

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN

**ĐỀ TÀI: TÌM NGHIỆM CỦA PHƯƠNG TRÌNH
ĐA THỨC BẰNG PHƯƠNG PHÁP CHIA ĐÔI**

Người hướng dẫn: **ThS. ĐỖ THỊ TUYẾT HOA**

Sinh viên thực hiện:

ĐỖ TRẦN BÌNH LỚP: 20T1 NHÓM: 20.10B

PHẠM SĨ CHIẾN LỚP: 20T1 NHÓM: 20.10B

Đà Nẵng, 06/2021

MỤC LỤC

DANH MỤC HÌNH VẼ	ii
MỞ ĐẦU.....	i
1. TỔNG QUAN ĐỀ TÀI.....	1
2. CƠ SỞ LÝ THUYẾT	1
2.1. Ý tưởng.....	1
2.2. Cơ sở lý thuyết.....	1
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN	3
3.1. Phát biểu bài toán	3
3.2. Cấu trúc dữ liệu	3
3.2.1. Các thư viện, các hàm được sử dụng	3
3.2.2. Danh sách liên kết đơn.....	4
3.3. Thuật toán.....	5
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ	7
4.1. Tổ chức chương trình	7
4.2. Ngôn ngữ cài đặt :	7
4.3. Kết quả.....	8
4.3.1. Giao diện chính của chương trình.....	8
4.3.2. Kết quả thực thi của chương trình.....	10
4.3.3. Nhận xét đánh giá	11
5. KẾT LUẬN	11
5.1. Kết quả đạt được.....	11
5.2. Hạn chế.....	11
5.3. Hướng phát triển.....	11
TÀI LIỆU THAM KHẢO	12
PHỤ LỤC	13

DANH MỤC HÌNH VẼ

1. Hình 1: Đồ thị hàm số $2^x + x - 4 = 0$
2. Hình 2: Cách tìm nghiệm μ
3. Hình 3: Khai báo danh sách liên kết đơn
4. Hình 4: Thuật toán đếm số phần tử có trong file
5. Hình 5: Thuật toán xoá phần tử ở vị trí $n/2$ trong mảng
6. Hình 6: Sơ đồ khối thuật toán tìm nghiệm bằng phương pháp chia đôi
7. Hình 7: Giao diện giới thiệu chương trình
8. Hình 8: Giao diện menu chương trình và thực hiện bước 1.
9. Hình 9: Chương trình thực hiện bước 2.
10. Hình 10: Chương trình thực hiện bước 3.

MỞ ĐẦU

Mục tiêu của đề tài là tìm hiểu, nghiên cứu cách xử lý chuỗi số trong file và giải phương trình đa thức bậc cao bằng phương pháp chia đôi. Cụ thể, tạo ra được một chương trình hỗ trợ tính toán mang tính trực quan, dễ sử dụng và cung cấp kết quả tính toán với độ chính xác cao.

Đối tượng nghiên cứu:

- File chứa dữ liệu là các phân tử số.
- Đa thức $G(x) = \sum_{i=0}^{M-1} a_i x^i$

Phạm vi nghiên cứu: đọc/ghi dữ liệu với file, chuẩn hoá nghiệm bằng phương pháp chia đôi, xây dựng và đánh giá thuật toán.

Phương pháp nghiên cứu được sử dụng trong đồ án:

- Phương pháp phân tích và tổng thích hợp thuyết.
- Phương pháp mô hình hoá.
- Phương pháp thực nghiệm.

Ngoài ra, đồ án lập trình tính toán còn sử dụng nhiều kiến thức từ các học phần song hành và học trước: phương pháp tính, toán rời rạc, kỹ thuật lập trình.

Xin gửi lời cảm ơn đến cô Đỗ Thị Tuyết Hoa đã hỗ trợ và hướng dẫn thực hiện đồ án này trong thời gian vừa qua. Trong quá trình nghiên cứu và thực hiện đồ án có thể còn có nhiều chỗ chưa chính xác và thiếu khoa học. Kính mong quý thầy cô góp ý giúp đỡ để đồ án có thể hoàn thiện hơn.

Xin chân thành cảm ơn!

1. TỔNG QUAN ĐỀ TÀI

- Nội dung đề tài: viết chương trình bằng ngôn ngữ C thực hiện các công việc sau đây
 - Đọc N ($N \geq 7$) phân tử số của danh sách liên kết từ file DAYSO.IN
 - Thực hiện xóa phân tử ở vị trí N2 (nếu N lẻ thì làm tròn), hiển thị kết quả và lưu kết quả ra file RESULT1.OUT.
 - Sau bước trên, ta có được một danh sách liên kết có M phân tử (a_0, a_1, \dots, a_{m-1}). Biết rằng các phân tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Tìm các nghiệm của phương trình $G(x) = 0$ bằng phương pháp chia đôi (với sai số $\varepsilon = 10^{-5}$), hiển thị kết quả và lưu kết quả ra file RESULT2.OUT.

- Quá trình thực thi chương trình:
 - Chọn chức năng tính toán và nhập dữ liệu đầu vào.
 - Kiểm tra dữ liệu với điều kiện của thuật toán và yêu cầu bài toán đặt ra.
 - Chương trình thực thi qua các bước như yêu cầu và trả về kết quả trong file.

2. CƠ SỞ LÝ THUYẾT

2.1. Ý tưởng

- Sử dụng kiến thức từ các học phần: Kỹ thuật lập trình, Phương pháp tính, Cấu trúc dữ liệu để xây dựng thuật toán, viết chương trình thực hiện những yêu cầu của đề tài. Cụ thể:
 - Thao tác với file: Đọc ghi dữ liệu từ file với các chế độ khác nhau.
 - Thao tác với mảng một chiều: đọc số phân tử trong mảng, nhập/xuất mảng, xóa một phân tử ở vị trí xác định trong mảng.
 - Chuẩn hoá nghiệm bằng phương pháp chia đôi: hệ số của đa thức $G(x)$ là các phân tử trong mảng sau khi xử lý mảng theo yêu cầu đề tài. Khoảng phân ly nghiệm $[a, b]$ được nhập từ bàn phím và sai số Epsilon cho trước.
- Xây dựng sơ đồ khối, thuật toán, viết từng chương trình con đảm nhiệm chức năng khác nhau. Sử dụng các chương trình con để xây dựng thành một chương trình hoàn chỉnh đáp ứng yêu cầu đề tài.

2.2. Cơ sở lý thuyết : Dùng phương pháp chia đôi

- Để tìm nghiệm gần đúng của phương trình $f(x) = 0$ ta tiến hành qua hai bước: tách nghiệm và chính xác hoá nghiệm.

2.2.1. Tách nghiệm:

- Xét tính chất nghiệm của phương trình, phương trình có nghiệm hay không, có bao nhiêu nghiệm, các khoảng chứa nghiệm nếu có. Đối với bước này, ta có thể dùng phương pháp đồ thị, kết hợp với các định lý mà toán học hỗ trợ.

❖ Phương pháp đồ thị:

▪ Trường hợp hàm $f(x)$ đơn giản

- Vẽ đồ thị $f(x)$.
- Nghiệm phương trình là hoành độ giao điểm của $f(x)$ với trục x , từ đó suy ra số nghiệm, khoảng nghiệm.

▪ Trường hợp $f(x)$ phức tạp

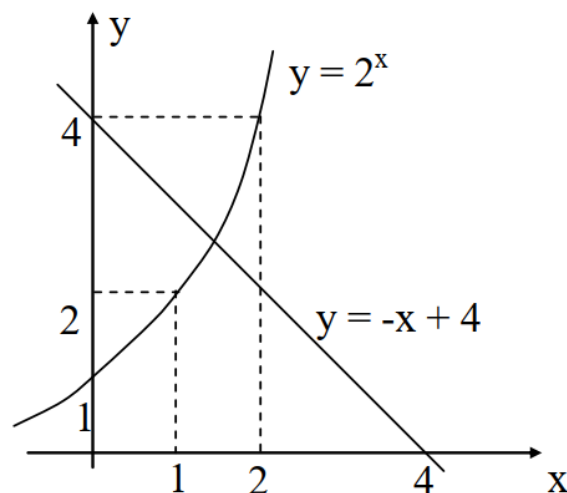
- Biến đổi tương đương $f(x) = 0 \Leftrightarrow g(x) = h(x)$
- Vẽ đồ thị của $g(x), h(x)$
- Vẽ hoành độ giao điểm của $g(x)$ và $h(x)$ là nghiệm của phương trình, từ đó suy ra số nghiệm, khoảng nghiệm.

❖ Định lý: Giả sử $f(x)$ liên tục và trái dấu tại hai đầu $[a, b]$. Khi đó trên (a, b) tồn tại một số lẻ nghiệm thực của phương trình $f(x) = 0$. Nghiệm là duy nhất nếu $f'(x)$ tồn tại và không đổi dấu trên (a, b) .

- Ví dụ: Tách nghiệm cho phương trình sau: $2^x + x - 4 = 0$

Giải: $2^x + x - 4 = 0 \Leftrightarrow 2^x = -x + 4$

Áp dụng phương pháp đồ thị:



Hình 1.

Từ đồ thị suy ra: Phương trình có 1 nghiệm $x \in (1, 2)$.

2.2.2. Chuẩn hoá nghiệm bằng phương pháp chia đôi:

Chuẩn hoá nghiệm là việc thu hẹp dần khoảng chứa nghiệm để hội tụ được đến giá trị nghiệm gần đúng với độ chính xác cho phép.

- Cho phương trình $f(x) = 0$, $f(x)$ liên tục và trái dấu tại 2 đầu $[a, b]$. Giả sử $f(a) < 0$, $f(b) > 0$ (nếu ngược lại thì xét $-f(x) = 0$). Theo định lý về sự tồn tại nghiệm của phương trình, trên $[a, b]$ phương trình có ít nhất 1 nghiệm μ .
- Cách tìm nghiệm μ :

Đặt $[a_0, b_0] = [a, b]$ và lập các khoảng lồng nhau $[a_n, b_n]$ ($n = 1, 2, 3, \dots$)

$$[a_{i+1}, b_{i+1}] = \begin{cases} [a_i, (a_i + b_i)/2] & \text{nếu } f((a_i + b_i)/2) > 0 \\ [(a_i + b_i)/2, b_i] & \text{nếu } f((a_i + b_i)/2) < 0 \end{cases}$$

Hình 2.

Như vậy:

- Hoặc nhận được nghiệm đúng ở một bước nào đó:
 $\mu = (a_i + b_i)/2$ nếu $f((a_i + b_i)/2) = 0$.
- Hoặc nhận được 2 dãy $\{a_n\}$ và $\{b_n\}$, trong đó:
 $\{a_n\}$ là dãy đơn điệu tăng và bị chặn trên.
 $\{b_n\}$ là dãy đơn điệu giảm và bị chặn dưới.

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1. Phát biểu bài toán

- Dữ liệu đầu vào:
 - Tên file chứa các phân tử số cần tính toán.
 - Khoảng nghiệm (a, b) nhập vào từ bàn phím.
- Dữ liệu đầu ra:
 - Kết quả thực hiện bài toán được lưu ở 2 file: RESULT1.OUT và RESULT2.OUT.
 - Đồng thời in ra màn hình để quan sát các bước thực hiện chương trình.

3.2. Cấu trúc dữ liệu

3.2.1. Các thư viện, các hàm được sử dụng

- Các thư viện được sử dụng:
 - Thư viện **stdio.h**
 - Các hàm thao tác với file:
 - Mở file: `FILE* fopen(const char *file, const char *mode)`
 - Đóng file: `fclose()`
 - Đọc/ ghi file: `fscanf()`, `fprintf()`
 - Các hàm nhập xuất dữ liệu: `printf()`, `scanf()`
 - Thư viện **conio.h**

- Hàm getch(): dừng màn hình console lại để xem kết quả. Nhấn để tiếp tục tính toán.
- Thư viện **windows.h**
 - Hàm sleep(int n): dừng chương trình trong khoảng n ms, sau đó chương trình tiếp tục thực thi.
- Thư viện **stdlib.h**
 - Hàm int atoi(const char *str): chuyển đổi một chuỗi thành một số nguyên.
 - Hàm system("cls"): xóa tất cả những dữ liệu xuất ra trên màn hình console.
 - Hàm system("color fx"): cài đặt màu nền console.
- Thư viện **math.h**
 - Hàm pow(x, y): sử dụng để tính lũy thừa giữa hai số x và y. Trong đó x là hệ số và y là số mũ.

3.2.2. Danh sách liên kết đơn

- Khai báo:

```
struct dslk{  
  
    char tenfile[100];  
    float dayA[100], dayB[100];  
    int n, k;  
};  
dslk data;
```

Hình 3.

- Danh sách liên kết này chứa các dữ liệu:
 - Tên file cần đọc lấy dữ liệu.
 - Mảng 1 chiều dayA chứa dữ liệu chưa xử lý.
 - Mảng 1 chiều dayB chứa dữ liệu đã qua bước 2 theo yêu cầu đồ án.
 - Số k là số phần tử số có trong file, n là số phần tử số cần lấy ra từ file.

3.3. Thuật toán

3.3.1. Thuật toán đếm số phần tử có trong file

```
FILE *fp;
int i;
data.k=0;
float temp;

fp = fopen(data.tenfile, "r");
while (fscanf(fp, "%f", &temp) > 0) data.k++;
fclose(fp);
printf("\nSố phần tử của day: %d", data.k);
```

Hình 4.

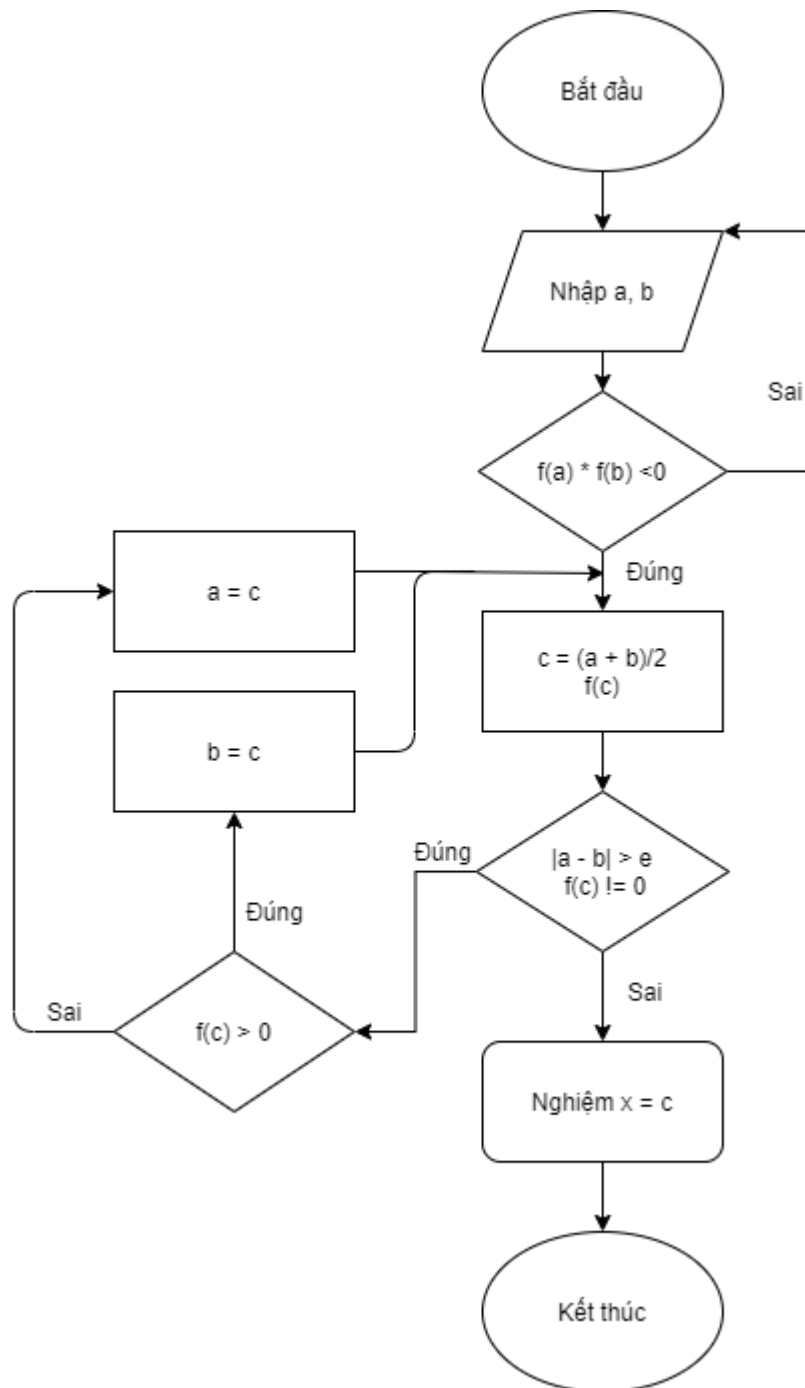
3.3.2. Thuật toán xóa phần tử ở vị trí $n/2$ (vị trí giữa) trong mảng

```
int i,x;
x=data.n/2;
printf("\n\nThực hiện xóa phần tử thứ %d của day so.",x);
printf("\n\nTa được day so mới: ");
for( i=0; i<data.n-1; i++)
{
if (i>=x-1){
data.dayB[i] = data.dayB[i+1];
printf("%f ", data.dayB[i]);
}
else printf("%f ", data.dayB[i]);
};
```

Hình 5.

3.3.3. Thuật toán tìm nghiệm bằng phương pháp chia đôi:

- Sơ đồ khối:



Hình 6.

- Độ phức tạp thuật toán: $O(\log(n))$

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1. Tổ chức chương trình

- Hàm main() sẽ in ra bảng giới thiệu gồm các thông tin: Họ tên, lớp của sinh viên làm đồ án; giáo viên hướng dẫn đồ án. Sau đó gọi hàm menu().
- Hàm menu() in các lựa chọn chức năng và xử lý lựa chọn của người dùng. Chức năng 2 sẽ kết thúc chương trình. Nếu chọn chức năng 1, sẽ gọi hàm kiemtratenfile().
- Hàm kiemtratenfile(): kiểm tra sự tồn tại của file. Nếu không có sẽ yêu cầu nhập lại, nếu có sẽ gọi hàm docfile()
- Hàm docfile(): yêu cầu nhập số phần tử cần lấy.
 - Nếu bé hơn 7 hoặc kiểm tra nhập không phải là số (kí tự,..) sẽ yêu cầu nhập lại.
 - Nếu lớn hơn số phần tử của file sẽ lấy tối đa là số phần tử có trong file.
 - Nếu không, chương trình đọc dữ liệu từ file và gán vào mảng dayA.
- Hàm xoapt() được gọi: sao chép mảng dayA vào mảng dayB. Thực hiện xóa phần tử vị trí $n/2$ và tiến các phần tử sau lên vị trí $n/2$. Sau đó lưu dayB vào file RESULT1.OUT và gọi hàm dathuc().
- Hàm dathuc() in đa thức với các hệ số tương ứng trong mảng dayB ra màn hình và gọi hàm khoangnghiem().
- Hàm khoangnghiem(): yêu cầu nhập khoảng (a, b) kiểm tra xem có thỏa mãn yêu cầu của thuật toán.
 - Nếu không, yêu cầu nhập lại.
 - Nếu có, thực hiện tính nghiệm gần đúng theo phương pháp chia đôi. Quá trình tính được in ra màn hình và kết quả được lưu vào file RESULT2.OUT
- Sau đó, người dùng có thể lựa chọn để thoát về Menu hoặc nhập khoảng nghiệm (a, b) mới để tìm nghiệm khác. Nghiệm mới được ghi nối tiếp vào file RESULT2.OUT.

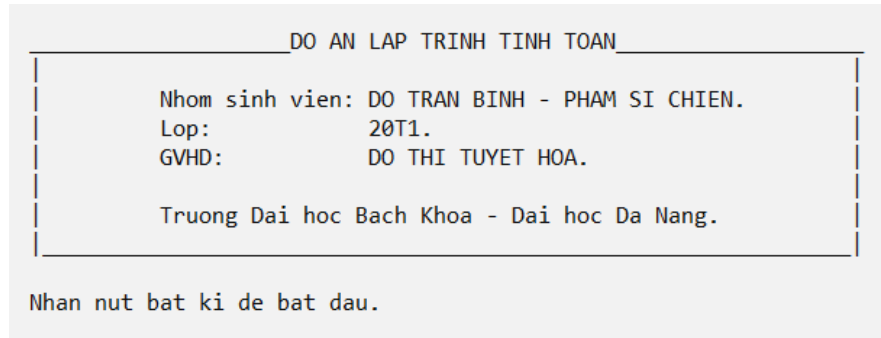
4.2. Ngôn ngữ cài đặt :

- Ngôn ngữ lập trình được sử dụng: C
- Môi trường lập trình được sử dụng: Dev – C ++

4.3. Kết quả

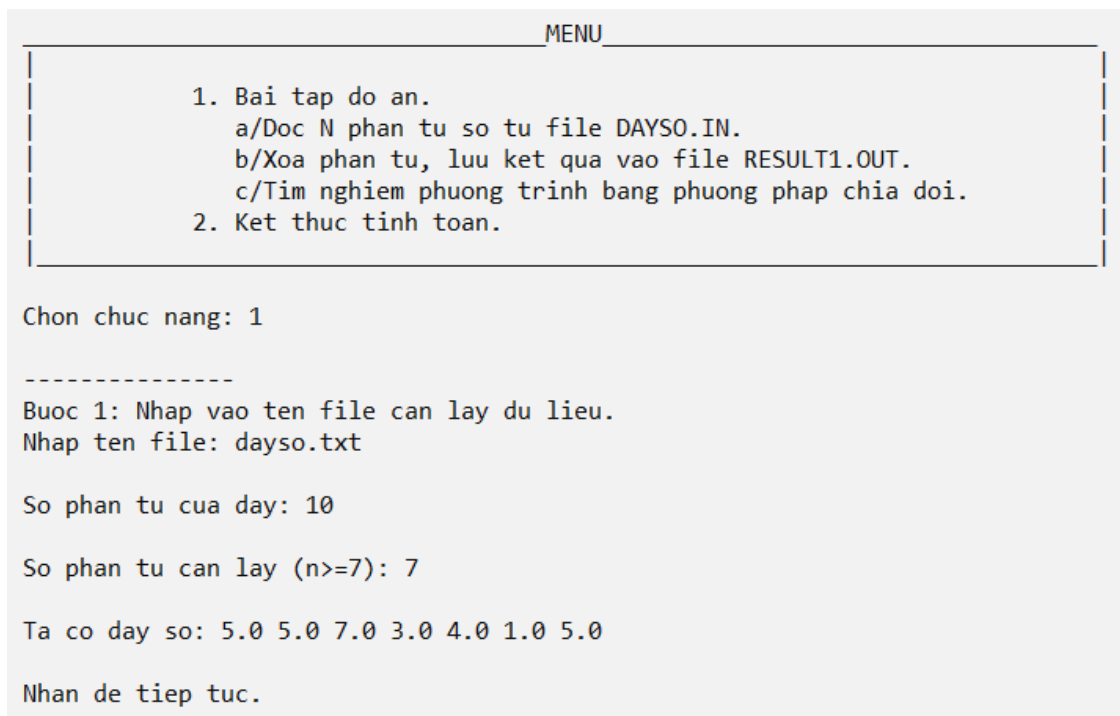
4.3.1. Giao diện chính của chương trình

- Sau khi thực thi chương trình, một bản giới thiệu sẽ hiện ra.



Hình 7.

- Nhấn Enter để vào menu chọn chức năng
 - Lựa chọn 1 để bắt đầu tính toán.
 - Lựa chọn 2 để kết thúc chương trình.
- Sau khi lựa chọn chức năng 1:
 - Nhập tên file. Kiểm tra có file sẽ đến bước 1, nếu không có file đó yêu cầu nhập file khác.
 - Bước 1: Nhập số phần tử cần lấy (≥ 7) nhập đúng sẽ đọc file và lấy dữ liệu. Nhập sai số phần tử (nhỏ hơn 7, hoặc nhập kí tự thay vì số) sẽ phải nhập lại.



Hình 8.

- Bước 2: Chương trình xoá đi phần tử ở vị trí $n/2$ (nếu n lẻ sẽ làm tròn). Rồi in kết quả dãy số mới ra màn hình, lưu kết quả vào file RESULT1.OUT

Buoc 2:

Thuc hien xoa phan tu thu 3 cua day so.

Ta duoc day so moi: 5.00 5.00 3.00 4.00 1.00 5.00

Ket qua da luu vao file RESULT1.OUT.

Nhan de tiep tuc.

Hình 9.

- Bước 3:
 - In ra màn hình đa thức có dạng $G(x) = \sum_{i=0}^{M-1} a_i x^i$. Nhập a, b thoả mãn yêu cầu thuật toán (nếu không chính xác sẽ phải nhập lại).
 - Sau khi a, b thoả mãn, chương trình tự động tính nghiệm đa thức theo phương pháp chia đôi và hiển thị quá trình tính lên màn hình. Kết quả được lưu vào file RESULT2.OUT.
 - Sau khi tính xong, có thể lựa chọn nhập lại khoảng (a, b) mới để tìm nghiệm khác hoặc về Menu.

```

Buoc 3:

Ta co da thuc:  $5.0x^0 + 5.0x^1 + 3.0x^2 + 4.0x^3 + 1.0x^4 + 5.0x^5$ 

Nhap a sao cho  $f(a) < 0$  : -2

Nhap b sao cho  $f(b) > 0$  : 3

f(a) = -169.00
f(b) = 1451.00

|           a           |           b           |           f(c)          |
|-----|-----|-----|
| -2.000000 | 0.500000 | 8.968750 |
| -2.000000 | -0.750000 | 0.379883 |
| -1.375000 | -0.750000 | -27.601532 |
| -1.062500 | -0.750000 | -7.219609 |
| -0.906250 | -0.750000 | -2.426439 |
| -0.828125 | -0.750000 | -0.832000 |
| -0.789063 | -0.750000 | -0.184362 |
| -0.789063 | -0.769531 | 0.107477 |
| -0.779297 | -0.769531 | -0.035927 |
| -0.779297 | -0.774414 | 0.036393 |
| -0.779297 | -0.776855 | 0.000389 |
| -0.778076 | -0.776855 | -0.017730 |
| -0.777466 | -0.776855 | -0.008661 |
| -0.777161 | -0.776855 | -0.004134 |
| -0.777008 | -0.776855 | -0.001872 |
| -0.776932 | -0.776855 | -0.000741 |
| -0.776894 | -0.776855 | -0.000176 |
| -0.776894 | -0.776875 | 0.000106 |
| -0.776884 | -0.776875 | -0.000035 |

Phuong trinh co nghiem: -0.776884

Ket qua da ghi vao file RESULT2.OUT.

1. Quay lai menu.
2. Nhap khoang nghiem khac.
Lua chon:

```

Hình 10.

4.3.2. Kết quả thực thi của chương trình

- File RESULT1.OUT




RESULT1.txt - Notepad

File Edit Format View Help

5.000000 5.000000 3.000000 4.000000 1.000000 5.000000

- File RESULT2.OUT

 RESULT2.txt - Notepad

File Edit Format View Help

Phương trình cơ nghiệm: -0.776884.

4.3.3. Nhận xét đánh giá

- Đã xây dựng được một chương trình cơ bản đáp ứng đủ yêu cầu của đề tài.
- Trong quá trình làm việc đã có thêm nhiều kinh nghiệm về xây dựng thuật toán, lập trình, xử lý lỗi, làm việc nhóm.
- Chương trình còn có một số hạn chế nhất định.

5. KẾT LUẬN

5.1. Kết quả đạt được : Đã đáp ứng được yêu cầu của đề tài.

- Từ đầu vào là danh sách liên kết các phần tử số trong file DAYSO.IN
- Có đầu ra là kết quả tính toán theo yêu cầu lưu trong 2 file RESULT1.OUT và RESULT2.OUT với sai số nhỏ hơn $\epsilon=10^{-5}$
- Khắc phục được một số lỗi nhập dữ liệu: nhập sai kiểu dữ liệu, không tồn tại file (lựa chọn),...
- Có thể nhập liên tục nhiều khoảng nghiệm khác nhau để tìm những nghiệm khác và ghi nối tiếp vào file kết quả.
- Giao diện chương trình dễ nhìn, thân thiện và dễ sử dụng.

5.2. Hạn chế

- Trong một khoảng (a, b) cho trước chỉ có thể tìm được một nghiệm.
- Chưa tìm được nhiều nghiệm khi cho một khoảng (a, b) có chứa nhiều nghiệm.
- Còn chưa xử lý được khi nhập sai kiểu dữ liệu.

5.3. Hướng phát triển

- Khắc phục những hạn chế đã nêu trên.
- Thêm lựa chọn tạo file và ghi dữ liệu vào file nếu thấy file không tồn tại.
- Tối ưu hoá thuật toán để xử lý nhanh hơn, chính xác hơn và tốn ít tài nguyên máy tính hơn.

TÀI LIỆU THAM KHẢO

1. Giáo trình Phương Pháp Tính. ThS Đỗ Thị Tuyết Hoa
2. Giáo trình Cấu Trúc Dữ Liệu. Thầy Phan Chí Tùng

PHỤ LỤC

1. Mã nguồn được lưu tại: <https://github.com/dtrbinh/PBL1>