuaciones de segundo grado.

I. DESCRIPCIÓN DEL PROBLEMA

Se requiere un programa que al ingresarle los coeficientes de un polinomio de segundo grado de una sola variable en su forma canónica, devuelva las raíces correspondientes. La forma general del polinomio es dado por la siguiente ecuación:

$$ax^2 + bx + c = 0$$

Y sus soluciones serán denotadas como x_1 y x_2 , independientemente de si sean iguales o distintas las soluciones. Las entradas y salidas serán todas por la terminal, y no se utiliza otra interfaz gráfica para el usuario.

II. COMPORTAMIENTO DETALLADO

El programa solicita los coeficientes del polinomio de segundo grado. Posteriormente valida que todos los coeficientes sean mayores a cero, y procede a calcular las raíces. Si las raíces son complejas, el programa devuelve las raíces iguales a cero, y muestra una advertencia que no representa a la solución verdadera.

Las entradas que necesita el programa son los coeficientes cuadrático, lineal e independiente del polinomio— a , b y c respectivamente.

La salida que devuelve son las raíces, denotadas como x_1 y x_2 .

III. PROCESOS

A continuación se muestra el proceso detallado del programa, separado en pasos para facilitar su lectura:

- **Mensaje introductorio:** se menciona que el programa encuentra las raíces del polinomio de segundo grado.
- **Entrada de datos:** se solicitan en orden los coeficientes de los términos cuadrático, lineal e independiente en ese orden. El programa verifica que todos los coeficientes

introducidos sean escritos en su forma numérica¹ y que sean mayores a 0.

- **Cálculo de raíces:** Una vez ingresados los datos, se procede a calcular las raíces, siguiendo los siguientes pasos:

- Se calcula el discriminante de la ecuación de segundo grado, definido de la siguiente forma:

$$\Delta = b^2 - 4ac$$

- Si el discriminante es mayor que cero, se procede a calcular las raíces. Si es menor a cero, se devuelven ceros como las soluciones, pues solamente calcula las raíces reales.
- Para calcular las raíces, se utiliza la fórmula general para las raíces de segundo grado, descrito en la siguiente ecuación:

$$\frac{-b \pm \sqrt{\Delta}}{2a}$$

- **Mostrar resultados:** se despliegan las raíces x_1 y x_2 .

IV. ALCANCES Y LIMITACIONES

El programa puede tomar como entradas a coeficientes racionales y mayores a cero. No recibe como argumentos a números escritos con letra, ni a números negativos. Solamente puede calcular raíces reales, y no puede encontrar raíces complejas.

V. DISEÑO DE PANTALLA

Las interacciones ocurren todas en la terminal. Por lo tanto, el diseño de pantalla es sumamente sencillo. A continuación se muestra un escenario en el que se use el programa:

@default

```
1 $pract0.c
2 Este programa calcula las raíces de un
3 polinomio de segundo grado
4 descritos en su forma canónica. Su forma
5 canónica es la siguiente:
6 a*x^2 + b*x + c = 0
```

¹No se pueden introducir datos de otra manera, (e.g. "DOS"no es una entrada válida, pero "2"sí

```

6 Inserte el coeficiente del termino
cuadratico:
7 3
8 Inserte el coeficiente del termino
lineal:
9 x
10 El valor otorgado no es un numero, o es
menor a 0.
11 Intente de nuevo.
12 10000
13 Inserte el coeficiente del termino
independiente:
14 2
15 Las raices para el polinomio:
16 3.00 x^2 + 10000.00 x + 2.00 = 0
17 Son las siguientes:
18 x1 = -0.000200
19 x2 = -3333.333133

```

VI. DISEÑO DE SOLUCIÓN

A continuación se muestra el pseudocódigo desarrollado para el programa. Anexo a este documento, se encuentra una prueba de escritorio para observar el comportamiento del programa en una situación ficticia. Se eliminan todos los acentos del pseudocódigo para hacer el render, pero se toma en cuenta que contiene errores ortográficos.

VI-A. Pseudocódigo

```

1 MAIN
2 float a = 0, b = 0, c = 0, x1 = 0, x2 =
0;
3
4 Print("Este programa calcula las raices
de un polinomio de segundo grado \n");
5 Print("descritos en su forma canonica.
Su forma cannica es la siguiente: \n");
6 Print("a*x^2 + b*x + c = 0 \n");
7
8 PedirValor(&a, "Inserte el coeficiente
del termino cuadratico:\n");
9 PedirValor(&b, "Inserte el coeficiente
del termino lineal:\n");
10 PedirValor(&c, "Inserte el coeficiente
del termino independiente:\n");
11 CalcularRaices(a, b, c, &x1, &x2);
12 Desplegar(a, b, c, x1, x2);
13
14 // Funciones auxiliares
15
16 Void PedirValor(float *a, string
description);
17
18     int flag = 1;
19     Print(description);
20     while (flag == 0) then:

```

```

21     Read(a);
22     if *a > 0
23         flag = 1;
24     else
25         Print("El valor otorgado no es un
numero, o es menor a 0. Intente de
nuevo.\n");
26     FIN
27
28 Void CalcularRaices(float a, float b,
float c, float *x1, float *x2)
29     float discriminante = power(b,2) -
4*a*c;
30
31     if discriminante < 0
32         Print("El valor del discriminante
del polinomio es menor a cero.\n");
33         Print("El programa no permite
resolver polinomios con raices
complejas.\n
34         El programa devolvera ceros. \n");
35     else
36         *x1 = (-b +
sqrt(discriminante))/(2*a);
37         *x2 = (-b -
sqrt(discriminante))/(2*a);
38
39
40 Void Desplegar(float a, float b, float
c, float x1, float x2)
41     Print(
42         "Las raices para el polinomio:\n
43         __float__x^2 +
__float__x+__float__=0\n
44         Son las siguientes:\n
45         x1 = __int__\n
46         x2 = __int__,
47         a, b, c, x1, x2
48     );

```