A close-up of a blue button

Description automatically generated

**UBND TP. HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC SÀI GÒN**

Bài tập 1 (làm nhóm)

**Môn:** Kiểm thử phần mềm

**Lớp:** DCT121C3

**Giảng viên hướng dẫn:** Đỗ Như Tài

**Sinh viên thực hiện:** 3121411052 - Nhâm Minh Đạt 3121411192 - Nguyễn Chí Tân

3121411125 – Nguyễn Tuấn Long

**Năm học: 2024-2025 / HK1**

**Bảng phân công**

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Nhiệm vụ |
| Nhâm Minh Đạt | 3121411052 | * Thực hiện làm giao diện, thiết kế cho bài toán 1 và bài toán 2. * Thực hiện cài đặt mã giả và viết chương trình python cho bài 2 và hỗ trợ cài đặt cho bài 1. * Thực hiện kiểm thử cả 2 bài toán |
| Nguyễn Chí Tân | 3121411192 | * Thực hiện giao diện, thiết kết cho bài toán 1 và bài toán 2 * Thực hiện cài đặt mã giả và viết chương trình python cho bài toán 1 và hỗ trợ cài đặt bài 2. Viết handtrace cho 2 bài toán |
| Nguyễn Tuấn Long | 3121411125 | * Thực hiện phân tích 2 bài toán |

**Bài toán 1**: Viết chương trình giải phương trình bậc 2 có dạng: ax²+ bx + c = 0

**Phân tích bài toán**: Người dùng nhập vào 3 số a, b và c từ bàn phím, chương trình thực hiện tính toán in ra kết quả của phương trình.

**Input**: a, b, c: float

**Output**: sn, x1, x2

Với 1 trong 4 kết quả sẽ có được, bao gồm:

1. sn = -1: “Phương trình vô số nghiệm“

2. sn = 0: “Phương trình vô nghiệm”

3. sn = 1: “Phương trình có nghiệm kép x = ”

4. sn = 2: “Phương trình có 2 nghiệm phân biệt: x1 = , x2 = “

**Ràng buộc**: x1 < x2

# Giao diện: Sau khi chạy chương trình yêu cầu người dùng nhập vào 3 số a,b,c cách nhau bởi khoảng trắng

# Ví dụ 1:

# Nhập 0 0 0 vào terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Phương trình vô số nghiệm

-------------------------------------------

# Ví dụ 2:

# Nhập vào 0 0 1 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Phương trình vô nghiệm:

-------------------------------------------

# Ví dụ 3

# Nhập vào 1 2 1 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 1 nghiệm x = -1

# Ví dụ 4

# Nhập vào 1 4 1 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 2 nghiệm phân biệt = -0.2679491924311228 và = -3.732050807568877

# Thiết kế:

# ta có

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Cài đặt:

# Handtrace

Testcase 1:

0x² + 0x + 0 = 0

Do tổng vế trái luôn bằng 0 với mọi x nên ta kết luận:

Phương trình vô số nghiệm

Testcase 2:

0x² + 0x + 1 = 0

Do tổng vế trái luôn khác 0 với mọi x nên ta kết luận:

Phương trình vô nghiệm

Testcase 3:

0x² + 1x - 2 = 0

Suy ra phương trình bậc 1:

1x - 2 = 0 → 1x = 2 → → x = 2

Phương trình có 1 nghiệm x = 2

Testcase 4:

1x² + 2x - 2 = 0

Tính (delta):

Phương trình vô nghiệm

Testcase 5:

1x² + 2x - 1 = 0

Tính (delta):

Phương trình có nghiệm kép

Phương trình có 1 nghiệm x =

Testcase 6:

1x² + 2x - 1 = 0

Tính (delta):

Kết luận: Phương trình có 2 nghiệm phân biệt

Phương trình có 2 nghiệm phân biệt:

= -0.2679491924311228

= -3.732050807568877

# Mã giả

BEGIN

INPUT a, b, c // Nhập các hệ số a, b, c của phương trình bậc 2

IF a = 0 THEN

IF b = 0 THEN

IF c = 0 THEN

// Phương trình vô số nghiệm. Testcase 1

sn = -1

ELSE

// Phương trình vô nghiệm. Testcase 2

sn = 0

END IF

ELSE

// Phương trình bậc nhất bx + c = 0. Testcase 3

sn = 1

x1 = -c / b

END IF

ELSE

// Tính biệt thức Delta

Delta = b^2 - 4 \* a \* c

IF Delta < 0 THEN

// Phương trình vô nghiệm. Testcase 4

sn = 0

ELSE IF Delta = 0 THEN

// Phương trình có nghiệm kép. Testcase 5

sn = 1

x1 = -b / (2 \* a)

ELSE

// Phương trình có hai nghiệm phân biệt. Testcase 6

sn = 2

x1 = (-b + sqrt(Delta)) / (2 \* a)

x2 = (-b - sqrt(Delta)) / (2 \* a)

// Ràng buộc x1 < x2

IF x1 > x2 THEN

SWAP x1, x2

END IF

END IF

END IF

// Hiển thị kết quả dựa trên giá trị của sn

IF sn = -1 THEN

PRINT "Phương trình vô số nghiệm."

ELSE IF sn = 0 THEN

PRINT "Phương trình vô nghiệm."

ELSE IF sn = 1 THEN

PRINT "Phương trình có 1 nghiệm: x = ", x1

ELSE IF sn = 2 THEN

PRINT "Phương trình có 2 nghiệm phân biệt: x1 = ", x1, ", x2 = ", x2

END IF

END

# Python Implementation

# Lưu ý: Toàn bộ quá trình thực hiện được ghi lại bằng các hình anh thông qua thư mục “Quá trình cài đặt ” được đính kèm.

# Các hình cài đặt cho phương trình bậc 2 là từ 1-25.

# Các hình cài đặt cho phương trình trùng phương là từ A-Z1

# import math

# def solve\_equation(a, b, c):

# if a == 0:

# if b == 0:

# if c == 0:

# return -1, None, None # Infinite solutions

# else:

# return 0, None, None # No solution

# else:

# return 1, -c / b, None # Linear equation

# else:

# Delta = b \*\* 2 - 4 \* a \* c

# if Delta < 0:

# return 0, None, None # No real solution

# elif Delta == 0:

# return 1, -b / (2 \* a), None # Repeated solution

# else:

# x1 = (-b + math.sqrt(Delta)) / (2 \* a)

# x2 = (-b - math.sqrt(Delta)) / (2 \* a)

# if x1 > x2:

# x1, x2 = x2, x1

# return 2, x1, x2 # Two distinct solutions

# # User inputs

# if \_\_name\_\_ == "\_\_main\_\_":

# # Read a single line from standard input

# line = input()

# a, b, c = map(float, line.split())

# sn, x1, x2 = solve\_equation(a, b, c)

# if sn == -1:

# print("Phương trình vô số nghiệm.")

# elif sn == 0:

# print("Phương trình vô nghiệm.")

# elif sn == 1:

# print(f"Phương trình có 1 nghiệm: x = {x1}")

# elif sn == 2:

# print(f"Phương trình có 2 nghiệm phân biệt: x1 = {x1}, x2 = {x2}")

# Testing

# test\_ptbac2.ipynb

%%writefile ptbac2.in1

0 0 0

%%writefile ptbac2.in2

0 0 1

%%writefile ptbac2.in3

0 2 4

%%writefile ptbac2.in4

1 0 4

%%writefile ptbac2.in5

1 2 1

%%writefile ptbac2.in6

1 -3 2

!echo Test Case 1:

!python ptbac2.py < ptbac2.in1

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 2:

!python ptbac2.py < ptbac2.in2

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 3:

!python ptbac2.py < ptbac2.in3

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 4:

!python ptbac2.py < ptbac2.in4

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 5:

!python ptbac2.py < ptbac2.in5

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 6:

!python ptbac2.py < ptbac2.in6

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Testing output:

Test Case 1:

Phương trình vô số nghiệm.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 2:

Phương trình vô nghiệm.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 3:

Phương trình có 1 nghiệm: x = -2.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 4:

Phương trình vô nghiệm.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 5:

Phương trình có 1 nghiệm: x = -1.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 6:

Phương trình có 2 nghiệm phân biệt: x1 = 1.0, x2 = 2.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Bài toán 2**:

Giải phương trình trùng phương có dạng: ax⁴ + bx² + c = 0

**Sơ lược về bài toán**: Cho người dùng nhập vào 3 số a, b và c từ bàn phím, chương trình thực hiện tính toán in ra kết quả của phương trình.

**Input**: a, b, c: float

**Output**: sn, x1, x2, x3, x4

Với 1 trong 6 kết quả sẽ có được, bao gồm:

1. sn = -1: “Phương trình vô số nghiệm“

2. sn = 0: “Phương trình vô nghiệm”

3. sn = 1: “Phương trình có 1 nghiệm/ nghiệm kép x = ”

4. sn = 2: “Phương trình có 2 nghiệm phân biệt: x1 = , x2 = “

5. sn = 3: “Phương trình có 3 nghiệm phân biệt: x1 = , x2 = , x3= “

6. sn = 4: “Phương trình có 4 nghiệm phân biệt: x1 = , x2 = , x3= , x4 = “

**Ràng buộc**: t1 = x1^2, t2 = x2^2, t2 > t1

# Giao diện: Sau khi chạy chương trình yêu cầu người dùng nhập vào 3 số a,b,c cách nhau bởi khoảng trắng

# Ví dụ 1:

# Nhập 0 0 0 vào terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Phương trình vô số nghiệm

-------------------------------------------

# Ví dụ 2:

# Nhập vào 0 0 1 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Phương trình vô nghiệm:

-------------------------------------------

# Ví dụ 3

# Nhập vào 1 0 0 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 1 nghiệm x = 0

# Ví dụ 4

# Nhập vào 1 -2 1 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 2 nghiệm phân biệt = 1 và = -1

# Ví dụ 5

# Nhập vào 1 -4 0 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 3 nghiệm phân biệt = 0, = 4 và

# Ví dụ 6

# Nhập vào 1 -5 4 trong terminal => Nhấn Enter

Chương trình bắt đầu xử lí dữ liệu và đưa ra kết quả trên terminal:

Kết luận: Phương trình có 2 nghiệm phân biệt = 1, = -1, và = -2

**Thiết kế:**

# ta có và hàm giải phương trình bậc 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Cài đặt:

# Handtrace

Testcase 1:

0x4 + 0x2 + 0 = 0

Do tổng vế trái luôn bằng 0 với mọi x nên ta kết luận:

Phương trình vô số nghiệm

Testcase 2:

0x4 + 0x2 + 1 = 0

Do tổng vế trái luôn khác 0 với mọi x nên ta kết luận:

Phương trình vô nghiệm

Testcase 3:

1x4 + 1x2 + 0 = 0

Suy ra:

Vậy ta có 2 trường hợp

Phương trình có 1 nghiệm x = 0

Testcase 4:

1x4 - 2x2 + 1 = 0

Suy ra:

Đặt

Ta có phương trình sau khi đặt t:

Tính (delta):

Phương trình có 2 nghiệm phân biệt

Hoặc

Dựa vào hằng đẳng thức số 2 ta có:

Vì , ta có:

Phương trình có 2 nghiệm = 1 và = -1

Testcase 5:

1x4 - 4x2 + 0 = 0

Suy ra:

Vậy ta có 2 trường hợp:

Phương trình có 3 nghiệm = 0, = -2 và = 2

Testcase 6:

1x4 - 5x2 + 4 = 0

Suy ra:

Tính (delta):

Vì , ta có các trường hợp:

Phương trình có 2 nghiệm phân biệt = 1, = -1, và = -2

# Mã giả

BEGIN

INPUT a, b, c // Nhập hệ số của phương trình

IF a = 0 THEN

IF b = 0 THEN

IF c = 0 THEN

sn = -1 // Phương trình có vô số nghiệm. Testcase 1

ELSE

sn = 0 // Phương trình vô nghiệm. Testcase 2

END IF

ELSE

CALL giaibac2(a, b, c) // Giải phương trình bậc 2: bx^2 + c = 0

sn = 2

x1 = -math.sqrt(-c / b)

x2 = math.sqrt(-c / b)

END IF

ELSE

// Giải phương trình trùng phương khi a ≠ 0

// Đặt t = x^2, phương trình trở thành at^2 + bt + c = 0

CALL giaibac2(a, b, c) // Giải phương trình bậc 2

CALCULATE delta = b^2 - 4ac

IF delta < 0 THEN

sn = 0 // Phương trình vô nghiệm thực

ELSE IF delta = 0 THEN

CALL giaibac2(a, b, c)

IF t1 < 0 THEN

sn = 0 // Phương trình vô nghiệm thực

ELSE IF t1 = 0 THEN

sn = 1 // Phương trình có 1 nghiệm. Testcase 3

x1 = 0

ELSE

sn = 2 // Phương trình có 2 nghiệm. Testcase 4

x1 = sqrt(t1)

x2 = -sqrt(t1)

END IF

ELSE

CALL giaibac2(a, b, c)

IF t1 >= 0 THEN

ADD sqrt(t1) và -sqrt(t1) vào tập nghiệm

END IF

IF t2 >= 0 THEN

// Testcase 5 xảy ra khi t2 > 0 và t1 = 0. Testcase 6 xảy ra khi cả t1 và t2 đều lớn hơn 0.

ADD sqrt(t2) và -sqrt(t2) vào tập nghiệm

END IF

REMOVE các nghiệm trùng lặp (chẳng hạn 0 và -0)

sn = số lượng nghiệm duy nhất trong tập nghiệm

IF sn = 1 THEN

PRINT nghiệm duy nhất

ELSE IF sn = 2 THEN

PRINT hai nghiệm

ELSE IF sn = 3 THEN

PRINT ba nghiệm

ELSE IF sn = 4 THEN

PRINT bốn nghiệm

END IF

END IF

END IF

END

# Python Implementation

# Lưu ý: Toàn bộ quá trình thực hiện được ghi lại bằng các hình anh thông qua thư mục “Quá trình cài đặt ” được đính kèm.

# Các hình cài đặt cho phương trình bậc 2 là từ 1-25.

# Các hình cài đặt cho phương trình trùng phương là từ A-Z1

# import math

# def giaibac2(a, b, c):

# if a == 0:

# if b == 0:

# return (-1, None, None) if c == 0 else (0, None, None)

# x1 = -c / b

# return (1, x1, None)

# 

# delta = b \* b - 4 \* a \* c

# if delta < 0:

# return (0, None, None)

# elif delta == 0:

# x1 = -b / (2 \* a)

# return (1, x1, None)

# else:

# x1 = (-b - math.sqrt(delta)) / (2 \* a)

# x2 = (-b + math.sqrt(delta)) / (2 \* a)

# return (2, min(x1, x2), max(x1, x2))

# def giaipttrungphuong(a, b, c):

# if a == 0:

# return (0, None, None, None, None) if b == 0 and c != 0 else giaibac2(a, b, c)

# 

# delta = b \* b - 4 \* a \* c

# if delta < 0:

# return (0, None, None, None, None)

# elif delta == 0:

# t1 = -b / (2 \* a)

# return (1, t1, None, None, None)

# 

# #delta > 0

# sqrt\_delta = math.sqrt(delta)

# t1 = (-b - sqrt\_delta) / (2 \* a)

# t2 = (-b + sqrt\_delta) / (2 \* a)

# 

# solutions = set() # Prevent duplicate 0

# if t1 >= 0:

# solutions.add(math.sqrt(t1))

# solutions.add(-math.sqrt(t1))

# if t2 >= 0:

# solutions.add(math.sqrt(t2))

# solutions.add(-math.sqrt(t2))

# 

# #Prevent -0

# solutions = sorted(solutions)

# 

# return (len(solutions), \*solutions) + (None,) \* (4 - len(solutions))

# if \_\_name\_\_ == "\_\_main\_\_":

# s = list(map(float, input().split()))

# a, b, c = s

# sn, \*roots = giaipttrungphuong(a, b, c)

# 

# if sn == -1:

# print("Phương trình vô số nghiệm")

# elif sn == 0:

# print("Phương trình vô nghiệm")

# elif sn == 1:

# print(f"Phương trình có 1 nghiệm: {roots[0]}")

# elif sn == 2:

# print(f"Phương trình có 2 nghiệm: {roots[0]} và {roots[1]}")

# elif sn == 3:

# print(f"Phương trình có 3 nghiệm: {roots[0]}, {roots[1]} và {roots[2]}")

# elif sn == 4:

# print(f"Phương trình có 4 nghiệm: {roots[0]}, {roots[1]}, {roots[2]} và {roots[3]}")

# 

# Testing

# test\_pttrungphuong.py

%%writefile pttrungphuong.in1

0 0 0

%%writefile pttrungphuong.in2

0 0 1

%%writefile pttrungphuong.in3

1 0 0

%%writefile pttrungphuong.in4

1 -2 1

%%writefile pttrungphuong.in5

1 -4 0

%%writefile pttrungphuong.in6

1 -5 4

!echo Test Case 1

!python pttrungphuong.py <pttrungphuong.in1

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 2

!python pttrungphuong.py <pttrungphuong.in2

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 3

!python pttrungphuong.py <pttrungphuong.in3

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 4

!python pttrungphuong.py <pttrungphuong.in4

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 5

!python pttrungphuong.py <pttrungphuong.in5

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

!echo Test Case 6

!python pttrungphuong.py <pttrungphuong.in6

!echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Testing output:**

Test Case 1

Phương trình vô số nghiệm

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 2

Phương trình vô nghiệm

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 3

Phương trình có 1 nghiệm: -0.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 4

Phương trình có 1 nghiệm: 1.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 5

Phương trình có 3 nghiệm: -2.0, 0.0 và 2.0

\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Case 6

Phương trình có 4 nghiệm: -2.0, -1.0, 1.0 và 2.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_