

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC
— o0o —



BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: KỸ THUẬT LẬP TRÌNH

Giảng viên hướng dẫn: TS. Nguyễn Thị Thanh Huyền

Sinh viên thực hiện: Vũ Hữu Sỹ

Mã số sinh viên: 20216958

Lớp: HTTTQL 01 - K66

Mã lớp học: 142298

Mục lục

Mở đầu	1
1 Trình bày bài toán giải gần đúng phương trình $f(x) = 0$ bằng phương pháp tiếp tuyến	2
1.1 Sự tồn tại nghiệm của phương trình	2
1.2 Khoảng phân ly nghiệm	2
1.3 Đặt vấn đề	3
1.4 Xây dựng công thức lặp của phương pháp tiếp tuyến	3
1.5 Sự hội tụ của phương pháp tiếp tuyến	4
1.6 Nhận xét về phương pháp tiếp tuyến	5
2 Thiết kế chương trình theo phương pháp tinh chỉnh dần từng bước	7
2.1 Bài toán	7
2.2 Yêu cầu	8
2.3 Quá trình thiết kế chương trình theo phương pháp tinh chỉnh dần từng bước	8
2.3.1 Chức năng 1	8
2.3.2 Chức năng 2	11
2.3.3 Chức năng 3	13
2.3.4 Chức năng 4	16
2.3.5 Chức năng 5	19
3 Mã nguồn chương trình	21
3.1 Thông tin về mã nguồn	22
3.2 Mã nguồn	25
4 Hình ảnh giao diện thực hiện chương trình và nội dung file text là kết quả ra của chương trình	60
4.1 Giao diện nhập vào vào đa thức	60
4.2 Giao diện MENU chính	61

4.3	Giao diện thực hiện chức năng 1: Tìm các khoảng phân ly nghiệm. .	62
4.4	Giao diện thực hiện chức năng 2: Rút gọn khoảng phân ly nghiệm tìm được ở chức năng 1	65
4.5	Giao diện thực hiện chức năng 3: Tìm nghiệm gần đúng với số lần lặp n cho trước	67
4.6	Giao diện thực hiện chức năng 4: Tìm nghiệm gần đúng với sai số e cho trước	70
4.7	Giao diện thực hiện chức năng 5: Tìm nghiệm gần đúng x_n thỏa mãn điều kiện $ x_n - x_{n-1} \leq \varepsilon$	74
4.8	Giao diện thực hiện chức năng 6: Hướng dẫn sử dụng MENU	77
4.9	Giao diện thực hiện chức năng 7: Giới thiệu về phương pháp tiếp tuyến	77
4.10	Giao diện thực hiện chức năng 8: Exit	78
5	Những tính năng và đồ họa của chương trình	79
5.1	Bắt phím người dùng nhập vào và xử lý phím đó để điều khiển MENU	79
5.2	Bắt lỗi nhập sai của người dùng và yêu cầu nhập lại	82
5.3	Lưu kết quả tính toán được vào file text	85
5.4	Số chữ số hiển thị ở phần thập phân	86
5.5	Đồ họa của chương trình	86
	Kết luận	88
	Tài liệu tham khảo	90

Mở đầu

Kỹ thuật lập trình - bức tranh phong phú của mã nguồn, nhịp điệu của logic, và sức sáng tạo không giới hạn. Đây là một môn học hấp dẫn, đưa ta vào một thế giới kỳ diệu của mã lệnh và thuật toán, nơi mà những ý tưởng trừu tượng chuyển hóa thành những ứng dụng thực tế đầy hữu ích.

Khi học, chúng ta không chỉ đơn thuần là việc học cú pháp và ngôn ngữ lập trình, mà còn là quá trình khám phá, tìm hiểu và giải quyết các vấn đề phức tạp bằng cách sử dụng sức mạnh của máy tính.

Đi sâu vào thế giới kỹ thuật lập trình, ta sẽ gặp nhiều ngôn ngữ lập trình phong phú và đa dạng như C/C++, Python, Java, JavaScript và nhiều ngôn ngữ khác nữa. Mỗi ngôn ngữ mang đến cái nhìn độc đáo về cách tiếp cận và giải quyết vấn đề, từ đó mở rộng tầm nhìn và cách suy nghĩ của chúng ta.

Nhưng môn học kỹ thuật lập trình không chỉ dừng lại ở việc học ngôn ngữ, nó còn thúc đẩy ta khám phá và áp dụng các khái niệm thiết yếu như cấu trúc dữ liệu, thuật toán, quản lý bộ nhớ,.. Từ những khái niệm này, chúng ta có thể xây dựng những ứng dụng phức tạp, từ phần mềm máy tính đơn giản đến hệ thống phân tán và các ứng dụng di động thú vị.

Và dưới bề mặt những dòng mã nguồn phức tạp, kỹ thuật lập trình còn chứa đựng sự thách thức và niềm vui của việc giải quyết những câu đố logic, tối ưu hóa mã nguồn, và cải tiến hiệu suất. Đó là cuộc hành trình không chỉ học hỏi từ sách vở mà còn bắt nguồn từ trí tưởng tượng và sự sáng tạo không giới hạn của mỗi người.

Em xin cảm ơn cô Nguyễn Thị Thanh Huyền vì những kiến thức quý báu mà cô đã giảng dạy. Những kiến thức đó đã giúp em tiến bộ hơn rất nhiều trong lập trình để rồi em đã có thể viết được một chương trình như em mong muốn. Trong quá trình thiết kế chương trình và làm báo cáo không thể tránh khỏi những thiếu sót, em mong cô sẽ có những lời nhận xét để chương trình và bài báo cáo của em được hoàn thiện hơn nữa ạ!

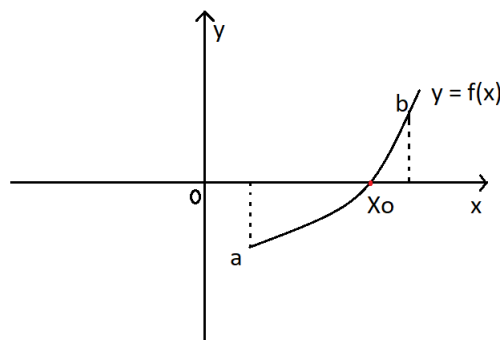
Chương 1

Trình bày bài toán giải gần đúng phương trình $f(x) = 0$ bằng phương pháp tiếp tuyến

1.1 Sự tồn tại nghiệm của phương trình

Định lý: Nếu hàm số $f(x)$ liên tục trên đoạn $[a, b]$ và $f(a)$ và $f(b)$ trái dấu, tức là $f(a) \cdot f(b) < 0$ thì phương trình $f(x) = 0$ có ít nhất một nghiệm trong khoảng (a, b) .

1.2 Khoảng phân ly nghiệm



Định nghĩa: Khoảng (a, b) được gọi là khoảng phân ly nghiệm của phương trình nếu nó chứa một và chỉ một nghiệm của phương trình đó.

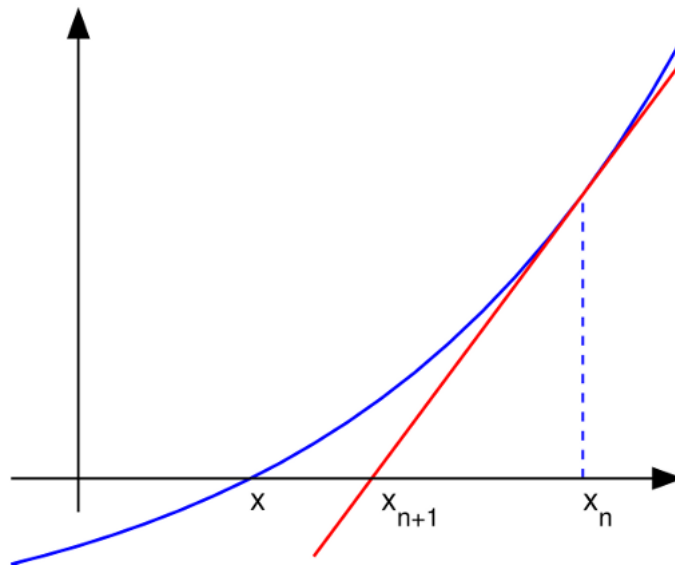
Định lý: Nếu hàm số $f(x)$ liên tục, đơn điệu trên đoạn (a, b) và $f(a) \cdot f(b) < 0$ thì đoạn $[a, b]$ được gọi là một khoảng phân ly nghiệm của phương trình $f(x) = 0$.

1.3 Đặt vấn đề

- Cho hàm số $f(x)$ xác định trên khoảng (a,b) . Với (a,b) là khoảng phân ly nghiệm. Tìm nghiệm gần đúng của phương trình?

- Ý tưởng của phương pháp:

- Thay thế đường cong $y = f(x)$ tại $[a,b]$ bằng tiếp tuyến.
- Tìm giao điểm của tiếp tuyến với trục hoành thay cho giao điểm của đường cong với trục hoành.



1.4 Xây dựng công thức lặp của phương pháp tiếp tuyến

- Xét phương trình $f(x) = 0$ và khoảng phân ly nghiệm (a,b)
- Gọi $M(x, f(x))$ là điểm bắt đầu nếu: $f(x).f''(x) > 0$
- Chọn điểm $x_0 \in (a, b)$ thỏa mãn điều kiện trên. Đặt $M_0(x_0, f(x_0))$
- Gọi d_k là tiếp tuyến của đồ thị tại M_k :

$$- d_0 \cap Ox \equiv (x_1, 0) \Rightarrow M_1(x_1, f(x_1))$$

$$- d_1 \cap Ox \equiv (x_2, 0) \Rightarrow M_2(x_2, f(x_2))$$

...

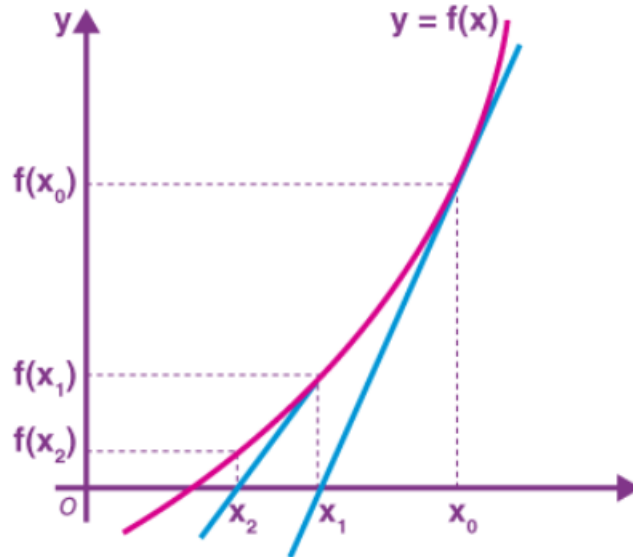
$$- d_{n-1} \cap Ox \equiv (x_n, 0) \Rightarrow x_n \approx x^*$$

- Phương trình tiếp tuyến:

$$d_k : y = f'(x_k)(x - x_k) + f(x_k) \quad (1)$$

- Vì $d_k \cap Ox \equiv (x_{k+1}, 0)$ nên ta có:

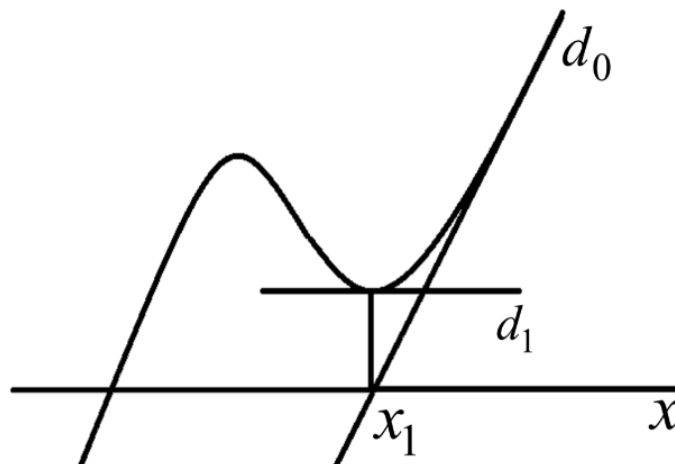
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2)$$



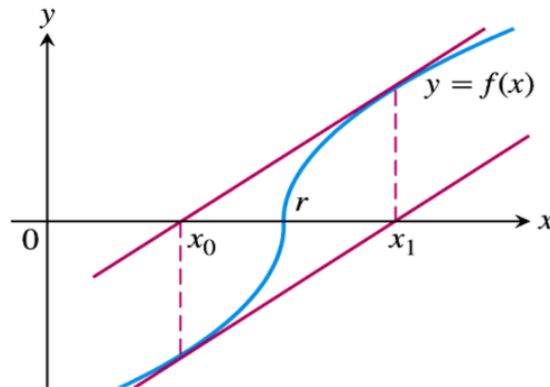
1.5 Sự hội tụ của phương pháp tiếp tuyến

Định lý: Điều kiện hội tụ theo Fourier (điều kiện đủ):

- Giả sử (a, b) là khoảng phân ly nghiệm của phương trình $y = f(x)$. Các đạo hàm cấp một: $f'(x)$ và đạo hàm cấp hai: $f''(x)$ liên tục, không đổi dấu, không tiêu diệt trên (a, b) . Khi đó ta chọn xấp xỉ nghiệm ban đầu x_0 thuộc a, b sao cho $f(x_0) \cdot f''(x_0) > 0$ thì quá trình lặp sẽ hội tụ đến nghiệm.
- Nếu $f'(x_1) = 0$ với $x_1 \in (a, b)$
 $\Rightarrow d_1 \cap Ox = \emptyset$. Chuỗi x_n bị dừng ở x_1 và không hội tụ đến nghiệm.



- Nếu $f''(x_1) = 0$ với $x_1 \in (a, b)$ thì hàm số vừa lồi vừa lõm.
- Đặc biệt: $f(x_1) = 0$ và $f''(x_1) = 0$



- Hàm số thỏa mãn điều kiện định lý trên thì dãy lặp (2) hội tụ đến nghiệm đúng của phương trình theo đánh giá sau:

1. Công thức sai số mục tiêu: $|x_n - x^*| \leq \frac{|f(x_n)|}{m_1}$ (3)

2. Công thức sai số hai xấp xỉ liên tiếp: $|x_n - x^*| \leq \frac{M_2}{2m_1} |x_n - x_{n-1}|^2$ (4)

Trong đó:

$$m_1 = \min_{x \in [a, b]} |f'(x)|$$

$$M_2 = \max_{x \in [a, b]} |f''(x)|$$

1.6 Nhận xét về phương pháp tiếp tuyến

1. Hội tụ nhanh: Phương pháp tiếp tuyến thường hội tụ với tốc độ nhanh và xấp xỉ nghiệm chính xác đến mức mong muốn với số lượng lặp ít.
2. Áp dụng rộng rãi: Phương pháp này có thể áp dụng cho nhiều loại hàm số phi tuyến và các vấn đề khác nhau trong các lĩnh vực như toán học, khoa học máy tính, kỹ thuật và kinh tế.
3. Đơn giản và hiệu quả: Phương pháp tiếp tuyến dễ hiểu và thực hiện, không yêu cầu các giả định phức tạp và không đòi hỏi quá nhiều kỹ thuật toán.
4. Nhược điểm đáng lưu ý: Một số hạn chế của phương pháp tiếp tuyến bao gồm:
 - Đòi hỏi kiến thức về đạo hàm: Phương pháp này yêu cầu tính toán đạo hàm của hàm số, và đôi khi tính toán đạo hàm có thể phức tạp hoặc không khả thi.

- Phụ thuộc vào điểm bắt đầu: Phương pháp có thể không hội tụ hoặc hội tụ chậm nếu điểm bắt đầu ban đầu lựa chọn không tốt.
- Dễ rơi vào điểm cực tiểu hoặc điểm cực đại: Phương pháp có thể dẫn đến nghiệm xấp xỉ không chính xác hoặc không hội tụ đúng nghiệm đúng khi hàm có điểm cực tiểu hoặc điểm cực đại.

Như vậy, phương pháp tiếp tuyến là một công cụ hữu ích trong việc tìm nghiệm gần đúng của các hàm phi tuyến. Tuy nhiên, như với bất kỳ phương pháp nào, nó cần được áp dụng một cách cân nhắc và kiểm tra tính hợp lệ và sự hội tụ của nghiệm.

Chương 2

Thiết kế chương trình theo phương pháp tinh chỉnh dần từng bước

2.1 Bài toán

Giải gần đúng phương trình $f(x) = 0$ ($f(x)$ là đa thức) bằng phương pháp tiếp tuyến. Thực hiện các yêu cầu:

1. Tìm các khoảng phân ly nghiệm của phương trình
2. Tìm khoảng phân ly nghiệm (a, b) của phương trình thoả mãn $|a - b| \leq 0,5$ bằng cách sử dụng phương pháp chia đôi để thu hẹp dần một khoảng phân ly nghiệm đã tìm được ở ý 1
3. Tìm nghiệm gần đúng với số lần lặp n cho trước trong khoảng phân ly nghiệm (a, b) và đánh giá sai số theo cả hai công thức (n được nhập vào từ bàn phím, (a, b) được nhập vào từ bàn phím)
4. Tìm nghiệm gần đúng trong khoảng (a, b) với sai số ε cho trước (ε được nhập vào từ bàn phím, (a, b) được nhập vào từ bàn phím). Tính toán theo 2 cách áp dụng công thức sai số.
5. Tìm nghiệm gần đúng x_n trong khoảng (a, b) thoả mãn điều kiện: $|x_n - x_{n-1}| \leq \varepsilon$ (với ε được nhập vào từ bàn phím).

2.2 Yêu cầu

- Mọi kết quả được hiển thị với số chữ số phần thập phân nhập vào từ bàn phím
- Ghi vào tệp văn bản thể hiện quá trình thực hiện chương trình và các kết quả ra
- Thực hiện chương trình bằng MENU điều khiển bởi các phím chức năng. SV tự code để thiết lập và điều khiển MENU

2.3 Quá trình thiết kế chương trình theo phương pháp tinh chỉnh dần từng bước

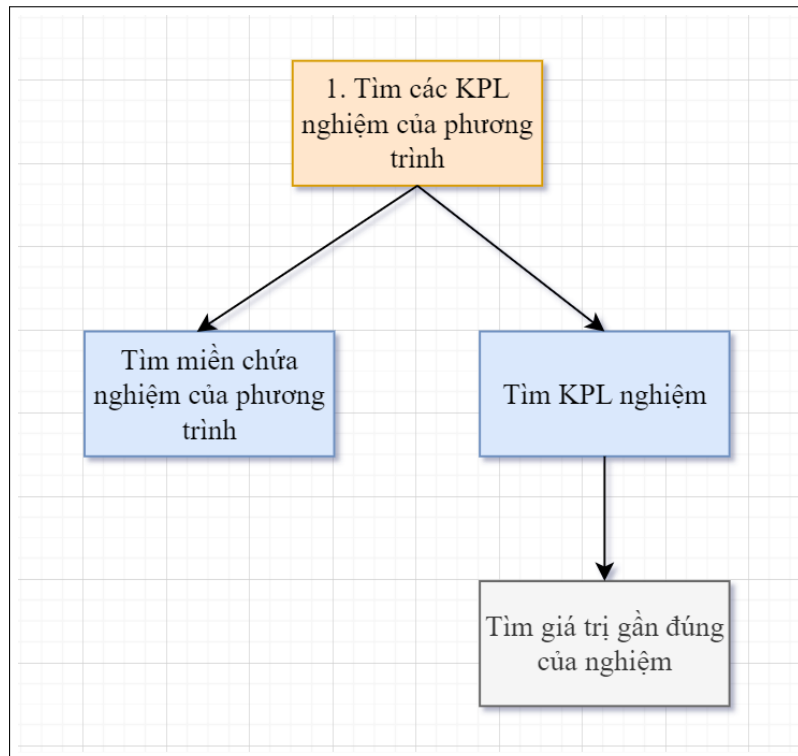
Từ đề bài, em đã thiết kế một chương trình lớn gồm có 5 chức năng để thực hiện các yêu cầu. Mỗi chức năng trong chương trình này là một bài toán nhỏ, được xây dựng một cách chặt chẽ và có liên kết với nhau. Ví dụ: Nếu ta không thực hiện chức năng 1 thì sẽ không có khoảng phân ly để chương trình thực hiện rút gọn ở chức năng 2, khi đó chương trình sẽ báo lỗi.

Mỗi chức năng sẽ gồm nhiều hàm thực hiện các tác vụ nhỏ của chức năng đó và các hàm của chức năng này có thể được tái sử dụng ở chức năng khác. Quá trình thiết kế theo phương pháp tinh chỉnh dần từng bước được em tham khảo trong slide của cô và mô tả chi tiết như sau.

2.3.1 Chức năng 1

Bài toán 1: Tìm các khoảng phân ly nghiệm của phương trình.

- **Bước 0:**



• **Bước 1:**

- Dữ liệu vào: Phương trình $f(x) = 0$, các hệ số của phương trình, bậc của phương trình đa thức, miền chứa nghiệm của phương trình.
- Dữ liệu ra: Các khoảng phân ly nghiệm của phương trình $f(x) = 0$.
- Ý tưởng thuật toán:
 - Tìm miền chứa nghiệm của phương trình đa thức trong khoảng từ `lower_bround` đến `upper_bround` bằng cách sử dụng phương pháp gần đúng (approximation method).
 - Sau khi thực hiện xong thuật toán, ta tìm được các giá trị gần đúng của nghiệm trong miền từ `lower_bround` đến `upper_bround`. Tuy nhiên, giá trị gần đúng này không chắc chắn là chính xác và không thể thể hiện được tất cả các nghiệm của phương trình đa thức.
 - Để đảm bảo rằng các khoảng phân ly nghiệm được xác định chính xác, chúng ta cần kiểm tra điều kiện $\text{value1} * \text{value2} < 0$. Điều này đảm bảo rằng giữa các giá trị gần đúng, tồn tại ít nhất một nghiệm. Nếu tích của hai giá trị này nhân với nhau và cho kết quả âm, có nghĩa là giữa hai giá trị đó, nằm ít nhất một nghiệm của phương trình.

- **Bước 2:** Hàm tìm miền chứa nghiệm.

- Fuction domain_solution:

- Vào: Phương trình $f(x)$, bậc của phương trình đa thức, hệ số của phương trình.
- Ra: **lower__bround:** cận dưới của miền chứa nghiệm và **upper__bround:** cận trên của miền chứa nghiệm.
- Ý tưởng:
 1. Khởi tạo cận dưới và cận trên ban đầu của miền chứa nghiệm là -1 và 1.
 2. Thực hiện một vòng lặp để tìm giá trị tuyệt đối lớn nhất của các hệ số âm trong mảng chứa các hệ số của đa thức.
 3. Nếu giá trị tuyệt đối lớn nhất là 0, kiểm tra và cập nhật cận dưới hoặc cận trên.
 4. Nếu giá trị tuyệt đối lớn nhất khác 0, tính toán cận dưới và cận trên dựa trên giá trị tuyệt đối lớn nhất, hệ số đầu tiên và chỉ số của hệ số âm đầu tiên.
 5. Đảo dấu các hệ số có chỉ số lẻ trong mảng tạm thời.
 6. Lặp lại các bước trên cho đến khi cận dưới không còn lớn hơn 0.
 7. Kết quả cuối cùng là cận dưới và cận trên của miền chứa nghiệm.

- **Bước 3:** Tìm các khoảng phân ly nghiệm.

- Fuction: Kiểm tra xem a có phải là điểm Fourier không, nếu không thì b là điểm Fourier.

- Vào: Miền chứa nghiệm lower__bround, upper__bround.
- Ra: Các khoảng phân ly nghiệm nếu có
- Ý tưởng:
 - + Xử lý các trường hợp đặc biệt: bậc của đa thức bằng 0, bằng 1
 - + Thực hiện thuật toán để tìm giá trị gần đúng của các nghiệm trong miền chứa nghiệm:
 1. Khởi tạo x_1 với giá trị của lower__bround.
 2. Sử dụng một vòng lặp để tính toán các giá trị gần đúng x_1 cho từng nghiệm.

3. Trong vòng lặp, tính toán giá trị của x_0 , temp0, sign, x_1 , temp1, eta để tiếp tục điều chỉnh bước tiếp cận và xác định giá trị gần đúng của nghiệm.
4. Nếu giá trị tuyệt đối của đạo hàm tại x_1 nhỏ hơn $1e-4$, ghi lại giá trị gần đúng vào mảng survey_value và tăng biến k lên 1.
5. Cuối cùng, gán lower_bround vào survey_value[0], uupper_bround vào survey_value[k] và cập nhật số lượng khoảng phân ly nghiệm distance_count.

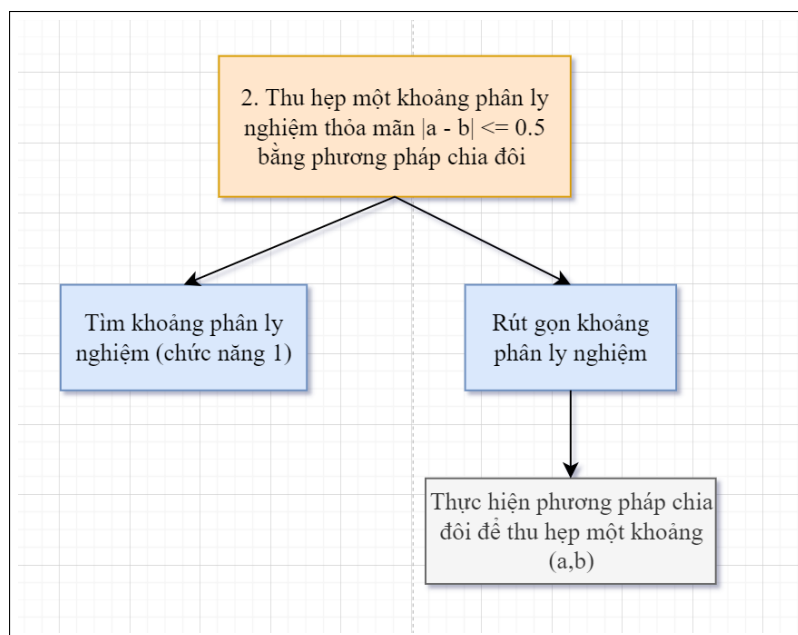
+ Duyệt qua các giá trị trong mảng survey_value để xác định và in ra các khoảng phân ly nghiệm:

1. Tính giá trị của hàm tại các điểm survey_value[i] và survey_value[i + 1].
2. Nếu tích của hai giá trị này nhân với nhau nhỏ hơn 0, in ra thông tin về khoảng phân ly nghiệm tương ứng.

2.3.2 Chức năng 2

Bài toán 2: Tìm khoảng phân ly nghiệm (a,b) của phương trình thỏa mãn $|a - b| \leq 0.5$ bằng cách sử dụng phương pháp chia đôi để thu hẹp dần một khoảng phân ly nghiệm đã tìm được ở ý 1.

• **Bước 0:**



- **Bước 1:**

- Dữ liệu vào: Khoảng phân ly nghiệm (a,b).
- Dữ liệu ra: Khoảng phân ly nghiệm được rút gọn thỏa mãn $|a - b| \leq 0.5$.
- Ý tưởng:
 - Tìm một khoảng phân ly nghiệm (a,b) của phương trình đa thức.
 - Sử dụng vòng lặp while để tiến hành chia đôi cho đến khi khoảng cách giữa a và b nhỏ hơn hoặc bằng 0.5.
 - Tính giá trị trung bình **temp** của a và b. Tính giá trị của hàm tại a và temp và kiểm tra dấu của tích hai giá trị này. Nếu tích dương, gán temp vào a, ngược lại gán temp vào b. Tăng biến lan_lap lên 1 sau mỗi vòng lặp.

- **Bước 2:** Tìm một khoảng phân ly nghiệm của phương trình.

Phần này đã được mô tả các bước thực hiện ở **Chức năng 1**.

- **Bước 3:** Rút gọn khoảng phân ly nghiệm tìm được.

- Vào: Khoảng phân ly nghiệm (a,b).
- Ra: Khoảng phân ly nghiệm được rút gọn thỏa mãn $|a - b| \leq 0.5$.
- Ý tưởng: Thực hiện lặp theo phương pháp chia đôi, rút gọn dần đến khi thỏa mãn yêu cầu đề bài.

- **Bước 4:** Thực hiện phương pháp chia đôi

- Vào: Khoảng (a,b).
- Ra: Khoảng (a,b) được rút gọn theo điều kiện.
- Ý tưởng:
 1. Vòng lặp while sẽ thực hiện cho đến khi khoảng cách giữa cận trên b và cận dưới a thỏa mãn điều kiện đề bài.
 2. Trong mỗi vòng lặp:
 - * Tính giá trị trung bình của a và b bằng cách gán $(a + b) / 2.0$ vào biến temp.

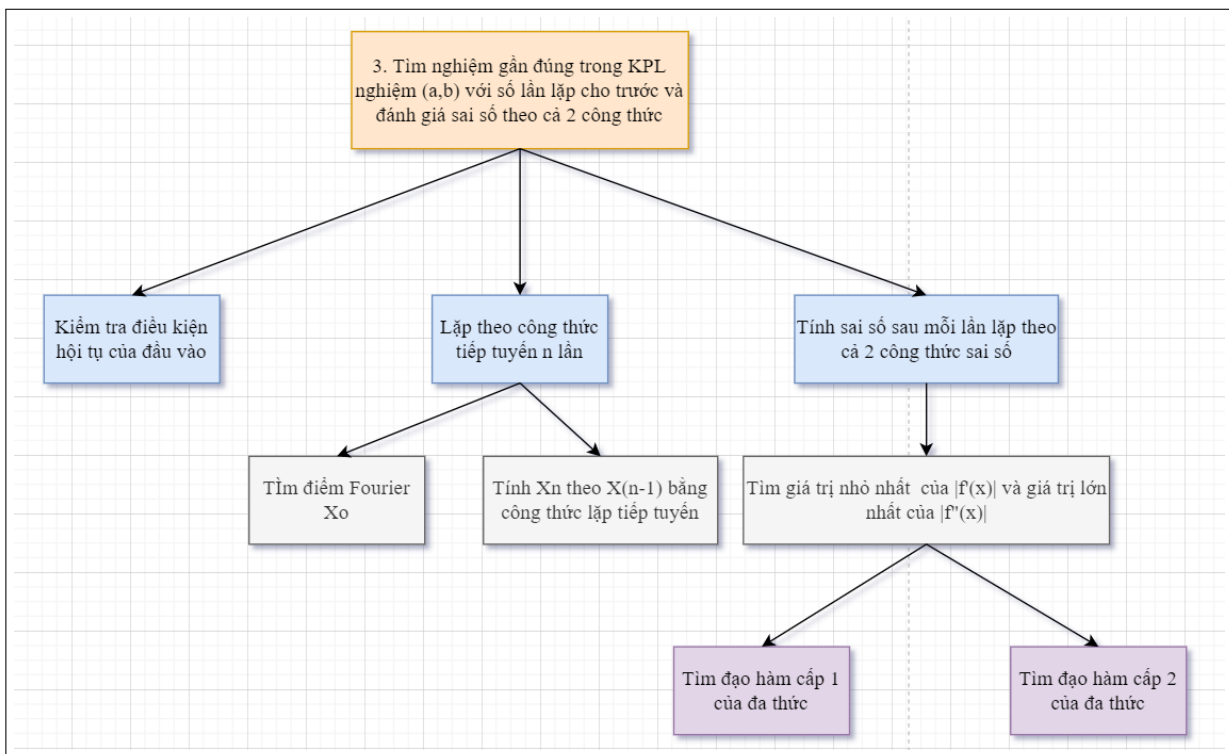
- * Kiểm tra dấu của tích giữa giá trị hàm tại a và temp. Nếu tích này nhỏ hơn 0 (có dấu khác nhau), tức là nghiệm nằm giữa a và temp, ta gán temp vào b để thu gọn khoảng về phía cận trên. ngược lại gán temp vào a.

- * Tăng biến lan_lap lên 1 để đếm số lần lặp đã thực hiện.

3. Khi kết thúc vòng lặp, khoảng (a,b) đã được thu gọn theo điều kiện.

2.3.3 Chức năng 3

• Bước 0:



• Bước 1:

- Dữ liệu vào: Phương trình $f(x)$, khoảng phân ly nghiệm (a,b), số lần lặp n.
- Dữ liệu ra: Nghiệm gần đúng trong KPL nghiệm (a,b) sau n lần lặp bằng phương pháp tiếp tuyến và đánh giá sai số theo 2 công thức sai số.

- Ý tưởng:

- Dùng công thức lặp của phương pháp tiếp tuyến n lần để tìm nghiệm gần đúng và lưu kết quả mỗi lần lặp vào file.
- Hiện thị kết quả của lần lặp thứ n và sai số ra màn hình.

• Bước 2: Hàm kiểm tra điều kiện hội tụ của phương pháp tiếp tuyến

- Fuction check_input:

- Vào: Cận dưới a, cận trên b của khoảng phân ly nghiệm
- Ra: **True** nếu KPL nghiệm hợp lệ, **False** nếu ngược lại.
- Thuật toán:
 - if (a == b) then Return False
 - else if dấu của $f(a)$ và $f(b)$ giống nhau then Return False
 - else if (a > b) then Return False
 - else if dấu của $f'(a)$ và $f'(b)$ khác nhau then Return False
 - else if dấu của $f''(a)$ và $f''(b)$ khác nhau then Return False
 - else Return True;

• **Bước 3:** Tìm điểm Fourier

- Fuction fourier_point: Kiểm tra xem a có phải là điểm Fourier không, nếu không thì b là điểm Fourier.

- Vào: Cận dưới a
- Ra: **True** nếu a là điểm Fourier, **False** nếu ngược lại.
- Thuật toán:
 - if dấu của $f(a).f''(a)$ lớn hơn 0 then Return True
 - else Return False;

• **Bước 4:** Tính x_n theo công thức lặp của phương pháp tiếp tuyến

- Vào: $x_{n-1}, f(x_{n-1}), f'(x_{n-1})$
- Ra: x_n
- Tính $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ và trả về kết quả.

• **Bước 5:** Tính sai số ở mỗi lần lặp bằng cả hai công thức.

Function saiso:

- Vào: $x_n, x_{n-1}, m1, M2$
- Ra: delta1 và delta2 lần lượt là giá trị của hai công thức sai số mục tiêu và sai số hai xấp xỉ liên tiếp.
- Trả về $\text{delta1} = \frac{|f(x_n)|}{m_1}$ và $\text{delta2} = \frac{M_2}{2m_1} |x_n - x_{n-1}|^2$

- **Bước 6:** Tính giá trị nhỏ nhất của $|f'(x)|$ trong KPL nghiệm (a,b).

Function Min:

- Vào: $|f'(x)|$, a, b
- Ra: Giá trị nhỏ nhất của $|f'(x)|$ trong khoảng (a,b)
- Ý tưởng: Chia khoảng (a,b) thành 10000 phần bằng nhau và kiểm tra giá trị của hàm tại từng điểm, sau đó trả về giá trị nhỏ nhất tìm được.

$x_0 := a$

$\min := a$

$\alpha := (b - a) / 10000$

for i in range(10000):

if $f'(x_0) < f'(\min)$:

$\min := x_0$

$x_0 := x_0 + \alpha$

Return $f'(\min)$;

- **Bước 7:** Tính giá trị lớn nhất của $|f''(x)|$ trong KPL nghiệm (a,b).

Function Max:

- Vào: $|f''(x)|$, a, b
- Ra: Giá trị lớn nhất của $|f''(x)|$ trong khoảng (a,b)
- Ý tưởng: Chia khoảng (a,b) thành 10000 phần bằng nhau và kiểm tra giá trị của hàm tại từng điểm, sau đó trả về giá trị lớn nhất tìm được.

$x_0 := a$

$\max := a$

$\alpha := (b - a) / 10000$

for i in range(10000):

if $f''(x_0) > f''(\max)$:

$\max := x_0$

$x_0 := x_0 + \alpha$

Return $f''(\max)$;

- **Bước 8:** Tính đạo hàm cấp 1 và đạo hàm cấp 2 tại x

Function df:

- Vào: $x, f(x)$
- Ra: xấp xỉ giá trị đạo hàm cấp 1 tại x
- Ý tưởng: Sử dụng công thức xấp xỉ đạo hàm đối xứng để tính toán đạo hàm gần đúng.

Gán $h := 1e-7$

Return $((f(x + h) - f(x - h)) / (2 * h));$

Function ddf:

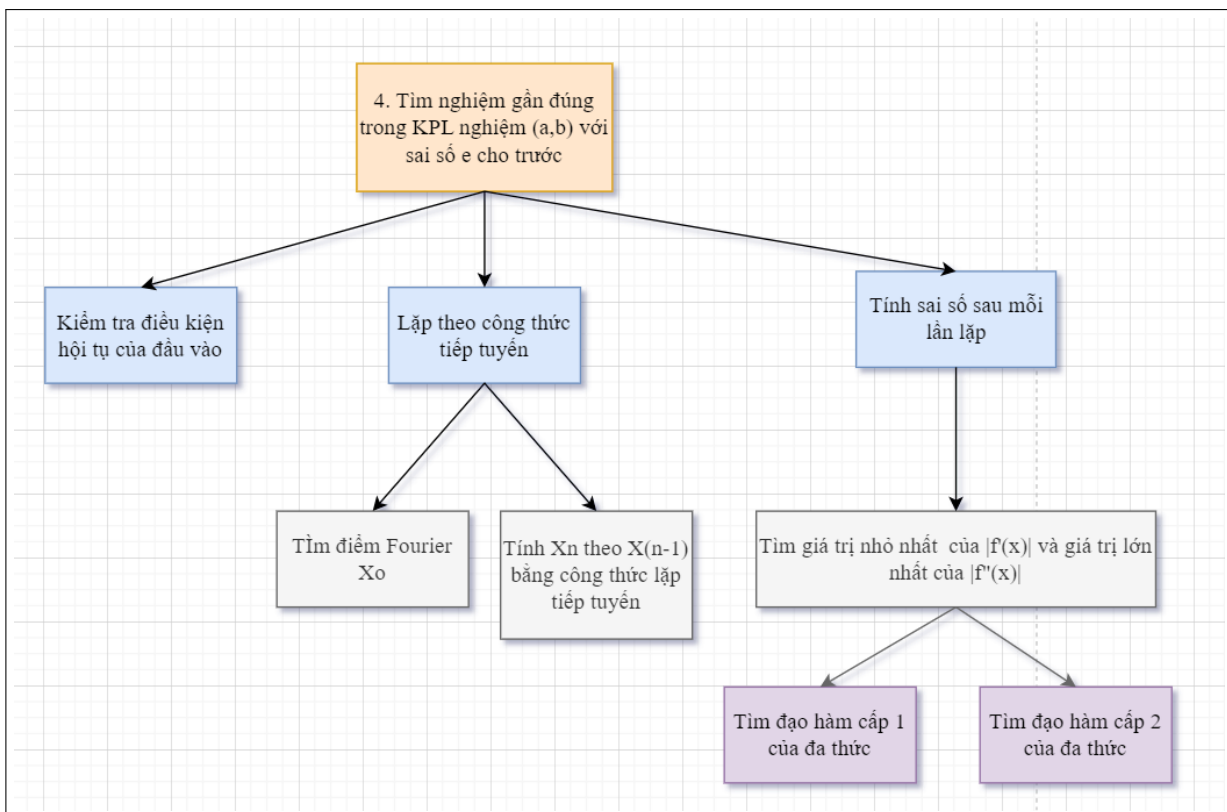
- Vào: $x, f'(x)$
- Ra: xấp xỉ giá trị đạo hàm cấp 2 tại x
- Ý tưởng: Sử dụng công thức xấp xỉ đạo hàm đối xứng để tính toán đạo hàm gần đúng.

Gán $h := 1e-7$

Return $((f'(x + h) - f'(x - h)) / (2 * h));$

2.3.4 Chức năng 4

• Bước 0:



• Bước 1:

- Dữ liệu vào: Khoảng phân ly nghiệm (a,b) , sai số ε , công thức sai số.
- Dữ liệu ra: Nghiệm gần đúng x_n thỏa mãn sai số ε .
- Ý tưởng:
 - Lập tìm nghiệm gần đúng bằng phương pháp tiếp tuyến cho đến khi thỏa mãn điều kiện sai số ε .
 - Hiển thị kết quả nghiệm thỏa mãn và sai số ra màn hình.
- **Bước 2:** Hàm kiểm tra điều kiện hội tụ của phương pháp tiếp tuyến
 - Fuction `check_input`:
 - Vào: Cận dưới a , cận trên b của khoảng phân ly nghiệm
 - Ra: **True** nếu KPL nghiệm hợp lệ, **False** nếu ngược lại.
 - Thuật toán:


```
if (a == b) then Return False
else if dấu của  $f(a)$  và  $f(b)$  giống nhau then Return False
else if ( $a > b$ ) then Return False
else if dấu của  $f'(a)$  và  $f'(b)$  khác nhau then Return False
else if dấu của  $f''(a)$  và  $f''(b)$  khác nhau then Return False
else Return True;
```
- **Bước 3:** Tìm điểm Fourier
 - Fuction `fourier_point`: Kiểm tra xem a có phải là điểm Fourier không, nếu không thì b là điểm Fourier.
 - Vào: Cận dưới a
 - Ra: **True** nếu a là điểm Fourier, **False** nếu ngược lại.
 - Thuật toán:


```
if dấu của  $f(a).f''(a)$  lớn hơn 0 then Return True
else Return False;
```
- **Bước 4:** Tính x_n theo công thức lặp của phương pháp tiếp tuyến
 - Vào: $x_{n-1}, f(x_{n-1}), f'(x_{n-1})$
 - Ra: x_n

– Tính $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ và trả về kết quả.

- **Bước 5:** Tính sai số ở mỗi lần lặp bằng cả hai công thức.

Function saiso:

– Vào: $x_n, x_{n-1}, m1, M2$

– Ra: delta1 và delta2 lần lượt là giá trị của hai công thức sai số mục tiêu và sai số hai xấp xỉ liên tiếp.

– Trả về $\text{delta1} = \frac{|f(x_n)|}{m_1}$ và $\text{delta2} = \frac{M_2}{2m_1}|x_n - x_{n-1}|^2$

- **Bước 6:** Tính giá trị nhỏ nhất của $|f'(x)|$ trong KPL nghiệm (a,b).

Function Min:

– Vào: $|f'(x)|, a, b$

– Ra: Giá trị nhỏ nhất của $|f'(x)|$ trong khoảng (a,b)

– Ý tưởng: Chia khoảng (a,b) thành 10000 phần bằng nhau và kiểm tra giá trị của hàm tại từng điểm, sau đó trả về giá trị nhỏ nhất tìm được.

$x_0 := a$

$\text{min} := a$

$\text{alpha} := (b - a) / 10000$

for i in range(10000):

if $f'(x_0) < f'(\text{min})$:

$\text{min} := x_0$

$x_0 := x_0 + \text{alpha}$

Return $f'(\text{min})$;

- **Bước 7:** Tính giá trị lớn nhất của $|f''(x)|$ trong KPL nghiệm (a,b).

Function Max:

– Vào: $|f''(x)|, a, b$

– Ra: Giá trị lớn nhất của $|f''(x)|$ trong khoảng (a,b)

– Ý tưởng: Chia khoảng (a,b) thành 10000 phần bằng nhau và kiểm tra giá trị của hàm tại từng điểm, sau đó trả về giá trị lớn nhất tìm được.

$x_0 := a$

$\text{max} := a$

```

alpha := (b - a) / 10000
for i in range(10000):
    if  $f''(x_0) > f''(max)$ :
        max :=  $x_0$ 
     $x_0 := x_0 + \alpha$ 
Return  $f''(max)$ ;

```

- **Bước 8:** Tính đạo hàm cấp 1 và đạo hàm cấp 2 tại x

Function df:

```

- Vào: x,  $f(x)$ 
- Ra: xấp xỉ giá trị đạo hàm cấp 1 tại x
- Ý tưởng: Sử dụng công thức xấp xỉ đạo hàm đối xứng để tính toán đạo hàm gần đúng.
Gán h := 1e-7
Return  $((f(x + h) - f(x - h)) / (2 * h))$ ;

```

Function ddf:

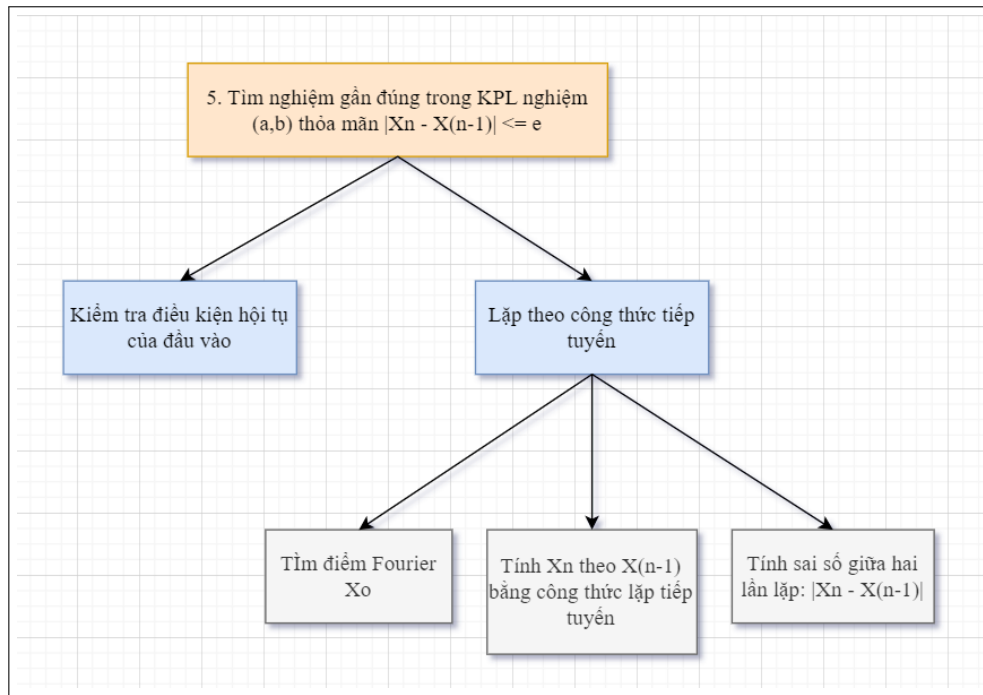
```

- Vào: x,  $f'(x)$ 
- Ra: xấp xỉ giá trị đạo hàm cấp 2 tại x
- Ý tưởng: Sử dụng công thức xấp xỉ đạo hàm đối xứng để tính toán đạo hàm gần đúng.
Gán h := 1e-7
Return  $((f'(x + h) - f'(x - h)) / (2 * h))$ ;

```

2.3.5 Chức năng 5

- **Bước 0:**



- Về cơ bản các bước để xây dựng chương trình cho chức năng này tương tự như chức năng 3 và 4. Các hàm đã được giới thiệu ở phần trước cũng được dùng cho chức năng 5. Chỉ cần thêm phần kiểm tra: $|x_n - x_{n-1}| \leq \epsilon$.

Chương 3

Mã nguồn chương trình

Từ các bước thiết kế chương trình theo phương pháp tinh chỉnh dần từng bước ở phần trước, em đã viết một chương trình hoàn chỉnh giải gần đúng phương trình $f(x) = 0$ ($f(x)$ là đa thức) bằng phương pháp tiếp tuyến. Chương trình có thể thực hiện các chức năng sau đây:

1. Tìm các khoảng phân ly nghiệm của phương trình.
2. Tìm khoảng phân ly nghiệm của phương trình thỏa mãn $|a - b| \leq 0,5$ bằng cách sử dụng phương pháp chia đôi để thu hẹp dần một khoảng phân ly nghiệm đã tìm được ở ý 1.
3. Tìm nghiệm gần đúng với số lần lặp n cho trước trong khoảng phân ly nghiệm (a, b) và đánh giá sai số theo cả hai công thức (n được nhập vào từ bàn phím, (a, b) được nhập vào từ bàn phím).
4. Tìm nghiệm gần đúng trong khoảng (a, b) với sai số ε cho trước (ε được nhập vào từ bàn phím, (a, b) được nhập vào từ bàn phím). Tính toán theo 2 cách áp dụng công thức sai số.
5. Tìm nghiệm gần đúng x_n trong khoảng (a, b) thỏa mãn điều kiện: $|x_n - x_{n-1}| \leq \varepsilon$ (với ε được nhập vào từ bàn phím).

Và chương trình có thể lưu kết quả tính toán vào file text, phần này em sẽ trình bày ở chương sau.

3.1 Thông tin về mã nguồn

- Các thư viện và Header File được sử dụng trong chương trình:

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <windows.h>
4 #include <math.h>
5 #include <stdbool.h>
6 #include "giao_dien.h"
```

- Chương trình có tổng cộng 31 hàm (không tính hàm main). Trong đó có 20 hàm **void**, 9 hàm trả về kiểu **double**, 2 hàm trả về kiểu **boolean**.

1. Các hàm trong File BTL.cpp:

```
1 void input();
2 void print_poly_to_file(FILE *file);
3 bool check_input(double a, double b);
4 void domain_solution();
5 double f(double x);
6 double df(double x);
7 double ddf(double x);
8 double f1(double x);
9 double f2(double x);
10 double min(double f(double x), double a, double b);
11 double max(double f(double x), double a, double b);
12 double sign(double x);
13 bool fourier_point(double a);
14 double sai_so(double x, double x_old, double m1, double M2, int
    ↪ ct_saiso);
15 void chucnang_1();
16 void chucnang_2();
17 void chucnang_3();
18 void chucnang_4();
19 void chucnang_5();
20 void menu();
```

2. Các hàm trong Header File "giao_dien.h":

```

1 void output();
2 void TextColor(int x);
3 void gotoxy(int x, int y);
4 void arrowHere(int realPosition, int arrowPosition);
5 void khungngoai();
6 void khungtrong();
7 void khungketqua();
8 void khunghuongdan();
9 void khungtieptuyen();
10 void khungexit();
11 void khung();

```

- Các hằng số được khai báo sử dụng **#define**:

```

1 #define MIN 1
2 #define MAX 8
3 #define N 100
4 #define max_loop 1000

```

1. MAX: là hằng chỉ số chức năng trong chương trình là 8.
2. MIN: là hằng chỉ chức năng bé nhất có số thứ tự là 1.
3. N: là hằng chỉ số phần tử tối đa trong mảng.
4. max_loop: là hằng chỉ số vòng lặp tối đa trong thuật toán tìm nghiệm.

- Các biến được khai báo toàn cục để sử dụng cho cả chương trình là:

```

1 const char* option[] =
2 {
3     "Tim cac KPL nghiem cua phuong trinh",
4     "Thu hep khoang phan ly nghiem",
5     "Tim nghiem gan dung voi so lan lap n",
6     "Tim nghiem gan dung voi sai so e",
7     "Tim Xn thoa man |Xn - X(n-1)| <= e",
8     "Huong dan dieu khien MENU",
9     "Gioi thieu ve phuong phap tiep tuyen",
10    "Exit"

```

```
11 };  
12  
13 int degree;  
14 int decimal_digit;  
15 int distance_count;  
16 double lower_bround;  
17 double upper_bround;  
18 double coeff_poly[N];  
19 double survey_value[N];  
20 double distance_value[N];
```

Trong đó

1. `const char* option[]`: là một mảng các con trỏ tới kiểu dữ liệu `char`, và các con trỏ này được khai báo là con trỏ tới kiểu dữ liệu hằng. Mảng này lưu trữ tên các chức năng trong chương trình.
2. `degree`: là một biến kiểu `int` dùng để lưu trữ bậc của đa thức.
3. `coeff_poly[N]`: là một mảng kiểu `double` được sử dụng để lưu trữ các hệ số của đa thức.
4. `decimal_digit`: là một biến kiểu `int` dùng để lưu trữ số chữ số thập phân hiển thị trong kết quả.
5. `lower_bround`: là một biến kiểu `double` dùng để lưu trữ giá trị của giới hạn dưới trong miền chứa nghiệm.
6. `upper_bround`: là một biến kiểu `double` dùng để lưu trữ giá trị của giới hạn trên trong miền chứa nghiệm.
7. `survey_value[N]`: là một mảng kiểu `double` dùng để lưu trữ các giá trị gần đúng của nghiệm.
8. `distance_value[N]`: là một mảng kiểu `double` được sử dụng để lưu trữ lần lượt các cận trên và cận dưới của các khoảng phân ly nghiệm.
9. `distance_count`: là một biến kiểu `int` dùng để lưu trữ số lượng khoảng phân ly nghiệm mà phương trình có.

3.2 Mã nguồn

Header File: giao_dien.h

```
1 void output();
2 void TextColor(int x);
3 void gotoxy(int x, int y);
4 void arrowHere(int realPosition, int arrowPosition);
5 void khungngoai();
6 void khungtrong();
7 void khungketqua();
8 void khunghuongdan();
9 void khungtieptuyen();
10 void khungexit();
11 void khung();
12
13 // ham doi mau chu
14 void TextColor(int x)
15 {
16     HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
17     SetConsoleTextAttribute(h, x);
18 }
19
20 // ham di chuyen vi tri con tro
21 void gotoxy(int x, int y)
22 {
23     COORD c = { x, y };
24     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
25 }
26
27 // ham lam noi bat chuc nang dang chon
28 void arrowHere(int realPosition, int arrowPosition)
29 {
30     if (realPosition == arrowPosition)
31     {
32         TextColor(207); printf("%c ", 175);
33     }
```

```
34     else {
35         TextColor(113); printf("    ");
36     }
37 }
38
39 // ham in ra thông tin
40 void khungthongtin()
41 {
42     TextColor(7);
43     for (int i = 82; i <= 118; i++)
44     {
45         gotoxy(i, 0); printf("%c", 205);
46         gotoxy(i, 10); printf("%c", 205);
47     }
48     for (int i = 0; i <= 10; i++)
49     {
50         gotoxy(82, i); printf("%c", 186);
51         gotoxy(118, i); printf("%c", 186);
52     }
53     gotoxy(82,0); printf("%c", 201);
54     gotoxy(118,0); printf("%c", 187);
55     gotoxy(82,10); printf("%c", 200);
56     gotoxy(118,10); printf("%c", 188);
57     gotoxy(95,0); TextColor(111); printf(" Thông tin ");
58     gotoxy(102,29); TextColor(8); printf("Made by Vu Huu Sy");
59     TextColor(14);
60     gotoxy(85,2); printf("Sinh vien: Vu Huu Sy - MI2-01 K66");
61     gotoxy(85,3); printf("MSSV: 20216958");
62     gotoxy(85,4); printf("Vien Toan ung dung va Tin hoc");
63     gotoxy(85,5); printf("Dai hoc Bach Khoa Ha Noi");
64     gotoxy(85,7); printf("BTL mon Ky Thuat Lap Trinh");
65     gotoxy(85,8); printf("GVHD: TS. Nguyen Thi Thanh Huyen");
66 }
67
68 // ham ve giao dien khung ngoai
69 void khungngoai()
```

```

70 {
71     TextColor(113);
72     for(int i=1; i<80;i++){
73         for(int j=1;j<29;j++){
74             gotoxy(i,j); printf(" ");
75         }
76     }
77     TextColor(11);
78     for (int i = 0; i <= 80; i++)
79     {
80         gotoxy(i, 0); printf("%c", 205);
81         gotoxy(i, 29); printf("%c", 205);
82     }
83     for (int i = 0; i <= 14; i++)
84     {
85         gotoxy(0, 14 - i); printf("%c", 186);
86         gotoxy(0, 14 + i); printf("%c", 186);
87         gotoxy(80, 14 - i); printf("%c", 186);
88         gotoxy(80, 14 + i); printf("%c", 186);
89     }
90     gotoxy(0, 0); printf("%c", 201);
91     gotoxy(80, 0); printf("%c", 187);
92     gotoxy(0, 29); printf("%c", 200);
93     gotoxy(80, 29); printf("%c", 188);
94     gotoxy(34, 0); TextColor(160); printf(" BAI TAP LON ");
95     gotoxy(2, 2); TextColor(31); printf("%c CHU DE 4: GIAI GAN DUNG PHUONG
↔ TRINH F(X) = 0 BANG PHUONG PHAP TIEP TUYEN %c", 175, 174);
96 }
97
98 // ham ve giao dien ben trong khung
99 void khungtrong()
100 {
101     TextColor(116);
102     for (int i = 0; i < 47; i++)
103     {
104         gotoxy(16 + i, 7); printf("%c", 205);

```

```

105     gotoxy(16 + i, 19); printf("%c", 205);
106 }
107 TextColor(31);
108     gotoxy(36,7); printf(" MENU ");
109     TextColor(113);
110     gotoxy(18,21); printf("$$$  $$  $$$$$$$$  $$$  $$$$$$$$\\n")
111     ↪ ; Sleep(40);
112     gotoxy(18,22); printf("$$$  $$  $$$$$$$$  $$$  $$$$$$$$\\n")
113     ↪ ; Sleep(40); TextColor(114);
114     gotoxy(18,23); printf("$$$$  $$$  $$$  $$$  \\n")
115     ↪ ; Sleep(40);
116     gotoxy(18,24); printf("$$$  $$  $$$  $$$  $$$  \\n")
117     ↪ ; Sleep(40); TextColor(116);
118     gotoxy(18,25); printf("$$$  $$  $$$  $$$$$$  $$$  \\n")
119     ↪ ; Sleep(40);
120     gotoxy(18,26); printf("$$$  $$  $$$  $$$$$$  $$$  \\n")
121     ↪ ;
122 }
123
124 // ham giao dien ket qua
125 void khungketqua()
126 {
127     TextColor(113);
128     for(int i=83; i<118;i++){
129         for(int j=1;j<29;j++){
130             gotoxy(i,j);
131             printf(" ");
132         }
133     }
134     TextColor(12);
135     for (int i = 82; i <= 118; i++)
136     {
137         gotoxy(i, 0); printf("%c", 205);
138         gotoxy(i, 29); printf("%c", 205);
139     }
140     for (int i = 0; i <= 29; i++)

```

```
135     {
136         gotoxy(82, i); printf("%c", 186);
137         gotoxy(118, i); printf("%c", 186);
138     }
139     gotoxy(82, 0); printf("%c", 201);
140     gotoxy(118, 0); printf("%c", 187);
141     gotoxy(82, 29); printf("%c", 200);
142     gotoxy(118, 29); printf("%c", 188);
143     gotoxy(96, 0); TextColor(160); printf(" Ket qua ");
144     gotoxy(100, 28); TextColor(120); printf("Made by Vu Huu Sy");
145 }
146
147 // ham in ra huong dan
148 void khunghuongdan(){
149     TextColor(113);
150     for(int i=84; i<118;i++){
151         for(int j=1;j<29;j++){
152             gotoxy(i,j);
153             printf(" ");
154         }
155     }
156     TextColor(10);
157     for (int i = 82; i <= 118; i++)
158     {
159         gotoxy(i, 0); printf("%c", 205);
160         gotoxy(i, 29); printf("%c", 205);
161     }
162     for (int i = 0; i <= 29; i++)
163     {
164         gotoxy(82, i); printf("%c", 186);
165         gotoxy(118, i); printf("%c", 186);
166     }
167     gotoxy(82, 0); printf("%c", 201);
168     gotoxy(118, 0); printf("%c", 187);
169     gotoxy(82, 29); printf("%c", 200);
170     gotoxy(118, 29); printf("%c", 188);
```



```

171 gotoxy(95, 0); TextColor(160); printf(" Huong dan ");
172 gotoxy(100, 28); TextColor(120); printf("Made by Vu Huu Sy");
173 TextColor(116);
174 gotoxy(84,2); printf("%c MENU duoc dieu khien bang cac", 175);
175 gotoxy(84,3); printf("phim chuc nang tren ban phim!");
176 gotoxy(84,5); printf("%c Chi tiet:", 175);
177 TextColor(117);
178 gotoxy(84,6); printf("- Phan ban phim chu:");
179 TextColor(113);
180 gotoxy(84,7); printf("+ An \"s\" hoac \"S\" de xuong");
181 gotoxy(84,8); printf("+ An \"w\" hoac \"W\" de len");
182 TextColor(117);
183 gotoxy(84,9); printf("- Phan ban phim so:");
184 TextColor(113);
185 gotoxy(84,10); printf("+ An \"2\" de xuong");
186 gotoxy(84,11); printf("+ An \"8\" de len");
187 TextColor(117);
188 gotoxy(84,13); printf("%c Cac phim mui ten hoat dong", 175);
189 gotoxy(84,14); printf("tuong tu.");
190 TextColor(116);
191 gotoxy(84,16); printf("%c Chu y:", 175);
192 TextColor(113);
193 gotoxy(84,17); printf("-Khi an xuong o chuc nang cuoi thi");
194 gotoxy(84,18); printf("se tro lai chuc nang dau tien.");
195 gotoxy(84,19); printf("-An phim Tab se xuong 2 lan.");
196 gotoxy(84,21); printf("%c An Enter de chuong trinh thuc", 175);
197 gotoxy(84,22); printf("hien chuc nang dang chon.");
198 gotoxy(84,24); printf("%c Khi thuc hien xong 1 chuc nang,", 175);
199 gotoxy(84,25); printf("an Enter de quay tro lai MENU.");
200 gotoxy(84,27); printf("%c An ESC de ket thuc chuong trinh.", 175);
201 }
202
203 // ham gioi thieu ve phuong phap tiep tuyen
204 void khungtieptuyen(){
205     TextColor(113);
206     for(int i=84; i<118;i++){

```

```

207     for(int j=1;j<29;j++){
208         gotoxy(i,j);
209         printf(" ");
210     }
211 }
212
213 TextColor(10);
214     for (int i = 82; i <= 118; i++)
215     {
216         gotoxy(i, 0); printf("%c", 205);
217         gotoxy(i, 29); printf("%c", 205);
218     }
219     for (int i = 0; i <= 29; i++)
220     {
221         gotoxy(82, i); printf("%c", 186);
222         gotoxy(118, i); printf("%c", 186);
223     }
224     gotoxy(82, 0); printf("%c", 201);
225     gotoxy(118, 0); printf("%c", 187);
226     gotoxy(82, 29); printf("%c", 200);
227     gotoxy(118, 29); printf("%c", 188);
228     gotoxy(89, 0); TextColor(160); printf(" Phương pháp tiếp tuyến ");
229     gotoxy(100, 28); TextColor(120); printf("Made by Vu Huu Sy");
230     TextColor(113);
231     gotoxy(84,2); printf("%c Trong GIAI TICH SO, phương pháp", 175);
232     gotoxy(84,3); printf("tiếp tuyến (con duoc goi la phương");
233     gotoxy(84,4); printf("phap Newton Raphson), dat ten theo");
234     gotoxy(84,5); printf("Isaac Newton va Joseph Raphson,");
235     gotoxy(84,6); printf("la mot phương pháp tìm nghiệm xấp");
236     gotoxy(84,7); printf("xi gần đúng của một hàm số f(x) có");
237     gotoxy(84,8); printf("tham số thực.");
238     gotoxy(84,10); printf("%c Định nghĩa: Điểm x0 duoc goi la", 175);
239     gotoxy(84,11); printf("điểm Fourier của f nếu: ");
240     gotoxy(84,12); printf("          f(x0).f\"(x0) > 0");
241     gotoxy(84,14); printf("%c Công thức lặp: ", 175);
242     gotoxy(84,15); printf("          X(n+1) = Xn - f(Xn)/f'(Xn)");

```

```

243     gotoxy(84,17); printf("%c Cong thuc sai so:", 175);
244     gotoxy(84,18); printf("- Sai so muc tieu: ");
245     gotoxy(84,19); printf("|Xn - X*| <= |f(Xn)| / m1");
246     gotoxy(84,21); printf("- Sai so hai xap xi lien tiep:");
247     gotoxy(84,22); printf("|Xn - X*|<=(M2/2m1)*(Xn-X(n-1))^2");
248     gotoxy(84,24); printf("Trong do: ");
249     gotoxy(84,25); printf("- m1 la min cua |f'(x)| tren (a,b)");
250     gotoxy(84,26); printf("- M2 la max cua |f\"(x)| tren (a,b)");
251
252 }
253
254 // ham giao dien khi ket thuc chuong trinh
255 void khungexit(){
256     int i = 0;
257     TextColor(113);
258     for(int i=0; i<120;i++){
259         for(int j=0;j<30;j++){
260             gotoxy(i,j); printf(" ", 178);
261         }
262     }
263
264     for (int i = 40; i <= 80; i++)
265     {
266         gotoxy(i, 0); printf("%c", 205);
267         gotoxy(i, 4); printf("%c", 205);
268         Sleep(1);
269     }
270     for (int i = 0; i <= 4; i++)
271     {
272         gotoxy(40, i); printf("%c", 186);
273         gotoxy(80, i); printf("%c", 186);
274         Sleep(1);
275     }
276     gotoxy(40, 0); printf("%c", 201);
277     gotoxy(80, 0); printf("%c", 187);
278     gotoxy(40, 4); printf("%c", 200);

```

```

279     gotoxy(80, 4); printf("%c", 188);
280
281     TextColor(116);
282     gotoxy(48,2); printf("KET THUC CHUONG TRINH !!!");
283     getch();
284 }
285
286 // ham giao dien chinh
287 void khung(){
288     khungngoai();
289     TextColor(113);
290     gotoxy(2,4); output();
291     khungketqua();
292     khungtrong();
293 }

```

Mã nguồn chính

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <windows.h>
4 #include <math.h>
5 #include <stdbool.h>
6 #include "giao_dien.h"
7
8 // cac hang so
9 #define MIN 1
10 #define MAX 8
11 #define N 100
12 #define max_loop 1000
13
14 // cac bien toan cuc
15 const char* option[] =
16 {
17     "Tim cac KPL nghiem cua phuong trinh",
18     "Thu hep khoang phan ly nghiem",
19     "Tim nghiem gan dung voi so lan lap n",

```

```

20     "Tim nghiem gan dung voi sai so e",
21     "Tim Xn thoa man |Xn - X(n-1)| <= e",
22     "Huong dan dieu khien MENU",
23     "Gioi thieu ve phuong phap tiep tuyen",
24     "Exit"
25 };
26
27 int degree;
28 int decimal_digit;
29 int distance_count;
30 double lower_bround;
31 double upper_bround;
32 double coeff_poly[N];
33 double survey_value[N];
34 double distance_value[N];
35
36 // cac ham trong chuong trinh chinh
37 void input();
38 void print_poly_to_file(FILE *file);
39 bool check_input(double a, double b);
40 void domain_solution();
41 double f(double x);
42 double df(double x);
43 double ddf(double x);
44 double f1(double x);
45 double f2(double x);
46 double min(double f(double x), double a, double b);
47 double max(double f(double x), double a, double b);
48 double sign(double x);
49 bool fourier_point(double a);
50 double sai_so(double x, double x_old, double m1, double M2, int ct_saiso);
51 void chucnang_1();
52 void chucnang_2();
53 void chucnang_3();
54 void chucnang_4();
55 void chucnang_5();

```

```

56 void menu();
57
58 // ham in phuong trinh ra file
59 void print_poly_to_file (FILE *file){
60     fprintf(file, "\n\nPhuong trinh: F(x) = %.*lfx^%d", decimal_digit,
        ↪ coeff_poly[0], degree);
61     for (int i = 1; i <= degree; i++){
62         if(coeff_poly[i] >= 0)
63             fprintf(file, " +%.*lfx^%d", decimal_digit, coeff_poly[i],
        ↪ degree - i);
64         else
65             fprintf(file, " %.*lfx^%d", decimal_digit, coeff_poly[i], degree
        ↪ - i);
66     }
67     fprintf(file, " = 0\n");
68 }
69
70 // ham nhap phuong trinh
71 void input() {
72     int c;
73     bool valid_input = false;
74     FILE* file = fopen("input.txt", "w");
75
76     if (file == NULL) {
77         TextColor(116);
78         gotoxy(4,4); printf("%c Error. Khong the mo file de ghi.", 175);
79         return;
80     }
81
82     while (!valid_input) {
83         khungngoai();
84         khungthongtin();
85         TextColor(113);
86         gotoxy(4,4); printf("- Xet %c(x) la da thuc bac n co dang:", 159);
87         gotoxy(4,6); printf(" %c(x) = a0*x^n + a1*x^(n-1) + a2*x^(n-2)
        ↪ +...+ an*x^0", 159);

```

```

88     gotoxy(4,8); printf("Hay nhap da thuc %c(x) co dang nhu tren de
↪ bat dau tinh toan!", 159);
89     gotoxy(4,10); printf("%c Nhap bac cua da thuc n = ", 175);
90
91     if (scanf("%d", &degree) != 1 || degree < 0 || degree > N) {
92         TextColor(116);
93         gotoxy(4,12); printf("%c Error! Hay nhap mot so nguyen khong am
↪ nho hon 100.", 175);
94         gotoxy(4,13); printf("%c An Enter de nhap lai da thuc!", 175);
95         while (getchar() != '\n');
96         getchar();
97         continue;
98     }
99     fprintf(file, "\nNguoi dung nhap vao: ");
100    fprintf(file, "\n- Bac cua da thuc: %d.", degree);
101
102    TextColor(112);
103    gotoxy(4,12); printf("%c So chu so phan thap phan muon hien thi la: ",
↪ 175);
104    if (scanf("%d", &decimal_digit) != 1 || decimal_digit < 0 ||
↪ decimal_digit > 20) {
105        TextColor(116);
106        gotoxy(4,13); printf("%c Error! Hay nhap mot so nguyen khong am
↪ nho hon 100.", 175);
107        gotoxy(4,14); printf("%c An Enter de nhap lai da thuc!", 175);
108        while (getchar() != '\n');
109        getchar();
110        continue;
111    }
112    fprintf(file, "\n- So chu so phan thap phan muon hien thi la: %d.",
↪ decimal_digit);
113
114    TextColor(117);
115    gotoxy(4,14); printf("%c Nhap cac he so cua da thuc: ", 175);
116    fprintf(file, "\n\n- Cac he so cua da thuc:");
117    for (int i = 0; i <= degree; i++) {

```

```

118         gotoxy(4,15 + i); printf("+) a%d = ", i);
119         if (scanf("%lf", &coeff_poly[i]) != 1) {
120             TextColor(116);
121             gotoxy(4,16 + i); printf("%c Error! He so khong hop le.",
↪ 175);
122             gotoxy(4,17 + i); printf("%c An Enter de nhap lai da thuc!",
↪ 175);
123             while (getchar() != '\n');
124             getchar();
125             valid_input = false;
126             break;
127         }
128         fprintf(file, "\n +> a%d = %.1f", i, decimal_digit,
↪ coeff_poly[i]);
129         valid_input = true;
130     }
131 }
132
133 print_poly_to_file(file);
134 fclose(file);
135
136 TextColor(113);
137 gotoxy(4,18 + degree); printf("%c Phuong trinh vua nhap la: ",175);
138 gotoxy(6,19 + degree); output();
139 TextColor(116);
140 gotoxy(4,21 + degree); printf("%c An Enter de bat dau su dung chuong
↪ trinh!",175);
141 getch();
142 }
143
144 // ham in ra phuong trinh khi chay chuong trinh
145 void output()
146 {
147     printf("%c(x) = %.3lfx~%d", 159, coeff_poly[0], degree);
148     for (int i = 1; i <= degree; i++){
149         if(coeff_poly[i] >= 0)

```



```
150     printf(" +%.3lfx^%d", coeff_poly[i], degree - i);
151     else
152         printf(" %.3lfx^%d", coeff_poly[i], degree - i);
153 }
154 printf(" = 0");
155 TextColor(0);
156 }
157
158 // ham tinh gia tri f(x)
159 double f(double x)
160 {
161     double temp = 0;
162     for (int i = degree; i >= 0; i--) {
163         temp += coeff_poly[degree - i] * pow(x, i);
164     }
165     return temp;
166 }
167
168 // ham tinh gia tri dao ham cap 1 cua f(x)
169 double df(double x) {
170     double h = 1e-7;
171     return ((f(x + h) - f(x - h)) / (2 * h));
172 }
173
174 // ham tinh gia tri dao ham cap 2 cua f(x)
175 double ddf(double x) {
176     double h = 1e-7;
177     return ((df(x + h) - df(x - h)) / (2 * h));
178 }
179
180 // ham tim min cua ham so
181 double min(double f(double x), double a, double b){
182     double x0 = a, alpha;
183     double min = a;
184     alpha = (b - a) / 10000;
185     for (int i = 0; i < 10000; i++){
```

```

186         if (f(min) > f(x0))
187             min = x0;
188         x0 = x0 + alpha;
189     }
190     return f(min);
191 }
192
193 // ham tim max cua ham so
194 double max(double f(double x), double a, double b){
195     double x0 = a, alpha;
196     double max = a;
197     alpha = (b - a) / 10000;
198     for (int i = 0; i < 10000; i++){
199         if (f(max) < f(x0))
200             max = x0;
201         x0 = x0 + alpha;
202     }
203     return f(max);
204 }
205
206 // ham xac dinh dau cua mot gia tri x
207 double sign(double x)
208 {
209     if (x>=0) return 1;
210     else return -1;
211 }
212
213 // ham tra ve tri tuyet doi dao ham cap 1 cua f(x)
214 double f1(double x){
215     return fabs(df(x));
216 }
217
218 // ham tra ve tri tuyet doi dao ham cap 2 cua f(x)
219 double f2(double x){
220     return fabs(ddf(x));
221 }

```

```

222
223 // ham tinh sai so
224 double sai_so(double x, double x_old, double m1, double M2, int ct_saiso){
225     if (ct_saiso == 1)
226         return fabs(f(x))/ m1;
227     else if (ct_saiso == 2)
228         return M2 * pow((x - x_old), 2)/ (2 * m1);
229 }
230
231 // ham tim diem fourier
232 bool fourier_point(double a){
233     if (sign(f(a)) == sign(ddf(a)))
234         return true;
235     else return false;
236 }
237
238 // ham kiem tra dieu kien hoi tu
239 bool check_input(double a, double b){
240     if (a == b) return false;
241     else if (sign(f(a)) == sign(f(b))) return false;
242     else if (a > b) return false;
243     else if (sign(max(df, a, b)) != sign(min(df, a, b))) return false;
244     else if (sign(max(ddf, a, b)) != sign(min(ddf, a, b))) return false;
245     else return true;
246 }
247
248 // ham tim mien chua nghiem
249 void domain_solution()
250 {
251     double temp[N], max, k;
252     upper_bround = -1;
253     lower_bround = 1;
254
255     for (int i = 0; i <= degree; i++) {
256         temp[i] = coeff_poly[i];
257     }

```

```

258     do {
259         max = 0;
260         k = 0;
261         if (temp[0] < 0) {
262             for (int i = 0; i <= degree; i++) {
263                 temp[i] = -temp[i];
264             }
265         }
266         for (int i = 1; i <= degree; i++) {
267             if (temp[i] < 0) {
268                 k = i;
269                 break;
270             }
271         }
272         for (int i = 1; i <= degree; i++) {
273             if (temp[i] < 0) {
274                 if (abs(temp[i]) > max) max = abs(temp[i]);
275             }
276         }
277         if (max == 0) {
278             if (upper_bround == -1) upper_bround = 0;
279             else lower_bround = 0;
280         }
281         else {
282             if (upper_bround == -1) upper_bround = 1 + pow((max / temp[0])
↪ , 1 / k);
283             else lower_bround = -(1 + pow((max / temp[0]), 1 / k));
284         }
285         for (int i = 0; i <= degree; i++) {
286             if (i % 2 == 1)
287                 temp[degree - i] = -temp[degree - i];
288         }
289     }
290     while (lower_bround > 0);
291 }
292

```

```

293 // ham thuc hien chuc nang 1
294 void chucnang_1()
295 {
296     double x0, x1, sign, temp0, temp1, value1, value2;
297     double eta = 1e-11;
298     int i = 0, j = 0, k = 1;
299     FILE* file = fopen("chucnang_1.txt", "w");
300
301     domain_solution();
302     TextColor(113);
303     fprintf(file, "\nChuc nang 1: Tim cac khoang phan ly nghiem cua phuong
↪ trinh.");
304     fprintf(file, "\n\nINPUT: Bac cua phuong trinh, cac he so cua phuong
↪ trinh f(x) = 0.");
305     fprintf(file, "\nOUTPUT: Cac khoang phan ly nghiem cua phuong trinh.")
↪ ;
306     print_poly_to_file(file);
307
308     if (degree == 0) {
309         if (coeff_poly[0] == 0){
310             gotoxy(84,2); printf("Khoang phan ly nghiem: (-00,+00)");
311             gotoxy(84,4); printf("Phuong trinh co vo so nghiem!");
312             fprintf(file, "\nKhoang phan ly nghiem: (-oo,+oo)\n");
313             fprintf(file, "\nPhuong trinh co vo so nghiem!\n");
314         }
315         else{
316             gotoxy(84,2); printf("Khoang phan ly nghiem: (--,--)");
317             gotoxy(84,4); printf("Phuong trinh vo nghiem!");
318             fprintf(file, "\nKhoang phan ly nghiem: (--,--)\n");
319             fprintf(file, "\nPhuong trinh vo nghiem!\n");
320         }
321     }
322     else if (degree == 1){
323         gotoxy(84,2); printf("Khoang phan ly nghiem: ");
324         gotoxy(84,3); printf("(%.1f , %.1f)", decimal_digit, (-
↪ coeff_poly[1] / coeff_poly[0]) - 1, decimal_digit, (- coeff_poly[1]

```

```

    ↪ / coeff_poly[0]) + 1);
325     fprintf(file, "\nKhoang phan ly nghiem: (%.*lf , %.*lf)\n",
    ↪ decimal_digit, (- coeff_poly[1] / coeff_poly[0]) - 1, decimal_digit
    ↪ , (- coeff_poly[1] / coeff_poly[0]) + 1);
326 }
327 else if (lower_bround == upper_bround) {
328     if (f(lower_bround) == 0) {
329         gotoxy(84,2); printf("Khoang phan ly nghiem: ");
330         gotoxy(84,3); printf("(%.*lf , %.*lf)", decimal_digit,
    ↪ lower_bround - 1, decimal_digit, upper_bround + 1);
331         fprintf(file, "\nKhoang phan ly nghiem: (%.*lf , %.*lf)\n",
    ↪ decimal_digit, lower_bround - 1, decimal_digit, upper_bround + 1);
332     }
333     else{
334         gotoxy(84,2); printf("Khoang phan ly nghiem: (--,--)");
335         gotoxy(84,4); printf("Phuong trinh vo nghiem!");
336         fprintf(file, "\nKhoang phan ly nghiem: (--,--)\n");
337         fprintf(file, "\nPhuong trinh vo nghiem!\n");
338     }
339 }
340 else {
341     x1 = lower_bround;
342     while (x1 < upper_bround) {
343         x0 = x1;
344         temp0 = df(x0);
345         if (temp0 < 0)
346             sign = -1;
347         else
348             sign = 1;
349         x1 = x0 + sign * eta * temp0;
350         temp1 = df(x1);
351         for (int i = 0; i < max_loop; i++) {
352             if (temp0 * temp1 > 0) {
353                 while (eta < 0.008) {
354                     eta = eta * 2;
355                     x1 = x0 + sign * eta * temp0;

```

```

356         if (df(x1) * temp0 < 0) {
357             eta = eta / 2;
358             break;
359         }
360     }
361 }
362 else {
363     while (eta > 0) {
364         eta = eta / 2;
365         x1 = x0 + sign * eta * temp0;
366         if (df(x1) * temp0 > 0)
367             break;
368     }
369 }
370 x1 = x0 + sign * eta * temp0;
371 x0 = x1;
372 if (abs(df(x1)) < 1e-4) {
373     survey_value[k] = x1;
374     k++;
375     break;
376 }
377 eta = 1e-11;
378 temp0 = df(x0);
379 x1 = x0 + sign * eta * temp0;
380 temp1 = df(x1);
381 if (x1 > upper_bround)
382     break;
383 }
384 x1 = x1 + 0.001;
385 }
386 survey_value[0] = lower_bround;
387 survey_value[k] = upper_bround;
388 distance_count = 1;
389 for (i = 0; i < k; i++) {
390     value1 = f(survey_value[i]);
391     value2 = f(survey_value[i + 1]);

```

```

392     if (value1 * value2 < 0) {
393         distance_value[j] = survey_value[i];
394         j += 1;
395         distance_value[j] = survey_value[i+1];
396         j += 1;
397         gotoxy(84, 2+3*(distance_count - 1));
398         printf("%d. Khoảng phân ly nghiệm thu %d", distance_count,
↪ distance_count);
399         gotoxy(84, 3+3*(distance_count - 1));
400         printf("%c Can duoi: %.*lf", 175, decimal_digit, survey_value[
↪ i]);
401         gotoxy(84, 4+3*(distance_count - 1));
402         printf("%c Can tren: %.*lf", 175, decimal_digit, survey_value[
↪ i+1]);
403         fprintf(file, "\n\nKhoang phan ly nghiệm thu %d",
↪ distance_count);
404         fprintf(file, "\n- Can duoi: %.*lf", decimal_digit,
↪ survey_value[i]);
405         fprintf(file, "\n- Can tren: %.*lf", decimal_digit,
↪ survey_value[i + 1]);
406         distance_count += 1;
407     }
408 }
409 }
410 fprintf(file, "\n\nKet thuc.");
411 fclose(file);
412 }
413
414 // ham thuc hien chuc nang 2
415 void chucnang_2()
416 {
417     int i = 0, chon, lan_lap = 1;
418     double temp=0;
419     double a, b, value1, value2;
420     FILE* file = fopen("chucnang_2.txt", "w");
421

```



```

422 fprintf(file, "\nChuc nang 2: Thu hep khoang phan ly nghiem (a,b) thoa
    ↪ man |a-b| <= 0.5 bang cach su dung phuong phap chia doi\n");
423 print_poly_to_file(file);
424 fprintf(file, "\n\nINPUT: cac KPL nghiem tim duoc o chuc nang 1, lua
    ↪ chon cua nguoi dung.");
425 fprintf(file, "\n\nOUTPUT: KPL nghiem duoc rut gon thoa man |a - b| <=
    ↪ 0.5.");
426
427 khung();
428 TextColor(287);
429 gotoxy(33,7); printf(" CHUC NANG 2 ");
430 TextColor(113);
431 gotoxy(16, 9); printf("2. Thu hep khoang phan ly nghiem (a,b) thoa man")
    ↪ ;
432 gotoxy(16, 10); printf("|a - b| <= 0.5 su dung phuong phap chia doi");
433
434 if (distance_count == 0){
435     TextColor(116);
436     gotoxy(16, 12); printf("!!! Khong ton tai khoang phan ly de rut gon.
    ↪ ");
437     gotoxy(16, 13); printf("- Neu phuong trinh co nghiem, hay thuc hien"
    ↪ );
438     gotoxy(16, 14); printf("chuc nang 1 truoc de co khoang phan ly
    ↪ nghiem!");
439     fprintf(file, "\n!!! Khong ton tai khoang phan ly nghiem de rut gon.
    ↪ ");
440     fprintf(file, "\n\nKet thuc.");
441     fclose(file);
442 }
443 else{
444     while (1){
445         TextColor(120);
446         gotoxy(16, 11); printf("(Nhap so nguyen duong nho hon %d!)",
    ↪ distance_count);
447         TextColor(113);
448         gotoxy(16, 13); printf("Chon khoang phan ly nghiem thu: ");

```

```

449     if (scanf("%d", &chon) != 1 || chon < 1 || chon >= distance_count) {
450         TextColor(116);
451         gotoxy(16,14); printf("%c Khoảng phân ly nghiệm không hợp lệ. ",
↪ 175);
452         gotoxy(16,15); printf("An Enter để nhập lại!");
453         while (getchar() != '\n');
454         getchar();
455         khung();
456         TextColor(113);
457         gotoxy(16, 9); printf("2. Thu hẹp khoảng phân ly nghiệm (a,b) thỏa
↪ man");
458         gotoxy(16, 10); printf("|a - b| <= 0.5 sử dụng phương pháp chia
↪ đôi");
459         continue;
460     }
461     break;
462 }
463 fprintf(file, "\nKhoảng phân ly người dùng chọn: %d\n", chon);
464 a = distance_value[2 * chon - 2];
465 b = distance_value[2 * chon - 1];
466
467 gotoxy(84, 3); printf("Khoảng phân ly ban đầu là:");
468 gotoxy(84, 4); printf("Can duoi a: %.*lf", decimal_digit, distance_value
↪ [2 * chon - 2]);
469 gotoxy(84, 5); printf("Can tren b: %.*lf", decimal_digit, distance_value
↪ [2 * chon - 1]);
470 fprintf(file, "\nKhoảng phân ly ban đầu là:\n");
471 fprintf(file, "- Can duoi a: %.*lf\n", decimal_digit, a);
472 fprintf(file, "- Can tren b: %.*lf\n", decimal_digit, b);
473 fprintf(file, "\nSử dụng phương pháp chia đôi để thu hẹp khoảng cách
↪ ly:\n", lan_lap);
474 while((b-a) > 0.5){
475     temp = (a+b) / 2.0;
476     value1 = f(a);
477     value2 = f(temp);
478     if(value1 * value2 < 0){

```

```

479         b = temp;
480     }
481     else a = temp;
482     fprintf(file, "\nLan lap thu %d\n", lan_lap);
483     fprintf(file, "- Can duoi a: %.*lf\n", decimal_digit, a);
484     fprintf(file, "- Can tren b: %.*lf\n", decimal_digit, b);
485     lan_lap++;
486 }
487
488 gotoxy(84, 7); printf("Khoang phan ly sau khi rut gon:");
489 gotoxy(84, 8); printf("Can duoi a: %.*lf", decimal_digit, a);
490 gotoxy(84, 9); printf("Can tren b: %.*lf", decimal_digit, b);
491 gotoxy(84, 11); printf("|a - b| = %.*lf <= 0.5", decimal_digit, fabs(a-b
    ↪ ));
492 gotoxy(84, 13); printf("So lan lap chia doi: %d", lan_lap - 1);
493 fprintf(file, "\n\nKhoang phan ly nghiem sau khi rut gon:");
494 fprintf(file, "\n- Can duoi a: %.*lf", decimal_digit, a);
495 fprintf(file, "\n- Can tren b: %.*lf", decimal_digit, b);
496 fprintf(file, "\n\n|a - b| = %.*lf <= 0.5 thoa man de bai.",
    ↪ decimal_digit, fabs(a-b));
497 fprintf(file, "\n\nSo lan lap chia doi: %d", lan_lap - 1);
498 }
499     fprintf(file, "\n\nKet thuc.");
500     fclose(file);
501 }
502
503 // ham thuc hien chuc nang 3
504 void chucnang_3()
505 {
506     double a, b, eps;
507     double x, x_old, m1, M2, delta1, delta2;
508     int n, i, j = 0;
509     bool check;
510     FILE* file = fopen("chucnang_3.txt", "w");
511
512     fprintf(file, "\nChuc nang 3: Tim nghiem gan dung voi so lan lap cho

```

```

    ↪ trước trong khoảng (a, b) và đánh giá sai số epsilon theo hai công
    ↪ thức sai số.");
513     print_poly_to_file(file);
514     fprintf(file, "\nINPUT: KPL nghiệm (a,b), số lần lặp.");
515     fprintf(file, "\nOUTPUT: Nghiệm gần đúng  $x_n$ , sai số epsilon.");
516     khung();
517     TextColor(287);
518     gotoxy(33,7); printf(" CHUC NANG 3 ");
519     TextColor(113);
520     gotoxy(16, 9); printf("%c Nhập khoảng phân ly nghiệm (a,b): ", 175);
521     gotoxy(16, 10); printf("%c Nhập cận dưới a: ", 175);scanf("%lf", &a);
522     gotoxy(16, 11); printf("%c Nhập cận trên b: ", 175);scanf("%lf", &b);
523     fprintf(file, "\n\nKhoảng phân ly nghiệm (a,b):");
524     fprintf(file, "\n- Cận dưới a: %.*lf", decimal_digit, a);
525     fprintf(file, "\n- Cận trên b: %.*lf", decimal_digit, b);
526     check = check_input(a, b);
527
528     if (check == true){
529         TextColor(113);
530         gotoxy(16,12); printf("%c Nhập số lần lặp: n = ", 175);scanf("%d", &
    ↪ n);
531         fprintf(file, "\n\nSố lần lặp người dùng nhập là: %d", n);
532         if (fourier_point(a) == true) x_old = a;
533         else if (fourier_point(a) == false) x_old = b;
534
535         m1 = min(f1, a, b);
536         M2 = max(f2, a, b);
537         fprintf(file, "\n\n- Giá trị nhỏ nhất của  $|f'(x)|$  là: m1 = %.*lf",
    ↪ decimal_digit, m1);
538         fprintf(file, "\n- Giá trị lớn nhất của  $|f''(x)|$  là: M2 = %.*lf",
    ↪ decimal_digit, M2);
539
540         gotoxy(84,2); printf("%c Điểm Fourier là:", 175);
541         gotoxy(84,3); printf("                x0 = %.*lf", decimal_digit, x_old);
542         fprintf(file, "\n\nĐiểm Fourier là: x0 = %.*lf", decimal_digit, x_old)
    ↪ ;

```

```

543
544     x = x_old;
545     for (i = 1; i<= n; i++){
546         TextColor(113);
547         x_old = x;
548         x = x_old - f(x_old) / df(x_old);
549         delta1 = sai_so(x, x_old, m1, M2, 1);
550         delta2 = sai_so(x, x_old, m1, M2, 2);
551         gotoxy(84,5 + j); printf("Lan lap %d: x%d = %.1f", i, i,
↪ decimal_digit, x);
552         TextColor(118);
553         gotoxy(84,6 + j); printf("- Sai so 1 = %.15lf", delta1);
554         TextColor(117);
555         gotoxy(84,7 + j); printf("- Sai so 2 = %.15lf", delta2);
556         j += 4;
557         fprintf(file, "\n\nLan lap thu %d: ", i);
558         fprintf(file, "\nNghiem xap xi x%d = %.1f", i, decimal_digit, x
↪ );
559         fprintf(file, "\nSai so muc tieu                = %.20lf", delta1);
560         fprintf(file, "\nSai so hai xap xi lien tiep = %.20lf", delta2);
561     }
562     if(delta1 < delta2){
563         TextColor(112);
564         gotoxy(16,14); printf("%c Danh gia sai so:", 175);
565         TextColor(114);
566         gotoxy(16,15); printf("- Phuong phap tiep tuyen co toc do hoi tu cao
↪ ");
567         gotoxy(16,16); printf("va ket qua co do chinh xac cao.");
568         gotoxy(16,17); printf("- Cong thuc sai so 1 co sai so nho hon so voi
↪ ");
569         gotoxy(16,18); printf("cong thuc sai so 2 o nhung lan lap dau.");
570         fprintf(file, "\n\nDanh gia sai so cua hai cong thuc:");
571         fprintf(file, "\n- Uu diem: Toc do hoi tu cao, ket qua co do chinh
↪ xac cao.");
572         fprintf(file, "\n- Nhuoc diem: Dieu kien lua chon x0 ban dau kha
↪ chat.");

```

```

573     fprintf(file, "\n- Cong thuc sai so 1 co sai so nho hon so voi cong
↪ thuc sai so 2 o nhung lan lap dau. ");
574 }
575 else {
576     TextColor(112);
577     gotoxy(16,14); printf("%c Danh gia sai so:", 175);
578     TextColor(114);
579     gotoxy(16,15); printf("- Phuong phap tiep tuyen co toc do hoi tu cao
↪ ");
580     gotoxy(16,16); printf("va ket qua co do chinh xac cao.");
581     gotoxy(16,17); printf("- Cong thuc sai so 2 co sai so nho hon so voi
↪ ");
582     gotoxy(16,18); printf("cong thuc sai so 1 o nhung lan lap dau.");
583     fprintf(file, "\n\nDanh gia sai so cua hai cong thuc:");
584     fprintf(file, "\n- Uu diem: Toc do hoi tu cao, ket qua co do chinh
↪ xac cao.");
585     fprintf(file, "\n- Nhuoc diem: Dieu kien lua chon x0 ban dau kha
↪ chat.");
586     fprintf(file, "\n- Cong thuc sai so 2 co sai so nho hon so voi cong
↪ thuc sai so 1 o nhung lan lap dau. ");
587 }
588 fprintf(file, "\n\nGhi chu: Vi phuong phap tiep tuyen hoi tu nhanh va
↪ co sai so rat nho gan 0 nen sai so em se de nhieu chu so phan thap
↪ phan hon.");
589 fprintf(file, "\n\nKet thuc.");
590 fclose(file);
591 }
592 else if (check == false){
593     TextColor(116);
594     gotoxy(16, 13); printf("!!! Error: Loi nhap KPL nghiem.");
595     gotoxy(16, 14); printf("%c KPL nghiem khong thoa man dieu kien hoi
↪ tu", 175);
596     gotoxy(16, 15); printf("cua phuong phap tiep tuyen.");
597     fprintf(file, "\n\nINPUT khong hop le. Khoang phan ly nghiem khong
↪ thoa man dieu kien hoi tu cua phuong phap tiep tuyen.\n\nKet thuc.
↪ ");

```

```

598         fclose(file);
599     }
600 }
601
602 // ham thuc hien chuc nang 4
603 void chucnang_4()
604 {
605     double a, b, eps;
606     double x, x_old, m1, M2, delta;
607     int ct_saiso, i = 1;
608     bool check;
609     FILE* file = fopen("chucnang_4.txt", "w");
610
611     fprintf(file, "\nChuc nang 4: Tim nghiem gan dung trong khoang (a, b)
    ↪ voi sai so epsilon cho truoc.");
612     print_poly_to_file(file);
613     fprintf(file, "\nINPUT: KPL nghiem (a,b), sai so epsilon, cong thuc
    ↪ sai so.");
614     fprintf(file, "\nOUTPUT: Nghiem gan dung Xn thoa man sai so epsilon.");
615
616     khung();
617     TextColor(287);
618     gotoxy(33,7); printf(" CHUC NANG 4 ");
619     TextColor(113);
620     gotoxy(16, 9); printf("%c Nhap khoang phan ly nghiem (a,b): ", 175);
621     gotoxy(16, 10); printf("%c Nhap can duoi a: ", 175);scanf("%lf", &a);
622     gotoxy(16, 11); printf("%c Nhap can tren b: ", 175);scanf("%lf", &b);
623     fprintf(file, "\n\nKhoang phan ly nghiem (a,b):");
624     fprintf(file, "\n- Can duoi a: %.*lf", decimal_digit, a);
625     fprintf(file, "\n- Can tren b: %.*lf", decimal_digit, b);
626
627     check = check_input(a, b);
628     if (check == true){
629         // tim nghiem trong KPL nghiem hop le
630         TextColor(113);
631         gotoxy(16, 12); printf("%c Nhap sai so: ", 175); scanf("%lf", &eps

```

```

    ↪ );
632     fprintf(file, "\nSai so epsilon: %.20lf", eps);
633     gotoxy(16,13); printf("Cong thuc sai so:");
634     gotoxy(16,14); printf("1. |Xn - X*| <= |f(Xn)| / m1");
635     gotoxy(16,15); printf("2. |Xn - X*| <= (M2 / 2m1) * (Xn - X(n-1))^2");
636     gotoxy(16,17); printf("%c Chon cong thuc sai so: ", 175); scanf("%d",
    ↪ &ct_saiso);

637
638     if (ct_saiso == 1){
639         fprintf(file, "\n\nNguoi dung chon cong thuc sai so muc tieu:");
640         fprintf(file, "\n\t\t|Xn - X*| <= |f(Xn)| / m1");
641     }
642     else if (ct_saiso == 2){
643         fprintf(file, "\n\nNguoi dung chon cong thuc sai so theo hai xap xi
    ↪ lien tiep:");
644         fprintf(file, "\n\t\t|Xn - X*| <= (M2 / 2m1) * (Xn - X(n-1))^2");
645     }
646
647     m1 = min(f1, a, b);
648     M2 = max(f2, a, b);
649     fprintf(file, "\n\n- Gia tri nho nhat cua |f'(x)| la: m1 = %.1lf",
    ↪ decimal_digit, m1);
650     fprintf(file, "\n\n- Gia tri lon nhat cua |f''(x)| la: M2 = %.1lf",
    ↪ decimal_digit, M2);
651
652     if (fourier_point(a) == true) x_old = a;    // diem fourier
653     else if (fourier_point(a) == false) x_old = b;
654     x = x_old;
655     gotoxy(84,3); printf("%c Diem Fourier la:", 175);
656     gotoxy(84,4); printf("          x0 = %.1lf", decimal_digit, x_old);
657     fprintf(file, "\n\nDiem Fourier la: x0 = %.1lf \n", decimal_digit,
    ↪ x_old);
658
659     do {
660         x_old = x;
661         x = x_old - f(x_old) / df(x_old);

```



```

662     delta = sai_so(x, x_old, m1, M2, ct_saiso);
663     gotoxy(84,4 + i); printf("Lan lap %d: x%d = %.1f", i, i,
↪ decimal_digit, x);
664     fprintf(file, "\n\nLan lap thu %d: ", i);
665     fprintf(file, "Nghiem x%d = %.1f, Sai so = %.20lf", i,
↪ decimal_digit, x, delta);
666     i++;
667 }
668 while (delta > eps);
669
670 gotoxy(84,5 + i); printf("Ket luan: ");
671 gotoxy(84,6 + i); printf("%c So lan lap: %d", 175, i - 1);
672 gotoxy(84,8 + i); printf("%c Nghiem gan dung:", 175);
673 gotoxy(84,9 + i); printf("          x = %.1f", decimal_digit, x);
674 gotoxy(84,11 + i); printf("%c Sai so = %.20lf", 175, delta);
675     fprintf(file, "\n\nKet luan: ");
676     fprintf(file, "\nSo lan lap: %d", i - 1);
677     fprintf(file, "\nNghiem gan dung: x = %.1f", decimal_digit, x);
678     fprintf(file, "\nSai so = %.20lf <= %.20lf thoa man yeu cau de bai."
↪ , delta, eps);
679     fprintf(file, "\n\nGhi chu: Vi phuong phap tiep tuyen hoi tu nhanh
↪ va co sai so rat nho gan 0 nen sai so em se de nhieu chu so phan
↪ thap phan hon.");
680     fprintf(file, "\n\nKet thuc.");
681     fclose(file);
682 }
683 else if (check == false){
684     TextColor(116);
685     gotoxy(16, 13); printf("!!! Error: Loi nhap KPL nghiem.");
686     gotoxy(16, 14); printf("%c KPL nghiem khong thoa man dieu kien hoi
↪ tu", 175);
687     gotoxy(16, 15); printf("cua phuong phap tiep tuyen.");
688     fprintf(file, "\n\nINPUT khong hop le. Khoang phan ly nghiem khong
↪ thoa man dieu kien hoi tu cua phuong phap tiep tuyen.\n\nKet thuc.
↪ ");
689     fclose(file);

```

```

690 }
691 }
692
693 // ham thuc hien chuc nang 5
694 void chucnang_5()
695 {
696     double a, b, eps;
697     double x, x_old, m1, M2, delta;
698     int i = 1, j = 1;
699     bool check;
700     FILE* file = fopen("chucnang_5.txt", "w");
701
702     fprintf(file, "\nChuc nang 5: Tim nghiem gan dung Xn trong khoang (a,
↪ b) thoa man dieu kien:  $|X_n - X_{n-1}| \leq e$ ");
703     print_poly_to_file(file);
704     fprintf(file, "\nINPUT: KPL nghiem (a,b), sai so epsilon");
705     fprintf(file, "\nOUTPUT: Nghiem gan dung Xn thoa man  $|X_n - X_{n-1}| \leq e$ .
↪ ");
706     m1 = min(f1, a, b);
707     M2 = max(f2, a, b);
708
709     khung();
710     TextColor(287);
711     gotoxy(33,7); printf(" CHUC NANG 5 ");
712     TextColor(113);
713     gotoxy(16, 9); printf("%c Nhap khoang phan ly nghiem (a,b): ", 175);
714     gotoxy(16, 10); printf("%c Nhap can duoi a: ", 175);scanf("%lf", &a);
715     gotoxy(16, 11); printf("%c Nhap can tren b: ", 175);scanf("%lf", &b);
716     fprintf(file, "\n\nKhoang phan ly nghiem (a,b):");
717     fprintf(file, "\n- Can duoi a:  $%.1f$ ", decimal_digit, a);
718     fprintf(file, "\n- Can tren b:  $%.1f$ ", decimal_digit, b);
719
720     check = check_input(a, b);
721     if (check == true){
722
723         TextColor(113);

```

```

724     gotoxy(16, 12); printf("%c Nhập sai so: ", 175); scanf("%lf", &eps
↪ );
725     fprintf(file, "\n\nSai so epsilon: %.20lf", eps);
726
727     if (fourier_point(a) == true) x_old = a;
728     else if (fourier_point(a) == false) x_old = b;
729     x = x_old;
730     gotoxy(84,2); printf("%c Diem Fourier la:", 175);
731     gotoxy(84,3); printf("          x0 = %.1lf", decimal_digit, x_old);
732     fprintf(file, "\n\nDiem Fourier la: x0 = %.1lf \n", decimal_digit,
↪ x_old);
733
734     do {
735         x_old = x;
736         x = x_old - f(x_old) / df(x_old);
737         delta = fabs(x - x_old);
738         gotoxy(84,4 + j); printf("Lan lap %d: x%d = %.1lf", i, i,
↪ decimal_digit, x);
739         gotoxy(84,5 + j); printf("|x%d - x%d| = %.15lf", i, i - 1, delta
↪ );
740         fprintf(file, "\n\nLan lap thu %d:", i);
741         fprintf(file, "    x%d = %.1lf", i, decimal_digit, x);
742         fprintf(file, "\n- |x%d - x%d| = %.20lf", i, i - 1, delta);
743         i++;
744         j += 3;
745     }
746     while (delta > eps);
747     TextColor(125);
748     gotoxy(16,14); printf("Ket luan: ");
749     gotoxy(16,15); printf("%c So lan lap: %d", 175, i - 1);
750     gotoxy(16,16); printf("%c |x%d - x%d| = %.15lf <= %.15lf", 175, i - 1,
↪ i - 2, delta, eps);
751     gotoxy(16,17); printf("%c Nghiem gan dung: x%d = %.1lf", 175, i - 1,
↪ decimal_digit, x);
752     fprintf(file, "\n\nKet luan: ");
753     fprintf(file, "\nSo lan lap: %d", i - 1);

```

```

754     fprintf(file, "\nNghiem gan dung: x%d = %.*lf", i - 1, decimal_digit
↪ , x);
755     fprintf(file, "\nSai so |x%d - x%d| = %.20lf <= %.20lf thoa man yeu
↪ cau de bai.\n\nKet thuc.", i - 1, i - 2, delta, eps);
756     fclose(file);
757 }
758 else if (check == false){
759     TextColor(116);
760     gotoxy(16, 13); printf("!!! Error: Loi nhap KPL nghiem.");
761     gotoxy(16, 14); printf("%c KPL nghiem khong thoa man dieu kien hoi
↪ tu", 175);
762     gotoxy(16, 15); printf("cua phuong phap tiep tuyen.");
763     fprintf(file, "\n\nINPUT khong hop le. Khoang phan ly nghiem khong
↪ thoa man dieu kien hoi tu cua phuong phap tiep tuyen.\n\nKet thuc.
↪ ");
764     fclose(file);
765 }
766 }
767
768 void menu()
769 {
770     int position = 1;
771     int keyPress = 0;
772     khung();
773     while (keyPress != 13) {
774         for (int i = 1; i <= MAX; i++) {
775             gotoxy(16, 9 + i);
776             arrowHere(i, position);
777             printf("%d. %s \n", i, option[i - 1]);
778         }
779         keyPress = getch();
780         if (keyPress == 80 || keyPress == 's' || keyPress == 'S' ||
↪ keyPress == '2') {
781             if (position == MAX) position = MIN;
782             else position++;
783         }

```

```
784     else if (keyPress == 72 || keyPress == 'w' || keyPress == 'W' ||  
↪ keyPress == '8') {  
785         if (position == MIN ) position = MAX;  
786         else position--;  
787     }  
788     else if (keyPress == 9) {  
789         position += 2;  
790         if (position > MAX) position = MIN;  
791     }  
792     else if (keyPress == 27) {  
793         system("cls");  
794         return;  
795     }  
796 }  
797 switch (position) {  
798 case 1:  
799     chucnang_1();  
800     getch();  
801     menu();  
802     break;  
803 case 2:  
804     chucnang_2();  
805     getch();  
806     menu();  
807     break;  
808 case 3:  
809     chucnang_3();  
810     getch();  
811     menu();  
812     break;  
813 case 4:  
814     chucnang_4();  
815     getch();  
816     menu();  
817     break;  
818 case 5:
```

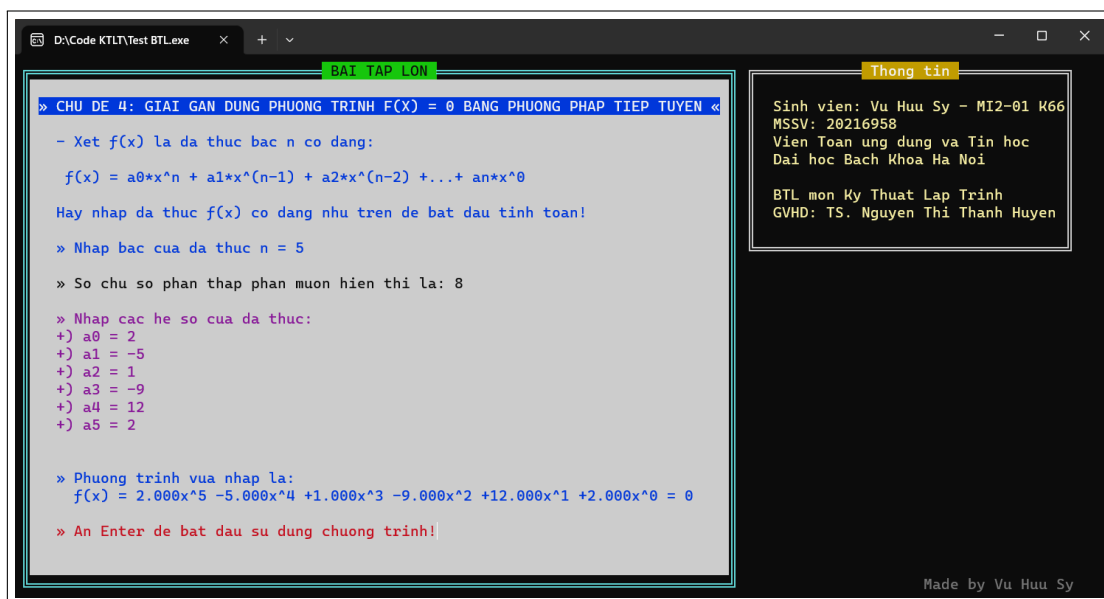
```
819         chucnang_5();
820         getch();
821         menu();
822         break;
823     case 6:
824         khunghuongdan();
825         getch();
826         menu();
827         break;
828     case 7:
829         khungtieptuyen();
830         getch();
831         menu();
832         break;
833     case 8:
834         khungexit();
835     exit(0);
836 }
837 }
838
839 int main()
840 {
841     input();
842     menu();
843     khungexit();
844     return 0;
845 }
```

Chương 4

Hình ảnh giao diện thực hiện chương trình và nội dung file text là kết quả ra của chương trình

4.1 Giao diện nhập vào vào đa thức

- Nhập vào bậc của đa thức.
- Nhập số chữ số phần thập phân muốn hiển thị ở kết quả.
- Nhập các hệ số của đa thức.



- File **input.txt** lưu đầu vào người dùng nhập.

```

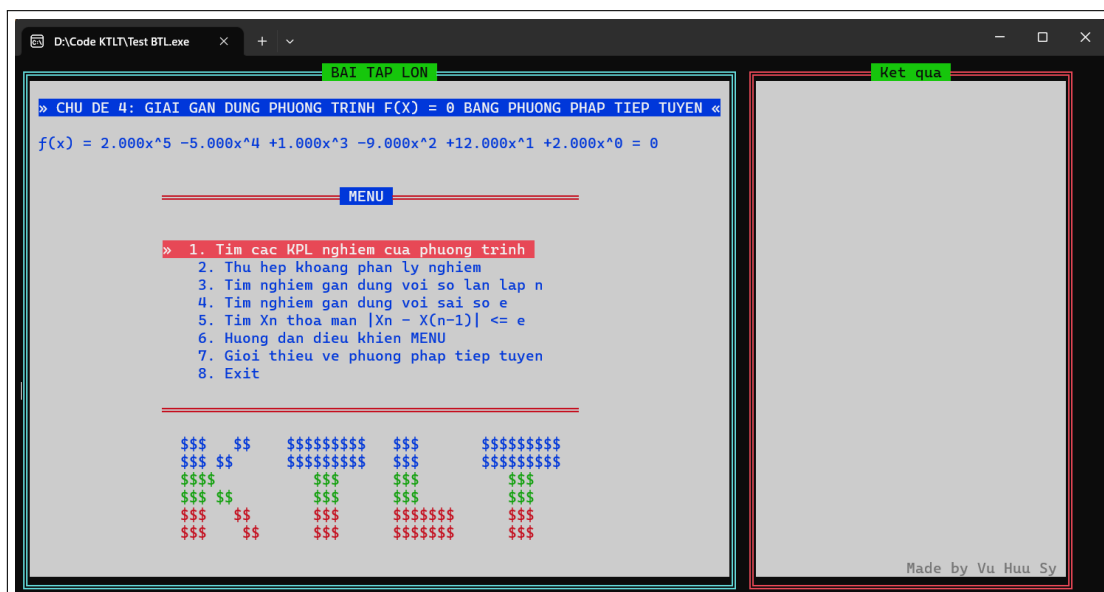
Ngươi dung nhap vao:
- Bac cua da thuc: 5.
- So chu so phan thap phan mun hien thi la: 8.

- Cac he so cua da thuc:
+> a0 = 2.00000000
+> a1 = -5.00000000
+> a2 = 1.00000000
+> a3 = -9.00000000
+> a4 = 12.00000000
+> a5 = 2.00000000

Phuong trinh: F(x) = 2.00000000x^5 -5.00000000x^4 +1.00000000x^3
-9.00000000x^2 +12.00000000x^1 +2.00000000x^0 = 0
    
```

4.2 Giao diện MENU chính

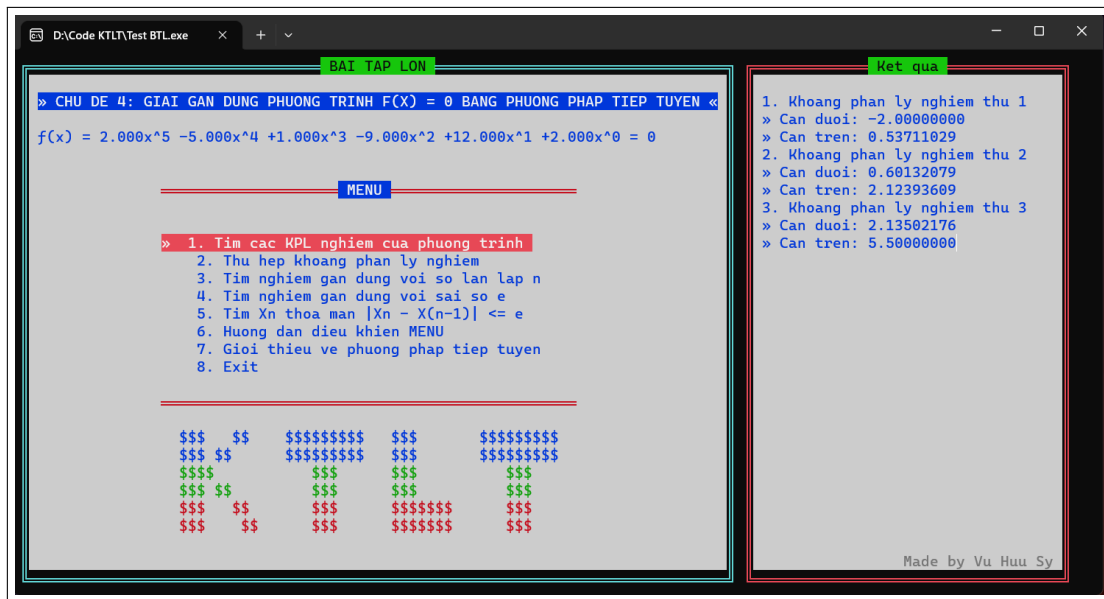
- Gồm MENU chính và khung hiển thị kết quả tính toán ở bên phải.



4.3 Giao diện thực hiện chức năng 1: Tìm các khoảng phân ly nghiệm.

- INPUT: Phương trình $f(x) = 0$.
- OUTPUT: Các khoảng phân ly nghiệm của phương trình.
- Giao diện các kết quả:

1. Trường hợp phương trình có nghiệm.



File **chucnang_1.txt** lưu kết quả:

```

chucnang_1
File Edit View

Chuc nang 1: Tim cac khoang phan ly nghiêm cua phuong trinh.

INPUT: Bac cua phuong trinh, cac he so cua phuong trinh  $f(x) = 0$ .
OUTPUT: Cac khoang phan ly nghiêm cua phuong trinh.

Phuong trinh:  $F(x) = 2.00000000x^5 - 5.00000000x^4 + 1.00000000x^3 - 9.00000000x^2 + 12.00000000x^1 + 2.00000000x^0 = 0$ 

Khoang phan ly nghiêm thu 1
- Can duoi: -2.00000000
- Can tren: 0.53711029

Khoang phan ly nghiêm thu 2
- Can duoi: 0.60132079
- Can tren: 2.12393609

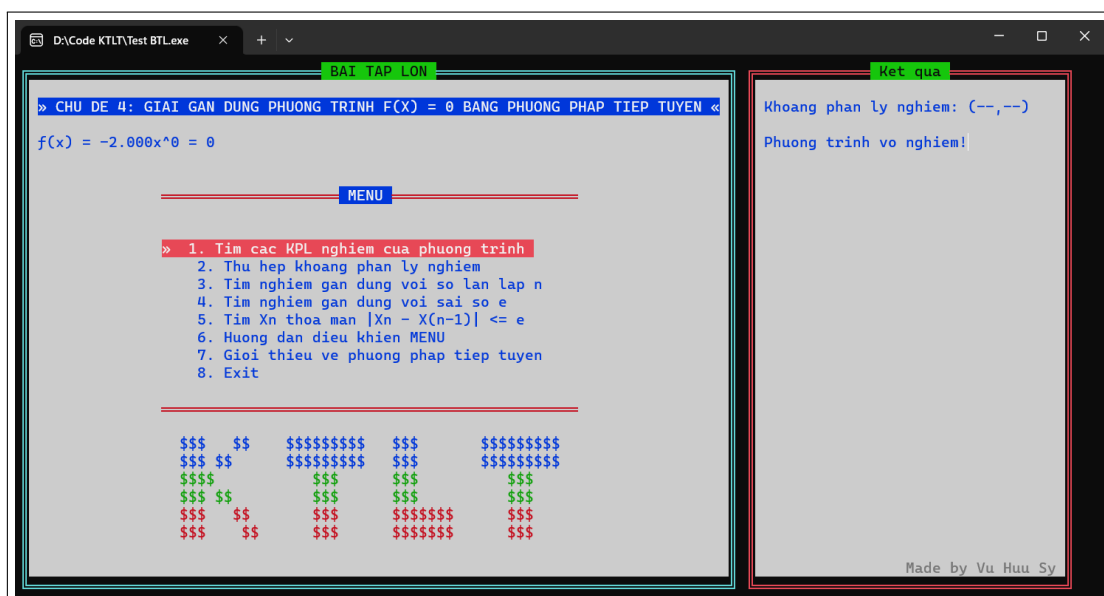
Khoang phan ly nghiêm thu 3
- Can duoi: 2.13502176
- Can tren: 5.50000000

Ket thuc.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

```

2. Trường hợp phương trình vô nghiệm.



File **chucnang_1.txt** lưu kết quả:

```

chucnang_1
File Edit View
Chuc nang 1: Tim cac khoang phan ly nghiêm cua phuong trinh.

INPUT: Bac cua phuong trinh, cac he so cua phuong trinh f(x) =
0.
OUTPUT: Cac khoang phan ly nghiêm cua phuong trinh.

Phuong trinh: F(x) = -2.00000000x^0 = 0

Khoang phan ly nghiêm: (--,--)

Phuong trinh vo nghiêm!

Ket thuc.

```

3. Trường hợp phương trình có vô số nghiệm.

```

D:\Code KTLT\Test BT\Lexa
BAI TAP LON
>> CHU DE 4: GIAI GAN DUNG PHUONG TRINH F(X) = 0 BANG PHUONG PHAP TIEP TUYEN <<
f(x) = 0.000x^0 = 0

MENU

» 1. Tim cac KPL nghiêm cua phuong trinh
2. Thu hep khoang phan ly nghiêm
3. Tim nghiêm gan dung voi so lan lap n
4. Tim nghiêm gan dung voi sai so e
5. Tim Xn thoa man |Xn - X(n-1)| <= e
6. Huong dan dieu khien MENU
7. Gioi thieu ve phuong phap tiep tuyen
8. Exit

$$$ $$$ $$$$$$$$ $$$ $$$$$$$$
$$$ $$$ $$$$$$$$ $$$ $$$$$$$$
$$$$$ $$$ $$$$ $$$$
$$$$$ $$$ $$$$ $$$$
$$$ $$$ $$$$ $$$$$$$$ $$$
$$$ $$$ $$$$ $$$$$$$$ $$$

Khoang phan ly nghiêm: (-00,+00)
Phuong trinh co vo so nghiêm!

Made by Vu Huu Sy

```

File **chucnang_1.txt** lưu kết quả:

```

chucnang_1
File Edit View
Chuc nang 1: Tim cac khoang phan ly nghiêm cua phuong trinh.

INPUT: Bac cua phuong trinh, cac he so cua phuong trinh f(x) =
0.
OUTPUT: Cac khoang phan ly nghiêm cua phuong trinh.

Phuong trinh: F(x) = 0.00000000x^0 = 0

Khoang phan ly nghiêm: (-oo,+oo)

Phuong trinh co vo so nghiêm!

Ket thuc.

```

4.4 Giao diện thực hiện chức năng 2: Rút gọn khoảng phân ly nghiệm tìm được ở chức năng 1

- INPUT: các khoảng phân ly tìm được ở chức năng 1, lựa chọn của người dùng.
- OUTPUT: KPL nghiệm được rút gọn sử dụng phương pháp chia đôi.
- Giao diện các kết quả.

1. Trường hợp phương trình có KPL nghiệm.

```

D:\Code KTL\Test BT\Lexa
BÀI TẬP LỚN
» CHU DE 4: GIAI GAN DUNG PHUONG TRINH F(X) = 0 BANG PHUONG PHAP TIEP TUYEN «
f(x) = 2.000x^5 -5.000x^4 +1.000x^3 -9.000x^2 +12.000x^1 +2.000x^0 = 0

CHUC NANG 2

2. Thu hep khoang phan ly nghiêm (a,b) thoa man
|a - b| <= 0.5 su dung phuong phap chia doi
(Nhap so nguyên dương nhỏ hơn 4!)

Chon khoang phan ly nghiêm thu: 3

Khoang phan ly ban dau la:
Can duoi a: 2.13502176
Can tren b: 5.50000000

Khoang phan ly sau khi rut gon:
Can duoi a: 2.55564404
Can tren b: 2.97626632

|a - b| = 0.42062228 <= 0.5

So lan lap chia doi: 3

Made by Vu Huu Sy

```

File **chucnang_2.txt** lưu kết quả:

```

chucnang_2
File Edit View
Chuc nang 2: Thu hep khoang phan ly nghiem (a,b) thoa man |a-b| <= 0.5 bang cach su dung phuong phap chia doi

Phuong trinh: F(x) = 2.00000000x^5 -5.00000000x^4 +1.00000000x^3 -9.00000000x^2 +12.00000000x^1 +2.00000000x^0 = 0

INPUT: cac KPL nghiem tim duoc o chuc nang 1, lua chon cua nguoi dung.
OUTPUT: KPL nghiem duoc rut gon thoa man |a - b| <= 0.5.
Khoang phan ly nguoi dung chon: 1

Khoang phan ly ban dau la:
- Can duoi a: -2.00000000
- Can tren b: 0.53711029

Su dung phuong phap chia doi de thu hep khoang cach ly:

Lan lap thu 1
- Can duoi a: -0.73144486
- Can tren b: 0.53711029

Lan lap thu 2
- Can duoi a: -0.73144486
- Can tren b: -0.09716728

Lan lap thu 3
- Can duoi a: -0.41430607
- Can tren b: -0.09716728

Khoang phan ly nghiem sau khi rut gon:
- Can duoi a: -0.41430607
- Can tren b: -0.09716728

|a - b| = 0.31713879 <= 0.5 thoa man de bai.

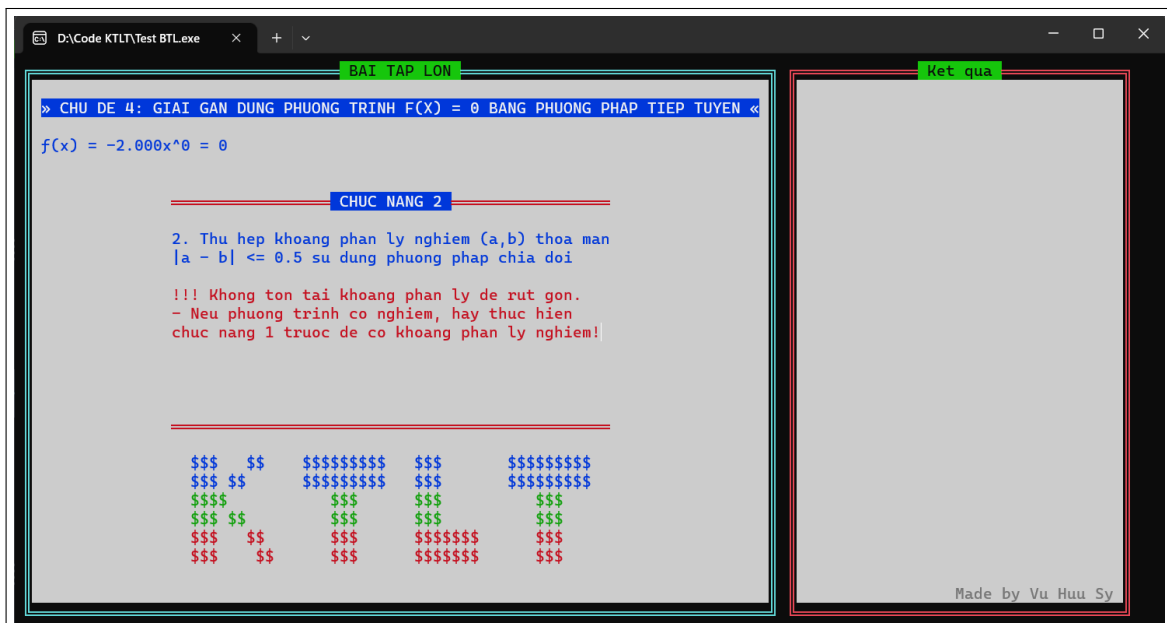
So lan lap chia doi: 3

Ket thuc.
Ln 16, Col 56
100% Windows (CRLF) UTF-8

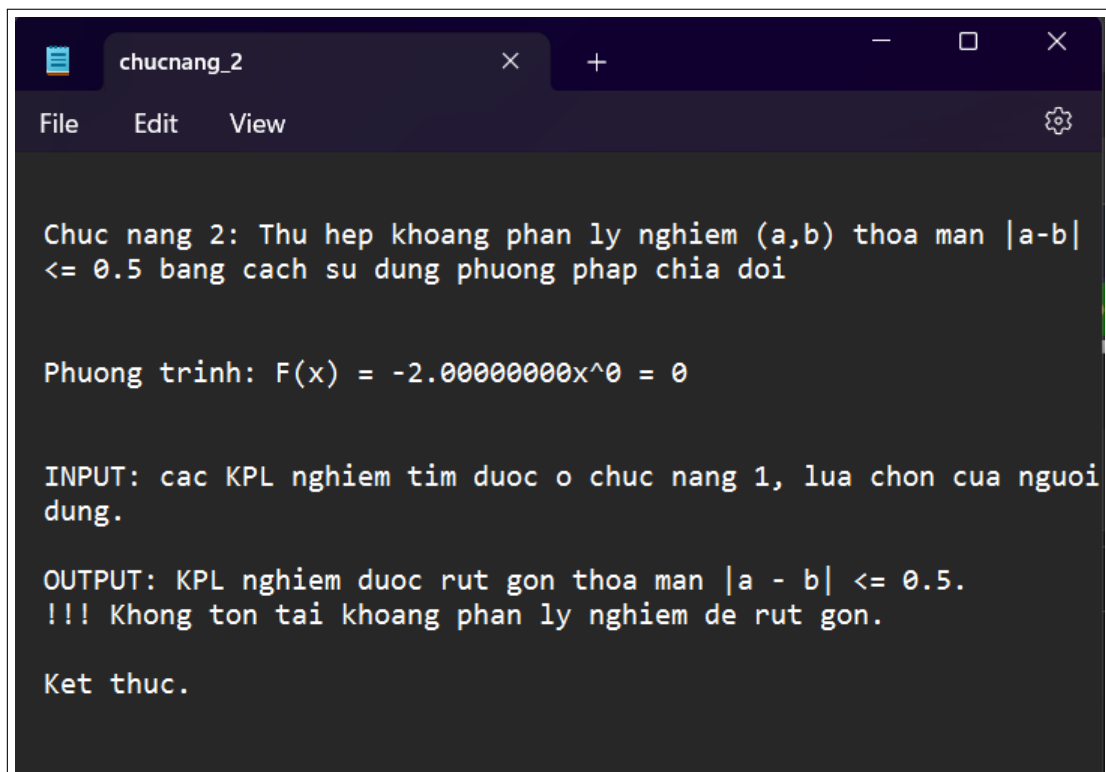
```

2. Trường hợp phương trình vô nghiệm / vô số nghiệm.

Vì phương trình vô nghiệm / vô số nghiệm nên chương trình sẽ thông báo cho người dùng rằng không tồn tại khoảng phân ly để rút gọn.



File **chucnang__2.txt** lưu kết quả:



```
chucnang_2
File Edit View
Chuc nang 2: Thu hep khoang phan ly nghiem (a,b) thoa man |a-b|
<= 0.5 bang cach su dung phuong phap chia doi

Phuong trinh: F(x) = -2.00000000x^0 = 0

INPUT: cac KPL nghiem tim duoc o chuc nang 1, lua chon cua nguoi
dung.

OUTPUT: KPL nghiem duoc rut gon thoa man |a - b| <= 0.5.
!!! Khong ton tai khoang phan ly nghiem de rut gon.

Ket thuc.
```

4.5 Giao diện thực hiện chức năng 3: Tìm nghiệm gần đúng với số lần lặp n cho trước

- INPUT: Khoảng phân ly nghiệm (a,b), số lần lặp n.
- OUTPUT: Nghiệm gần đúng x_n , đánh giá sai số.
- Giao diện các kết quả.

1. Trường hợp phương trình có nghiệm.

Chương trình sẽ đưa ra đánh giá sai số dựa trên kết quả tính toán.

```

>> CHU DE 4: GIAI GAN DUNG PHUONG TRINH F(X) = 0 BANG PHUONG PHAP TIEP TUYEN <<
f(x) = 2.000x^5 - 5.000x^4 + 1.000x^3 - 9.000x^2 + 12.000x^1 + 2.000x^0 = 0

CHUC NANG 3

» Nhap khoang phan ly nghiem (a,b):
» Nhap can duoi a: -0.5
» Nhap can tren b: 0
» Nhap so lan lap: n = 6

» Danh gia sai so:
- Phuong phap tiep tuyen co toc do hoi tu cao
va ket qua co do chinh xac cao.
- Cong thuc sai so 1 co sai so nho hon so voi
cong thuc sai so 2 o nhung lan lap dau.

$$$  $$$  $$$$$$$$  $$$  $$$$$$$$
$$$  $$$  $$$$$$$$  $$$  $$$$$$$$
$$$$$  $$$  $$$  $$$  $$$
$$$  $$$  $$$  $$$  $$$
$$$  $$$  $$$  $$$  $$$

» Diem Fourier la:
x0 = -0.50000000

Lan lap 1: x1 = -0.22864322
- Sai so 1 = 0.103416040949498
- Sai so 2 = 0.125886670235920

Lan lap 2: x2 = -0.15360237
- Sai so 1 = 0.005178774084196
- Sai so 2 = 0.009627064049521

Lan lap 3: x3 = -0.14943505
- Sai so 1 = 0.000014792658508
- Sai so 2 = 0.000029690042380

Lan lap 4: x4 = -0.14942308
- Sai so 1 = 0.000000000121638
- Sai so 2 = 0.000000000245031

Lan lap 5: x5 = -0.14942308
- Sai so 1 = 0.000000000000000
- Sai so 2 = 0.000000000000000

Lan lap 6: x6 = -0.14942308
- Sai so 1 = 0.000000000000000
- Sai so 2 = 0.000000000000000
Made by Vu Huu Sy
    
```

File **chucnang_3.txt** lưu kết quả:

```

chucnang_3

File Edit View

Chuc nang 3: Tim nghiem gan dung voi so lan lap cho truoc trong khoang (a, b) va danh gia sai so epsilon theo hai
cong thuc sai so.

Phuong trinh: F(x) = 2.00000000x^5 - 5.00000000x^4 + 1.00000000x^3 - 9.00000000x^2 + 12.00000000x^1 + 2.00000000x^0 = 0

INPUT: KPL nghiem (a,b), so lan lap.
OUTPUT: Nghiem gan dung Xn, sai so epsilon.

Khoang phan ly nghiem (a,b):
- Can duoi a: -0.50000000
- Can tren b: 0.00000000

So lan lap nguoi dung nhap la: 6

- Gia tri nho nhat cua |f'(x)| la: m1 = 12.00090001
- Gia tri lon nhat cua |f''(x)| la: M2 = 41.03384299

Diem Fourier la: x0 = -0.50000000

Lan lap thu 1:
Nghiem xap xi x1 = -0.22864322
Sai so muc tieu = 0.10341604094949765000
Sai so hai xap xi lien tiep = 0.12588667023592004000

Lan lap thu 2:
Nghiem xap xi x2 = -0.15360237
Sai so muc tieu = 0.00517877408419638390
Sai so hai xap xi lien tiep = 0.00962706404952146620

Lan lap thu 3:
Nghiem xap xi x3 = -0.14943505
Sai so muc tieu = 0.00001479265850805040
Sai so hai xap xi lien tiep = 0.00002969004237962061
    
```

```

Lan lap thu 4:
Nghiem xap xi x4 = -0.14942308
Sai so muc tieu      = 0.00000000012163779185
Sai so hai xap xi lien tiep = 0.00000000024503103091

Lan lap thu 5:
Nghiem xap xi x5 = -0.14942308
Sai so muc tieu      = 0.00000000000000003700
Sai so hai xap xi lien tiep = 0.0000000000000000002

Lan lap thu 6:
Nghiem xap xi x6 = -0.14942308
Sai so muc tieu      = 0.00000000000000000000
Sai so hai xap xi lien tiep = 0.00000000000000000000

Danh gia sai so cua hai cong thuc:
- Uu diem: Toc do hoi tu cao, ket qua co do chinh xac cao.
- Nhuoc diem: Dieu kien lua chon x0 ban dau kha chat.
- Cong thuc sai so 1 co sai so nho hon so voi cong thuc sai so 2 o nhung lan lap dau.

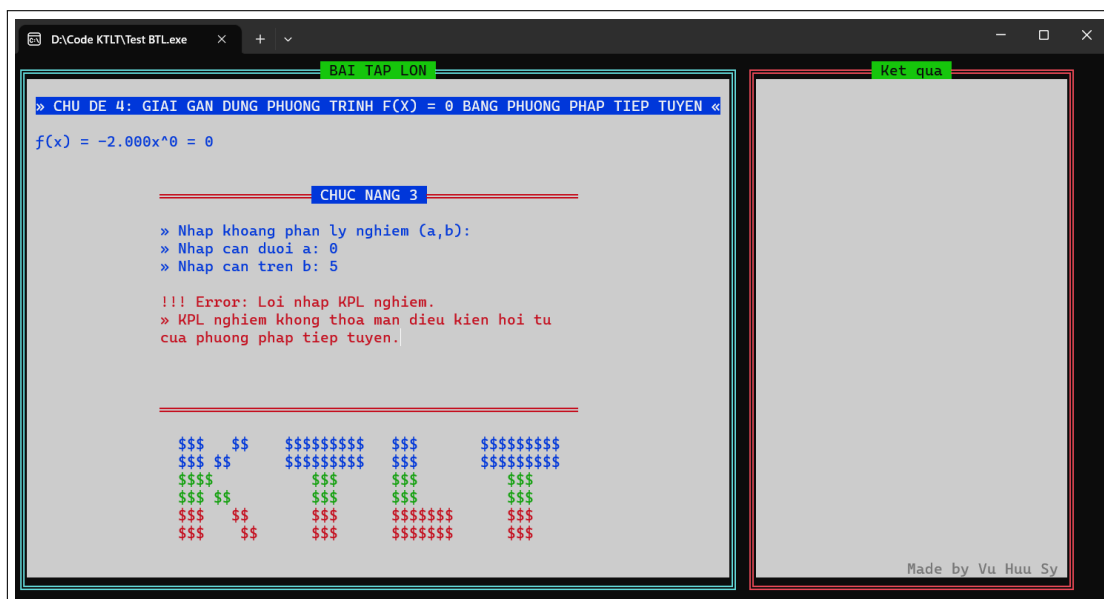
Ghi chu: Vi phuong phap tien tuyen hoi tu nhanh va co sai so rat nho gan 0 nen sai so em se de nhieu chu so phan
thap phan hon.

Ket thuc.

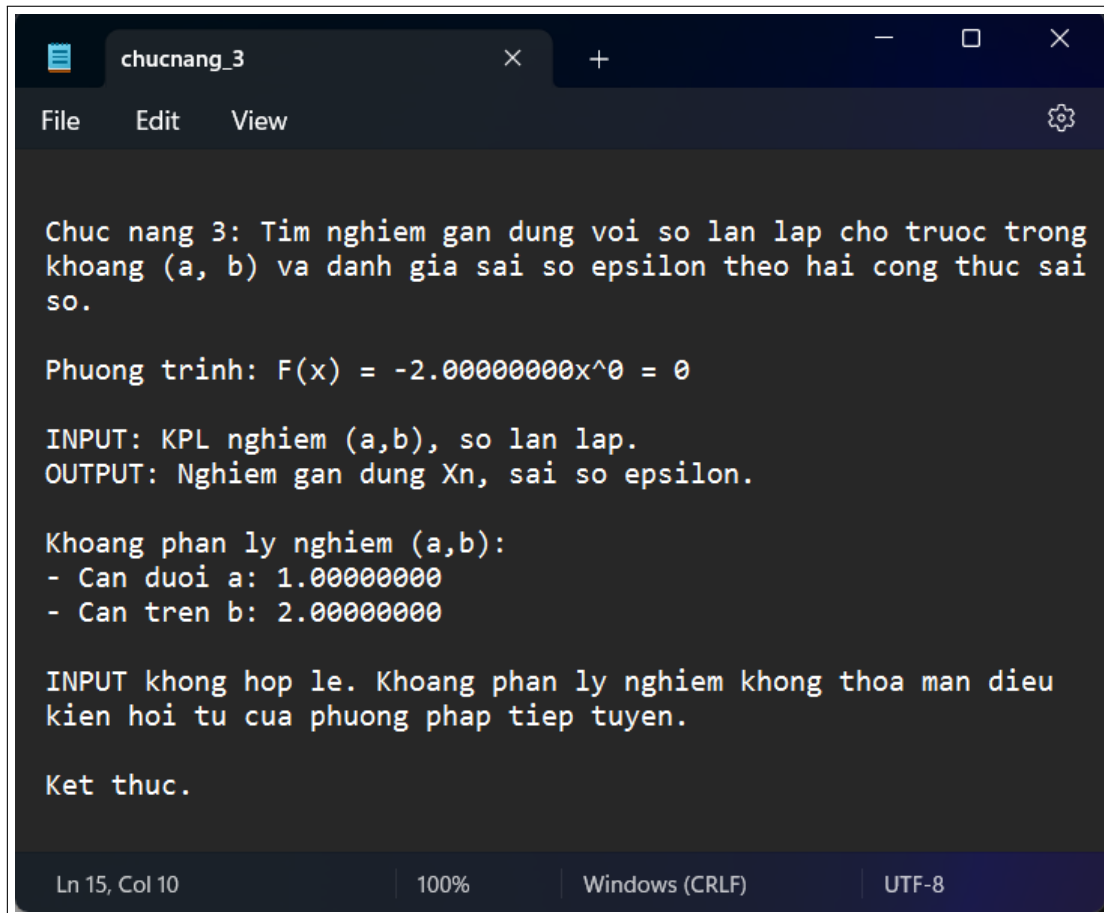
```

2. Trường hợp phương trình vô nghiệm/vô số nghiệm .

Vì phương trình vô nghiệm nên không có khoảng phân ly nào thỏa mãn điều kiện hội tụ của phương pháp tiếp tuyến. Chương trình sẽ báo lỗi như bên dưới.



File **chucnang_3.txt** lưu kết quả:



```

chucnang_3
File Edit View
Chuc nang 3: Tim nghiem gan dung voi so lan lap cho truoac trong
khoang (a, b) va danh gia sai so epsilon theo hai cong thuc sai
so.

Phuong trinh: F(x) = -2.00000000x^0 = 0

INPUT: KPL nghiem (a,b), so lan lap.
OUTPUT: Nghiem gan dung Xn, sai so epsilon.

Khoang phan ly nghiem (a,b):
- Can duoi a: 1.00000000
- Can tren b: 2.00000000

INPUT khong hop le. Khoang phan ly nghiem khong thoa man dieu
kien hoi tu cua phuong phap tiep tuyen.

Ket thuc.

Ln 15, Col 10 | 100% | Windows (CRLF) | UTF-8
  
```

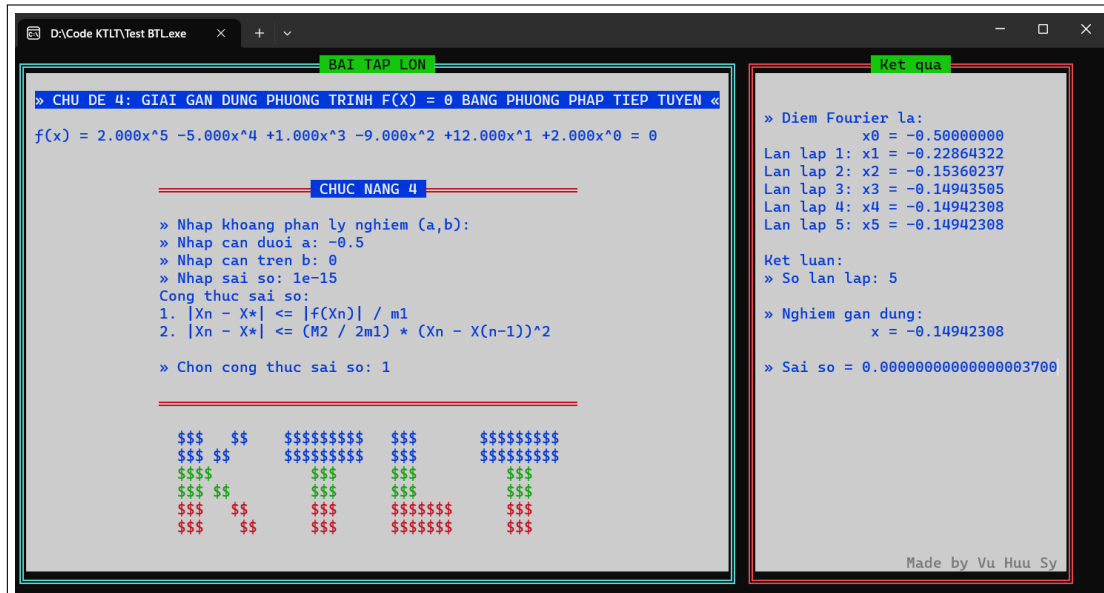
4.6 Giao diện thực hiện chức năng 4: Tìm nghiệm gần đúng với sai số ϵ cho trước

- INPUT: Khoảng phân ly nghiệm (a,b), sai số ϵ .
- OUTPUT: Nghiệm gần đúng x_n thỏa mãn sai số nhập vào từ người dùng.
- Giao diện các kết quả.

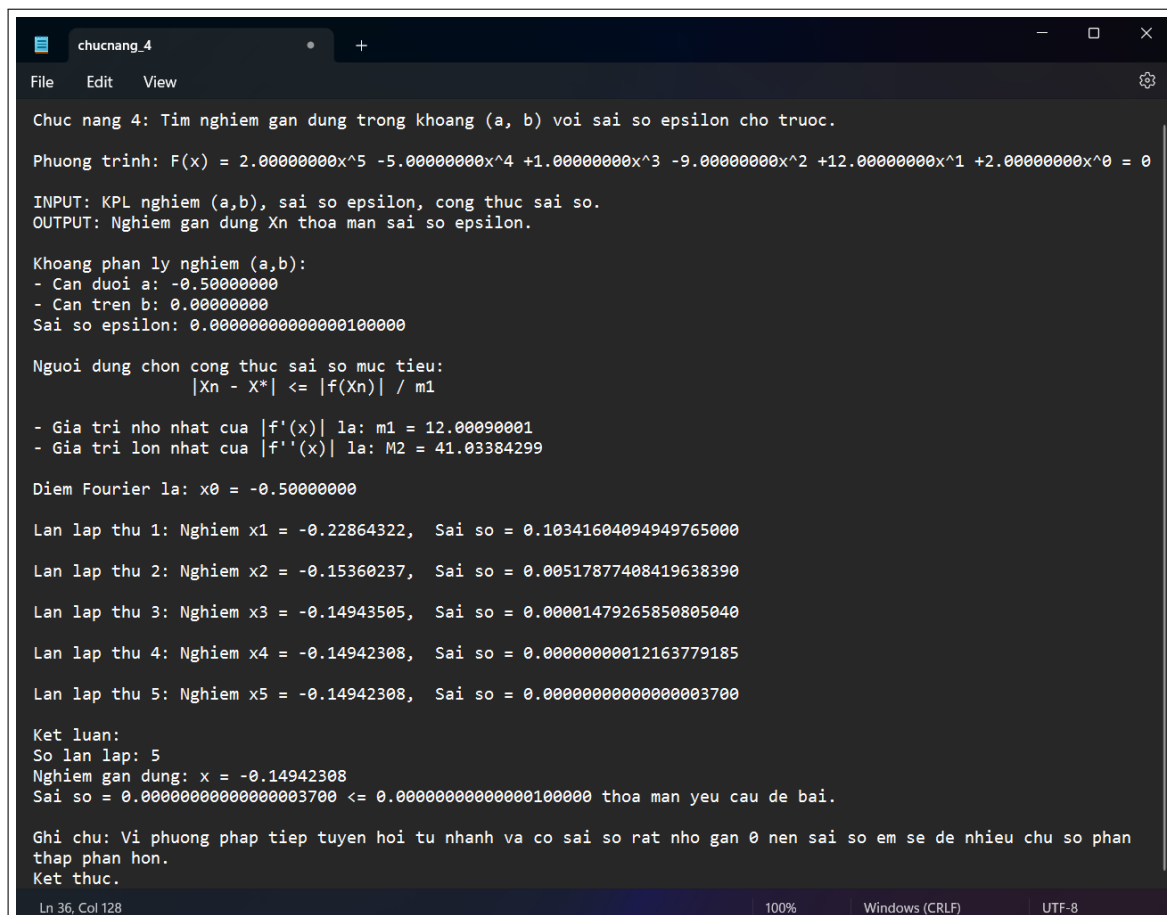
1. Trường hợp phương trình có nghiệm.

Chương trình sẽ đánh giá sai số dựa trên công thức sai số mà người dùng chọn.

Công thức sai số 1:



File `chucnang_4.txt` lưu kết quả:



Công thức sai số 2:

```

>> CHU DE 4: GIAI GAN DUNG PHUONG TRINH F(X) = 0 BANG PHUONG PHAP TIEP TUYEN <<
f(x) = 2.000x^5 -5.000x^4 +1.000x^3 -9.000x^2 +12.000x^1 +2.000x^0 = 0

=====
CHUC NANG 4
=====
>> Nhap khoang phan ly nghieng (a,b):
>> Nhap can duoi a: -0.5
>> Nhap can tren b: 0
>> Nhap sai so: 1e-15
Cong thuc sai so:
1. |Xn - X*| <= |f(Xn)| / m1
2. |Xn - X*| <= (M2 / 2m1) * (Xn - X(n-1))^2

>> Chon cong thuc sai so: 2

=====
$$$  $$$  $$$$$$$$  $$$  $$$$$$$$
$$$  $$$  $$$$$$$$  $$$  $$$$$$$$
$$$$$  $$$  $$$  $$$  $$$
$$$  $$$  $$$  $$$  $$$
$$$  $$$  $$$  $$$  $$$
$$$  $$$  $$$  $$$  $$$

>> Diem Fourier la:
      x0 = -0.50000000
Lan lap 1: x1 = -0.22864322
Lan lap 2: x2 = -0.15360237
Lan lap 3: x3 = -0.14943505
Lan lap 4: x4 = -0.14942308
Lan lap 5: x5 = -0.14942308

Ket luan:
>> So lan lap: 5

>> Nghieng gan dung:
      x = -0.14942308

>> Sai so = 0.0000000000000000002

Made by Vu Huu Sy
    
```

File chucnang_4.txt lưu kết quả:

```

chucnang_4
File Edit View
Chuc nang 4: Tim nghieng gan dung trong khoang (a, b) voi sai so epsilon cho truoc.

Phuong trinh: F(x) = 2.0000000x^5 -5.0000000x^4 +1.0000000x^3 -9.0000000x^2 +12.0000000x^1 +2.0000000x^0 = 0

INPUT: KPL nghieng (a,b), sai so epsilon, cong thuc sai so.
OUTPUT: Nghieng gan dung Xn thoa man sai so epsilon.

Khoang phan ly nghieng (a,b):
- Can duoi a: -0.50000000
- Can tren b: 0.00000000
Sai so epsilon: 0.00000000000000000000

Nguoi dung chon cong thuc sai so theo hai xap xi lien tiep:
|Xn - X*| <= (M2 / 2m1) * (Xn - X(n-1))^2

- Gia tri nho nhat cua |f'(x)| la: m1 = 12.00090001
- Gia tri lon nhat cua |f''(x)| la: M2 = 41.03384299

Diem Fourier la: x0 = -0.50000000

Lan lap thu 1: Nghieng x1 = -0.22864322, Sai so = 0.12588667023592004000
Lan lap thu 2: Nghieng x2 = -0.15360237, Sai so = 0.00962706404952146620
Lan lap thu 3: Nghieng x3 = -0.14943505, Sai so = 0.00002969004237962061
Lan lap thu 4: Nghieng x4 = -0.14942308, Sai so = 0.0000000024503103091
Lan lap thu 5: Nghieng x5 = -0.14942308, Sai so = 0.0000000000000000002

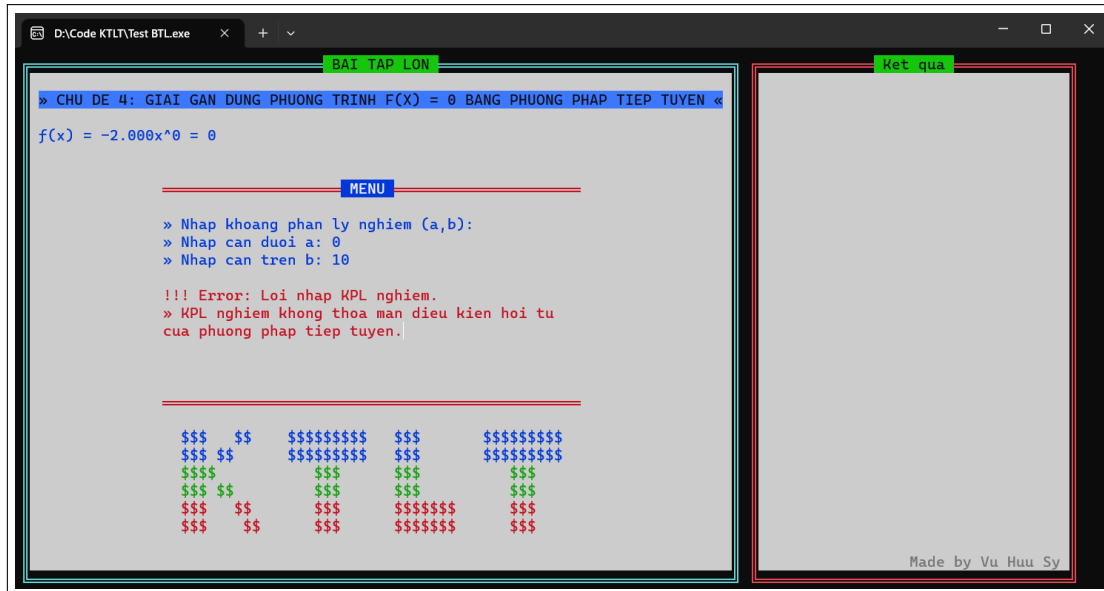
Ket luan:
So lan lap: 5
Nghieng gan dung: x = -0.14942308
Sai so = 0.0000000000000000002 <= 0.00000000000000000000 thoa man yeu cau de bai.

Ghi chu: Vi phuong phap tiep tuyen hoi tu nhanh va co sai so rat nho gan 0 nen sai so em se de nhieu chu so phan
thap phan hon.
Ket thuc.

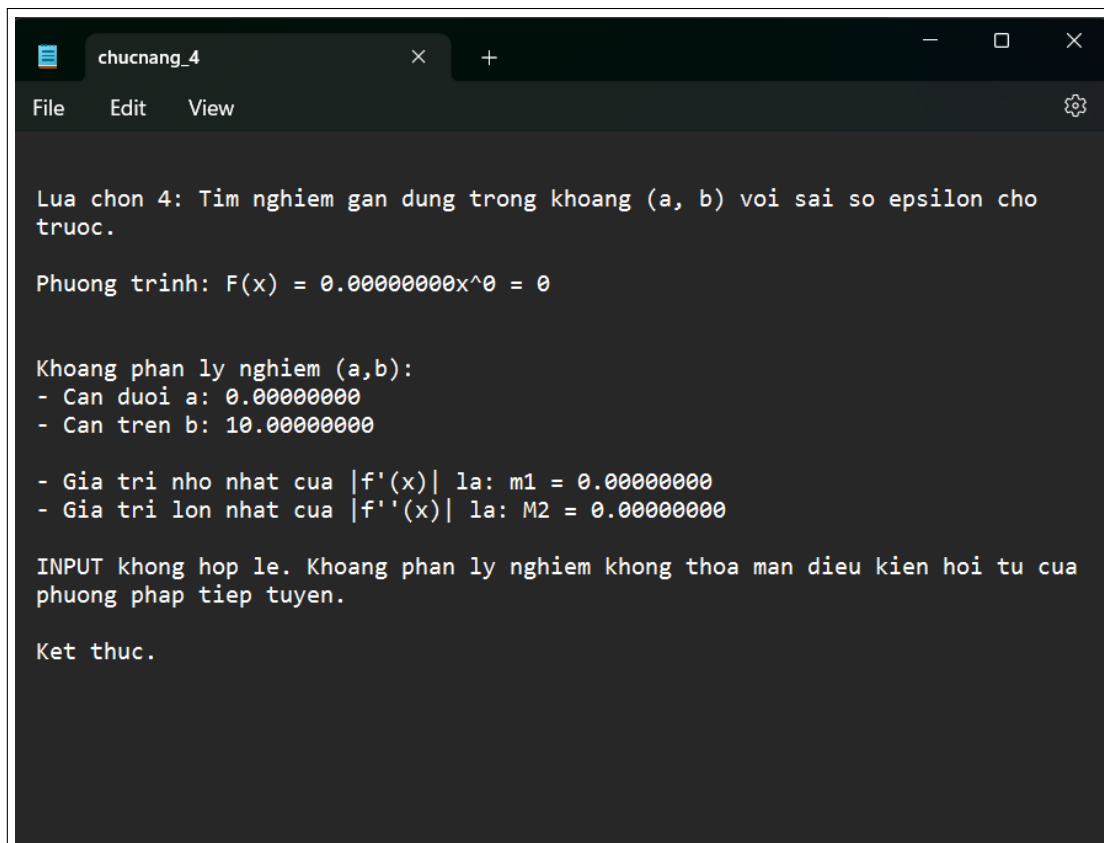
Ln 36, Col 128 100% Windows (CRLF) UTF-8
    
```

2. Trường hợp phương trình vô nghiệm / vô số nghiệm.

Vì phương trình vô nghiệm nên không có khoảng phân ly nào thỏa mãn điều kiện hội tụ của phương pháp tiếp tuyến. Chương trình sẽ báo lỗi như bên dưới.

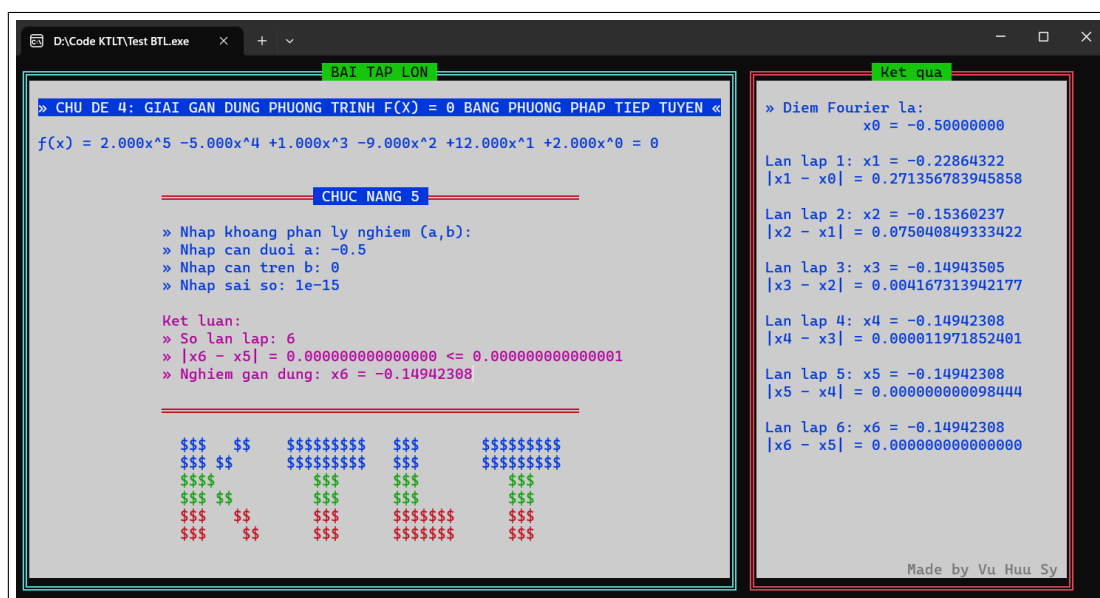


File **chucnang__4.txt** lưu kết quả:



- INPUT: Khoảng phân ly nghiệm (a,b), sai số ε .
- OUTPUT: Nghiệm gần đúng x_n thỏa mãn điều kiện $|x_n - x_{n-1}| \leq \varepsilon$, số lần lặp.
- Giao diện các kết quả.
 1. Trường hợp phương trình có nghiệm.

Chương trình sẽ tìm điểm Fourier và gán vào x_0 , sau đó bắt đầu thực hiện thao tác lặp cho đến khi thỏa mãn điều kiện của đề bài.



File **chucnang_5.txt** lưu kết quả:

```

chucnang_5
File Edit View
Chuc nang 5: Tim nghiem gan dung Xn trong khoang (a, b) thoa man dieu kien: |Xn - X(n-1)| <= e.

Phuong trinh: F(x) = 2.00000000x^5 - 5.00000000x^4 + 1.00000000x^3 - 9.00000000x^2 + 12.00000000x^1 + 2.00000000x^0 = 0
INPUT: KPL nghiem (a,b), sai so epsilon
OUTPUT: Nghiem gan dung Xn thoa man |Xn - X(n-1)| <= e.

Khoang phan ly nghiem (a,b):
- Can duoi a: -0.50000000
- Can tren b: 0.00000000

Sai so epsilon: 0.0000000000000100000

Diem Fourier la: x0 = -0.50000000

Lan lap thu 1:  x1 = -0.22864322
- |x1 - x0| = 0.27135678394585777000

Lan lap thu 2:  x2 = -0.15360237
- |x2 - x1| = 0.07504084933342211600

Lan lap thu 3:  x3 = -0.14943505
- |x3 - x2| = 0.00416731394217739750

Lan lap thu 4:  x4 = -0.14942308
- |x4 - x3| = 0.00001197185240142673

Lan lap thu 5:  x5 = -0.14942308
- |x5 - x4| = 0.00000000009844433602

Lan lap thu 6:  x6 = -0.14942308
- |x6 - x5| = 0.0000000000000002776

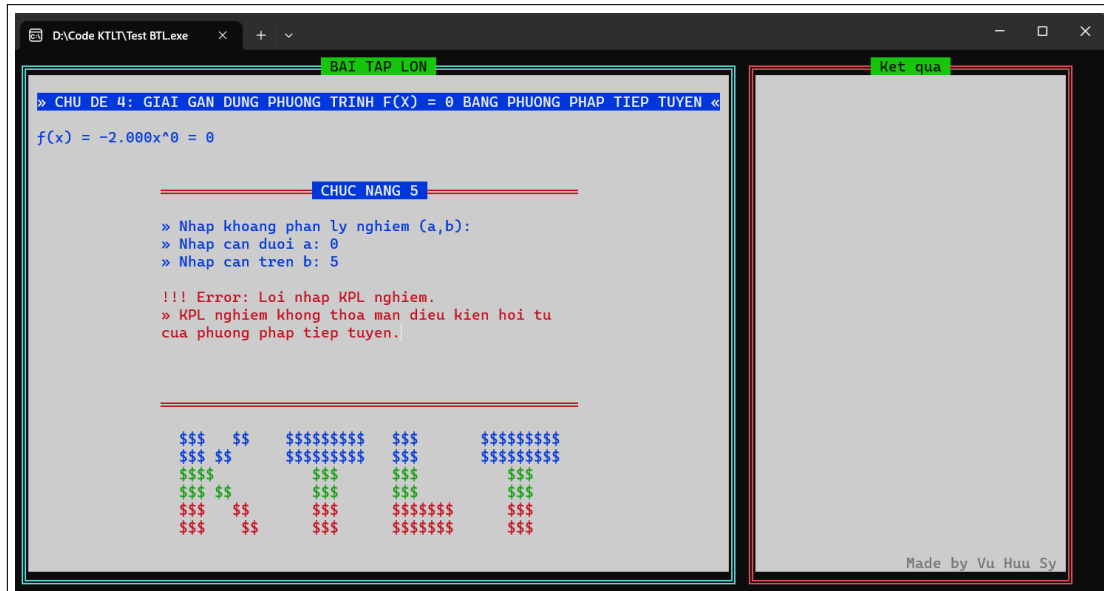
Ket luan:
So lan lap: 6
Nghiem gan dung: x6 = -0.14942308
Sai so |x6 - x5| = 0.0000000000000002776 <= 0.0000000000000100000 thoa man yeu cau de bai.

Ket thuc.
Ln 3, Col 115      100%      Windows (CRLF)      UTF-8

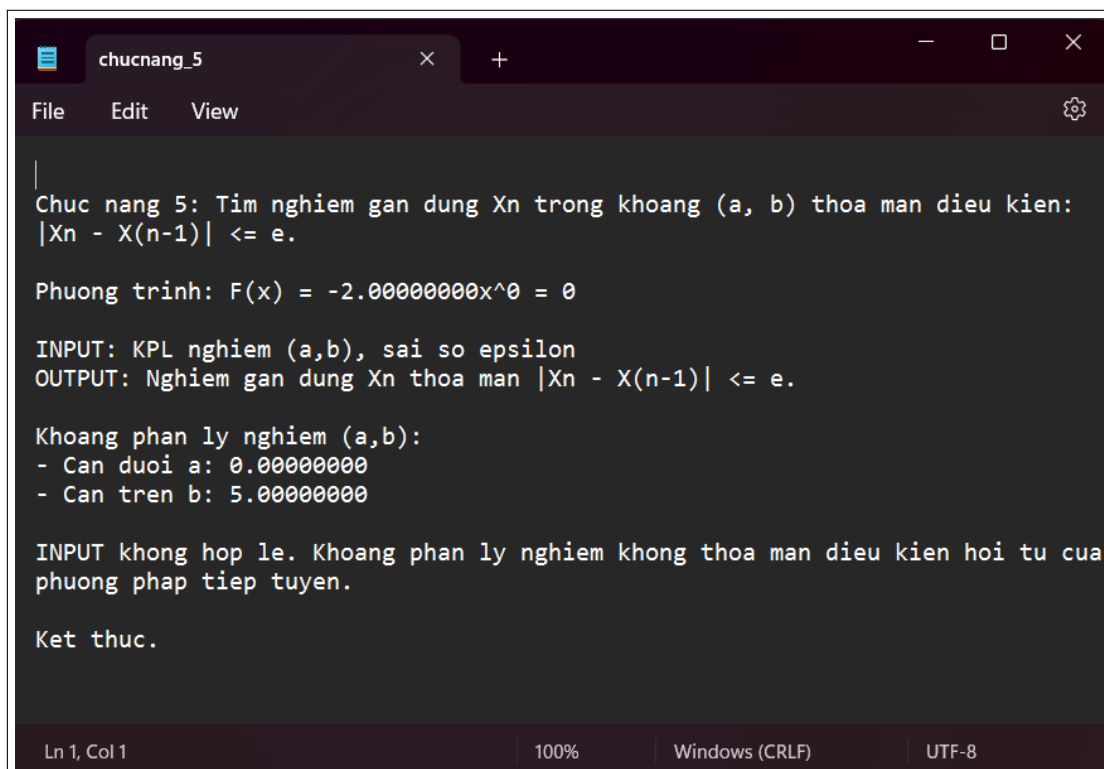
```

2. Trường hợp phương trình vô nghiệm / vô số nghiệm.

Vì phương trình vô nghiệm / vô số nghiệm nên không có khoảng phân ly nào thỏa mãn điều kiện hội tụ của phương pháp tiếp tuyến. Chương trình sẽ báo lỗi như bên dưới.



File `chucnang_5.txt` lưu kết quả:



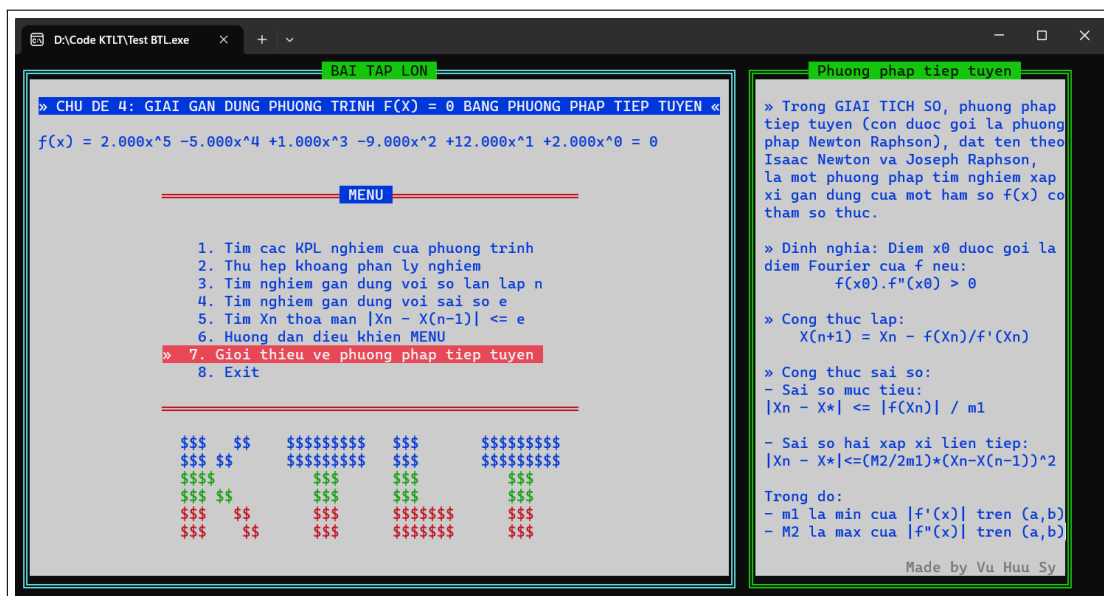
4.8 Giao diện thực hiện chức năng 6: Hướng dẫn sử dụng MENU

Chức năng này để hướng dẫn người dùng sử dụng MENU đúng cách.



4.9 Giao diện thực hiện chức năng 7: Giới thiệu về phương pháp tiếp tuyến

Chức năng này để là giới thiệu cho người dùng một số thông tin về phương pháp tiếp tuyến.



4.10 Giao diện thực hiện chức năng 8: Exit

Chức năng này để kết thúc chương trình (hoặc người dùng có thể ấn phím Esc để thoát).



Chương 5

Những tính năng và đồ họa của chương trình

5.1 Bắt phím người dùng nhập vào và xử lý phím đó để điều khiển MENU

Đây là đoạn mã để xử lý phím người dùng nhập vào khi điều khiển MENU:

```
1 void arrowHere(int realPosition, int arrowPosition)
2 {
3     if (realPosition == arrowPosition)
4     {
5         TextColor(207); printf("%c ", 175);
6     }
7     else {
8         TextColor(113); printf(" ");
9     }
10 }
11
12 void menu()
13 {
14     int position = 1;
15     int keyPress = 0;
16     khung();
17     while (keyPress != 13) {
18         for (int i = 1; i <= MAX; i++) {
19             gotoxy(16, 9 + i);
```

```

20     arrowHere(i, position);
21     printf("%d. %s \n", i, option[i - 1]);
22 }
23 keyPress = getch();
24 if (keyPress == 80 || keyPress == 's' || keyPress == 'S' || keyPress ==
↪ '2') {
25     if (position == MAX) position = MIN;
26     else position++;
27 }
28 else if (keyPress == 72 || keyPress == 'w' || keyPress == 'W' ||
↪ keyPress == '8') {
29     if (position == MIN ) position = MAX;
30     else position--;
31 }
32 else if (keyPress == 9) {
33     position += 2;
34     if (position > MAX) position = MIN;
35 }
36 else if (keyPress == 27) {
37     system("cls");
38     return;
39 }
40 }
41 switch (position) {
42     // doan ma xu ly chuc nang dang chon
43 }
44 }

```

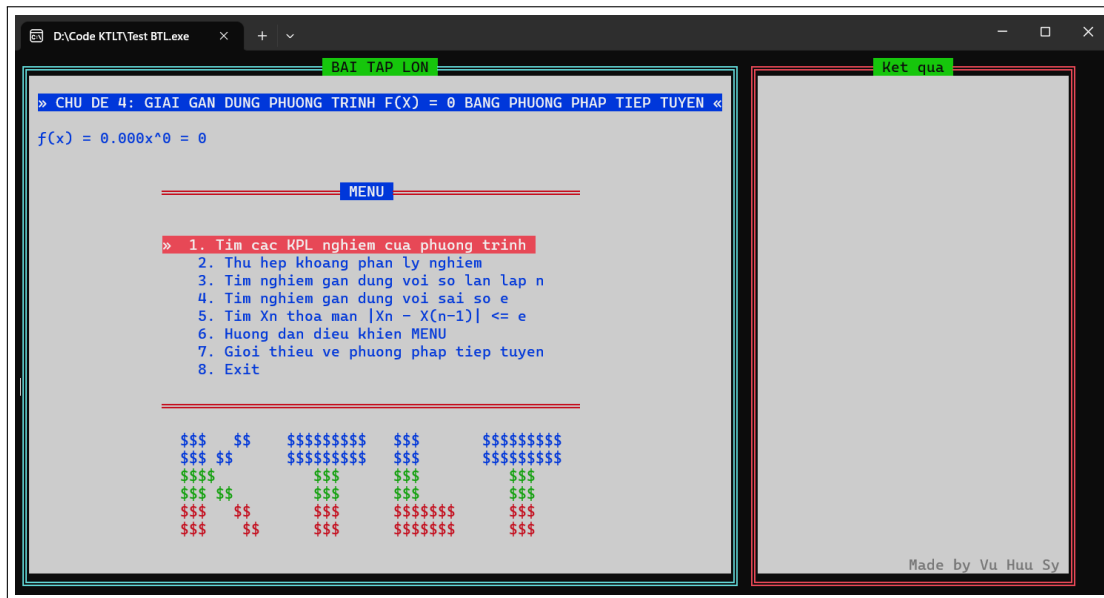
Đoạn mã trên gồm có 2 hàm, cụ thể:

1. Hàm `arrowHere(int realPosition, int arrowPosition)`: Đây là một hàm được sử dụng để ký tự 175 trong bảng mã ASCII chỉ đến vị trí hiện tại của chức năng trong menu. Hàm nhận vào hai tham số `realPosition` và `arrowPosition`.

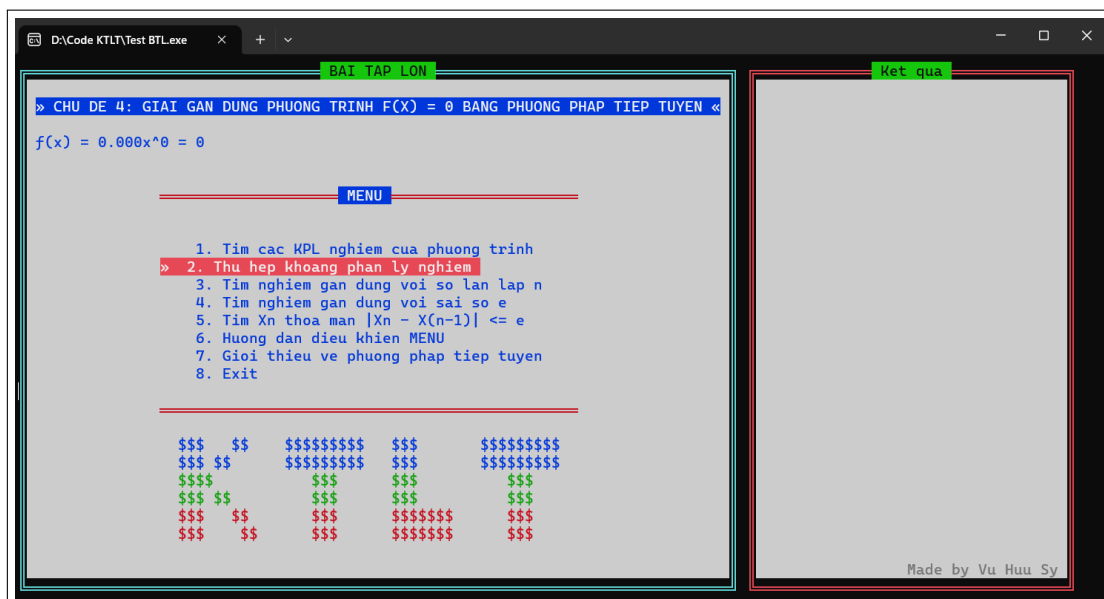
- `realPosition`: Đây là vị trí thực tế của chức năng đang được xem xét.
- `arrowPosition`: Đây là vị trí mà mũi tên hiện tại đang chỉ đến.

- Nếu `realPosition == arrowPosition`, chức năng đó sẽ được in màu đậm (màu 207) để chỉ rõ vị trí hiện tại. Nếu không, chức năng sẽ được in màu nhạt (màu 113).
2. Hàm `menu()`: Đây là hàm chính để hiển thị menu và xử lý lựa chọn của người dùng.
- Biến `position`: Đây là biến lưu trữ vị trí hiện tại của mũi tên menu. Ban đầu, `position` được thiết lập bằng 1, giả định rằng mũi tên sẽ chỉ đến tùy chọn đầu tiên của menu.
 - Biến `keyPress`: Đây là biến lưu trữ mã ASCII của phím mà người dùng ấn.
 - Trong hàm này, ta có một vòng lặp vô hạn (`while`) để hiển thị menu liên tục cho đến khi người dùng nhấn phím "Enter" (ASCII code 13) hoặc phím "Esc" (ASCII code 27).
 - Trong vòng lặp, ta sử dụng một vòng lặp `for` để từng chức năng của menu (từ 1 đến MAX).
 - Hàm `arrowHere(i, position)` được gọi để in ra màn hình các lựa chọn menu và mũi tên chỉ đến vị trí hiện tại của menu.
3. Biến `keyPress` sẽ lấy giá trị của phím mà người dùng ấn thông qua hàm `getch()`. Sau đó, ta sử dụng câu lệnh `switch` để xử lý lựa chọn người dùng.
4. Sau khi người dùng chọn một tùy chọn menu bằng phím "Enter", chương trình sẽ gọi hàm tương ứng với lựa chọn đó (ví dụ: `chucnang_1()`, `chucnang_2()`, v.v.). Sau khi thực hiện chức năng tương ứng, chương trình sẽ gọi lại hàm `menu()` để hiển thị menu tiếp tục cho người dùng.

Ví dụ: khi người dùng ấn "80" hoặc "s" hoặc "S" tức là xuống chức năng tiếp theo thì chức năng tiếp theo sẽ được in nổi bật hơn so với các chức năng khác. Trước khi ấn, vị trí mũi tên và chức năng 1 được in màu nổi bật hơn so với các chức năng khác.



Sau khi ấn, vị trí mũi tên đã chuyển đến chức năng 2 và chức năng 2 đã được in nổi bật hơn so với các chức năng khác. Chức năng 1 bây giờ không còn được in nổi bật nữa mà có màu như các chức năng khác.



Các phím chức năng khác hoạt động tương tự.

5.2 Bắt lỗi nhập sai của người dùng và yêu cầu nhập lại

Khi thực hiện chương trình, có thể xảy ra một số lỗi như:

- Không mở được file để ghi.
- Chương trình yêu cầu nhập số nguyên nhưng người dùng nhập chữ cái hoặc các ký tự đặc biệt.

- Người dùng nhập số quá to.
- Người dùng nhập sai khoảng phân ly nghiệm hoặc sai số quá lớn dẫn đến không tính toán được.
- ...

Để có thể kiểm soát được những lỗi trên, em đã viết những đoạn code chèn vào các hàm chức năng để có thể xử lý được những lỗi đó. Sau đây là những đoạn code kiểm soát lỗi và giải thích.

1. Lỗi không mở được file để ghi

```
1 FILE* file = fopen("input.txt", "w");
2     if (file == NULL) {
3         printf("Khong the mo file de ghi.\n");
4         return;
5     }
```

- if (file == NULL): Đoạn mã này kiểm tra xem việc mở tệp tin có thành công không bằng cách kiểm tra biến file xem có bằng NULL hay không. Biến file sẽ chứa con trỏ đến tệp tin nếu việc mở thành công. Nếu việc mở tệp tin không thành công, hàm fopen() sẽ trả về NULL.
- Trong trường hợp tệp tin không mở thành công (tức file bằng NULL), chương trình sẽ báo lỗi: **"Khong the mo file de ghi."** ra màn hình và thoát khỏi hàm bằng lệnh **"return;"**.

2. Lỗi ký tự nhập vào không phải là số nguyên.

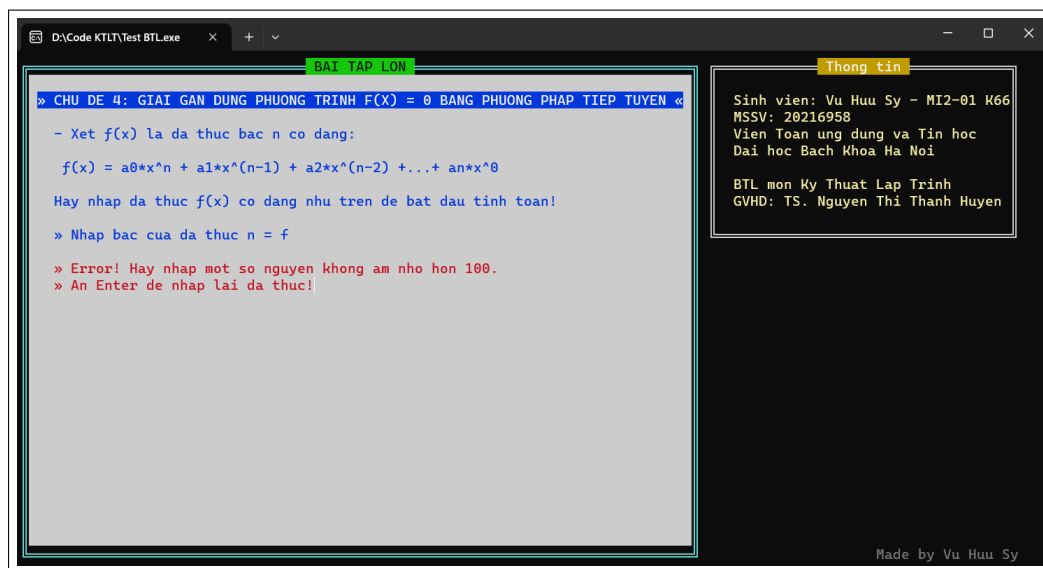
```
1 if (scanf("%d", &degree) != 1 || degree < 0 || degree > N) {
2     printf("Error! Hay nhap mot so nguyen khong am nho hon 100.");
3     printf("An Enter de nhap lai da thuc!");
4     while (getchar() != '\n');
5         getchar();
6         continue;
7 }
```

- Câu lệnh điều kiện (if) để kiểm tra đầu vào của người dùng. Hàm scanf được sử dụng để nhận giá trị nhập vào từ người dùng và lưu vào biến degree (với degree là một biến kiểu nguyên).

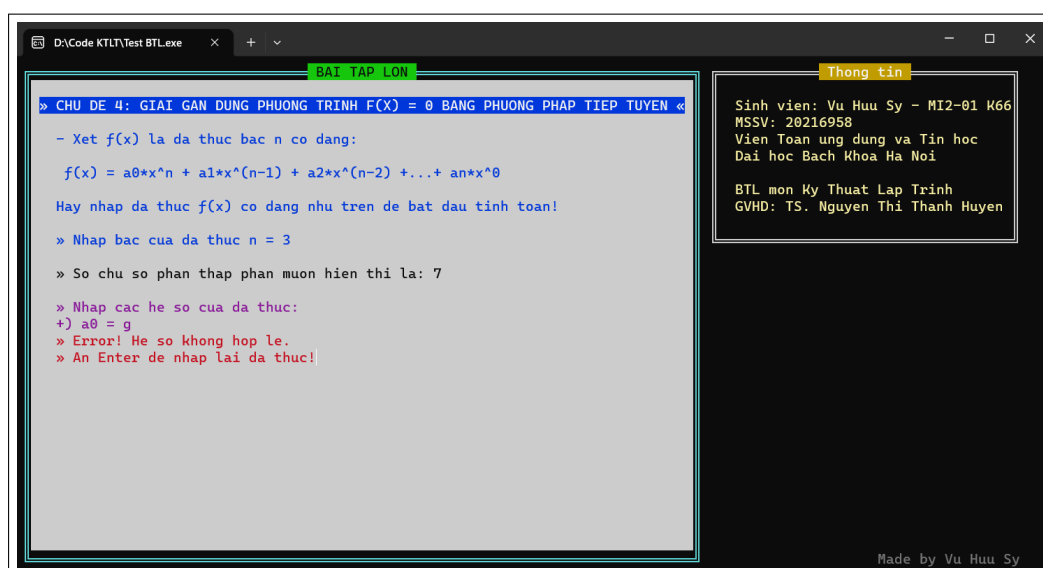
- Nếu một trong các điều kiện của câu lệnh if đúng, chương trình sẽ vào khối lệnh bên trong if để xử lý lỗi.
- Nếu chương trình vào khối lệnh bên trong if, tức là người dùng đã nhập sai hoặc không hợp lệ. Chương trình sẽ yêu cầu người dùng nhập lại đến khi nào đúng thì thôi.

Trong chương trình, trong mỗi lần yêu cầu người dùng nhập đầu vào vào thì em sẽ kèm theo đoạn code trên để bắt lỗi của người dùng nếu có. Vì chương trình chủ yếu chỉ yêu cầu nhập số nên 2 đoạn mã trên đã bao hàm được gần như tất cả các trường hợp gây lỗi có thể xảy ra.

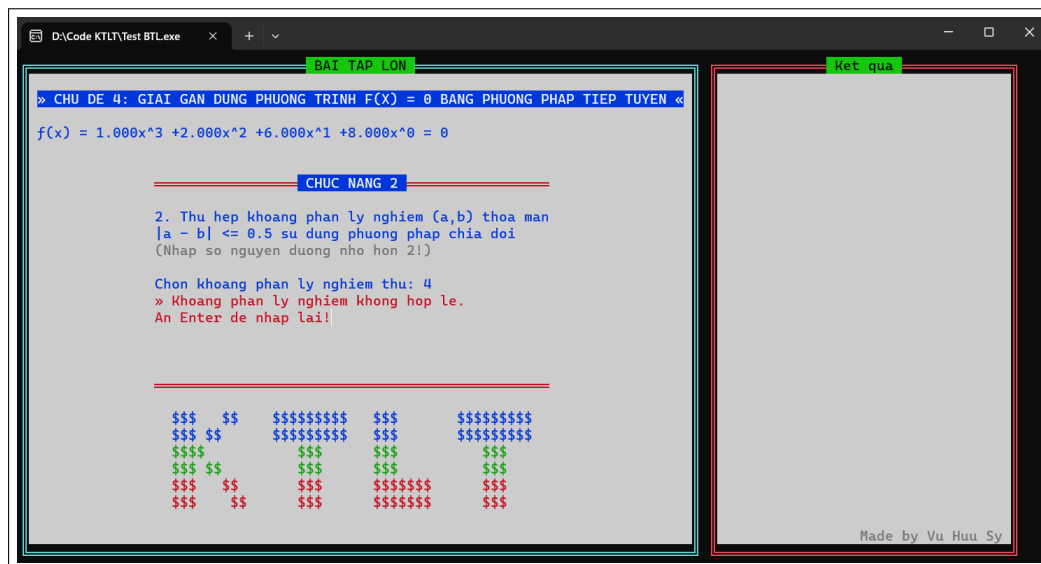
Một số hình ảnh giao diện bắt lỗi:



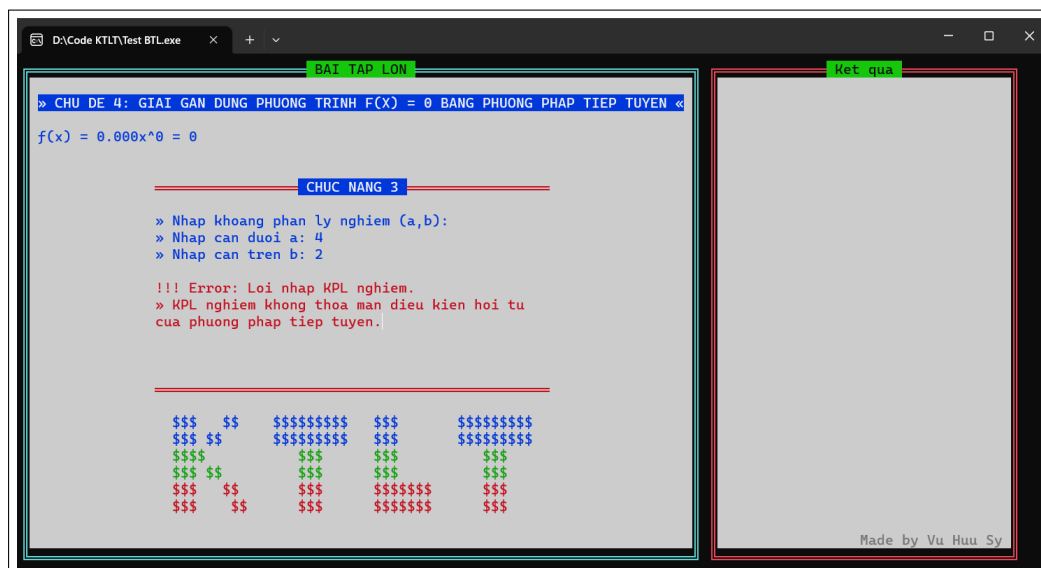
Lỗi nhập bậc của đa thức là một chữ cái



Lỗi nhập hệ số là một chữ cái



Lỗi nhập khoảng phân ly chọn từ chức năng 1



Lỗi nhập khoảng phân ly không hội tụ

5.3 Lưu kết quả tính toán được vào file text

Mỗi khi người dùng thực hiện một chức năng nào đó thì chương trình sẽ tạo ra một file mới hoặc là ghi đè lên file đã tồn tại kết quả của chức năng đang thực hiện, file được tạo ra nằm trong cùng một thư mục với file code của chương trình. Trong file sẽ bao gồm:

1. Tên chức năng thực hiện. VD: Chuc nang 1: Tim cac khoang phan ly nghiem cua phuong trinh.
2. Phương trình $f(x)$.

3. INPUT, OUTPUT

4. Các input người dùng nhập.

5. Quá trình lặp (nếu có).

6. Kết luận (số lần lặp, kết quả cuối cùng, đánh giá sai số,..).

Các đoạn mã ghi kết quả vào file được lồng vào xen kẽ giữa các đoạn mã nhận input từ người dùng và tính toán trong chương trình. Các giao diện kết quả trong file đã được em nêu ở **Chương 4**.

5.4 Số chữ số hiển thị ở phần thập phân

Em sử dụng một biến là **decimal_digit** để lưu số chữ số phần thập phân mà người dùng muốn hiển thị và toàn bộ kết quả tính toán của chương trình sẽ được hiển thị với decimal_digit chữ số phần thập phân. Đây là ví dụ một đoạn code in ra kết quả của lần lặp thứ %d nào đó:

```
1 printf("Lan lap %d: x%d = %.*lf", i, i, decimal_digit, x);
```

Trong đoạn code này:

- **".*"**: là một ký tự đặc biệt trong đặc tả định dạng. Nó cho phép chúng ta cung cấp một giá trị chính xác của số chữ số sau dấu thập phân. Chỗ này sẽ được thay thế bởi giá trị của biến decimal_digit.
- **lf**: là đặc tả định dạng để hiển thị một số thực dấu động. Biến x được hiển thị ở đây và sẽ có số chữ số sau dấu thập phân như đã được chỉ định bởi giá trị của biến decimal_digit.

5.5 Đồ họa của chương trình

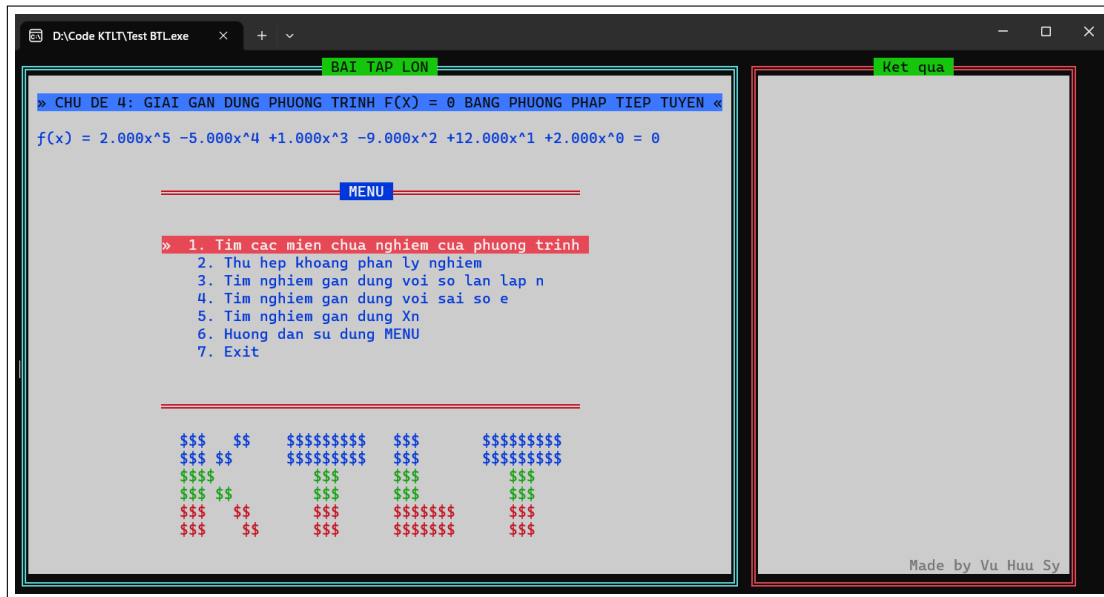
Đây là một chương trình hoạt động trơn tru và được phối màu một cách hợp lý. Đoạn mã tạo đồ họa cho chương trình nằm trong file header **"giao_dien.h"**.

Lý do em nói chương trình hoạt động trơn tru là do chương trình rất ít sử dụng câu lệnh xóa màn hình (system("cls")). Màn hình sẽ bị nháy mỗi khi xóa màn hình, điều này có thể gây chói mắt, mỏi mắt, khó chịu,.. cho người dùng. Thay vì xóa màn hình, em sẽ in đè lên màn hình cũ và điều đó đã giảm đáng kể việc màn hình bị nháy, chương trình hoạt động rất mượt mà.

Màu của chữ cũng được thay đổi linh hoạt. VD: màu chữ thông báo lỗi sẽ là màu đỏ, màu của comment là màu xanh. Màu của nền được đặt là màu xám để nhìn dịu mắt hơn.

Giao diện của chương trình được chia làm 2 phần:

- Phần điều khiển các chức năng ở bên trái
- Phần hiển thị kết quả tính toán ở bên phải



Giao diện của chương trình được em bỏ công sức và thời gian ra để căn từng góc sao cho đẹp mắt nhất nhưng không vì thế mà độ chính xác tính toán của chương trình giảm đi. Các chức năng của chương trình vẫn đưa ra kết quả chính xác.

Kết luận

1. Những điều đã học và làm được

- Biết cách in một xâu ký tự, vị trí của nó trên màn hình.
- Viết được chương trình có thể điều khiển được bằng các phím trên bàn phím, bầy một số phím người nhập vào như Tab, Esc, phím mũi tên,..
- Thay đổi màu chữ, màu nền để giao diện chương trình trông bớt nhàm chán và đẹp dễ hơn.
- Lường trước sai sót do người dùng đưa vào và xử lý lỗi đó.
- Biết cách in kết quả ra file.
- Và nhiều điều nữa.

2. Đánh giá chung

(a) Ưu điểm

- Đã làm được đủ các yêu cầu của đề bài.
- Thêm một số chức năng như: Hướng dẫn điều khiển menu và Giới thiệu về phương pháp tiếp tuyến.
- Giao diện chương trình đẹp mắt, dễ điều khiển.

(b) Nhược điểm

- Mã nguồn khá dài: 845 dòng trong chương trình chính và 293 dòng trong file header `giao_dien.h` tổng cộng là 1138 dòng mã.

3. Đánh giá chi tiết

- Các hằng số sử dụng define, sử dụng một mảng lưu các option để thuận tiện thay đổi và in ra.
- Chương trình viết có cấu trúc, các biến luôn được khai báo ở đầu hàm.

TT	Nội dung yêu cầu	Đã viết code? (y/n)	Đã thực hiện đúng? (y/n)	Tự đánh giá nhược điểm/ưu điểm, những điểm sáng tạo
1	Tìm các miền chứa nghiệm của phương trình	Y	Y	Làm tốt, có đánh số thứ tự KPL nghiệm để người dùng chọn ở chức năng 2
2	Tìm KPL nghiệm (a, b) của phương trình thỏa mãn $ a - b \leq 0,5$ bằng cách sử dụng phương pháp chia đôi để thu hẹp dần một KPL nghiệm đã tìm được ở ý 1	Y	Y	Làm tốt, có xử lý trường hợp không có KPL và người dùng phải thực hiện chức năng 1 trước khi thực hiện chức năng 2, nếu không chương trình sẽ báo lỗi
3	Tìm nghiệm gần đúng với số lần lặp n cho trước trong KPL nghiệm (a,b) và đánh giá sai số theo cả hai công thức (n được nhập vào từ bàn phím, (a,b) được nhập vào từ bàn phím)	Y	Y	Làm tốt, nhưng khi người dùng nhập vào số lần lặp lớn (>6) thì khi in kết quả ra console sẽ bị tràn ra khỏi menu nhưng kết quả trong file thì vẫn bình thường
4	Tìm nghiệm gần đúng trong khoảng (a, b) với sai số ε cho trước (ε được nhập vào từ bàn phím, (a, b) được nhập vào từ bàn phím). Tính toán theo 2 cách áp dụng công thức sai số	Y	Y	Làm tốt, có thêm phần cho phép người dùng chọn công thức sai số để tính toán
5	Tìm nghiệm gần đúng x_n trong khoảng (a,b) thỏa mãn điều kiện: $ a - b \leq \varepsilon$ (ε được nhập vào từ bàn phím)	Y	Y	Làm tốt

Tài liệu tham khảo

1. Phan Xuân Chánh, *Lập trình C/C++ - Đổi màu Console*, 2021.
2. Lập trình không khó, *Cách bắt sự kiện bàn phím trong C/C++*, 2018.
3. Tấn Chấn, *Lập trình game Rắn săn mồi trên C/C++ màn hình Console(Demo Snake)*, 2023.
4. *The complete table of ASCII characters, letters, codes, symbols and signs*, 2018.
5. Nguyễn Thị Thanh Huyền, *Bài giảng môn Kỹ thuật lập trình*, 2023.