TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

**TÊN ĐỀ TÀI**

**Đề 320**

Người hướng dẫn**: TS. PHẠM CÔNG THẮNG**

Sinh viên thực hiện**:**

**Họ và tên: Nguyễn Nhật Huy**

**LỚP: 22T\_Nhat2**

**Họ và tên: Trần Minh Toàn**

**LỚP: 22T\_DT3**

**NHÓM: 16**

**Đà Nẵng, 06/2023**

Mục lục

[1. TỔNG QUAN VỀ ĐỀ TÀI 3](#_Toc138630504)

[2. CƠ SỞ LÝ THUYẾT 4](#_Toc138630505)

[2.1. Ý tưởng. 4](#_Toc138630506)

[2.2 Cơ sở lý thuyết. 4](#_Toc138630507)

[3. THUẬT TOÁN 5](#_Toc138630508)

[3.1. Phát biểu bài toán. 5](#_Toc138630509)

[3.2. Cấu trúc dữ liệu. 6](#_Toc138630510)

[4. CÁC HÀM XỬ LÝ CHÍNH 8](#_Toc138630511)

[5. KẾT QUẢ CHƯƠNG TRÌNH 11](#_Toc138630512)

[5.1. Tổ chức chương trình. 11](#_Toc138630513)

[5.2. Ngôn ngữ cài đặt. 11](#_Toc138630514)

[5.3. Giao diện chính. 12](#_Toc138630515)

[5.4. Kết quả thực thi 12](#_Toc138630516)

[5.5. Nhận xét 15](#_Toc138630517)

[6. KẾT LUẬN 16](#_Toc138630518)

[6.1. Kết quả đạt được 16](#_Toc138630519)

[6.2. Hạn chế 16](#_Toc138630520)

[6.3. Hướng phát triển 16](#_Toc138630521)

[7. TÀI LIỆU THAM KHẢO 17](#_Toc138630522)

[8. PHỤ LỤC 18](#_Toc138630523)

# TỔNG QUAN VỀ ĐỀ TÀI

Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào (𝑁 5), danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm N hàng, mỗi hàng gồm M phần tử số (M = N - 1). Mỗi hàng này tương ứng là một danh sách liên kết).

b)Thực hiện thêm vào mỗi danh sách một phần tử, hiển thị kết quả và lưu kết quả ra file RESULT1.OUT.

c) Sau khi thực hiện ở câu b), chúng ta nhận được 𝑁 danh sách liên kết, mỗi danh sách có 𝑁 phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng với mỗi hàng của ma trận):

A =

Hãy tìm nghiệm X = ( x0, x1,…, xN)T của hệ phương trình 𝐴𝑋 = 𝐵 bằng phương pháp lặp đơn, với 𝐵 = (b0,b1,…, bN)T (với B là mảng một chiều được nhập vào), sau đó hiển thị kết quả và lưu kết quả ra file RESULT2.OUT.

# CƠ SỞ LÝ THUYẾT

## 2.1. Ý tưởng.

Sử dụng cấu trúc Stack để lưu trữ từng hàng của ma trận đọc vào từ file. Sau đó dùng một mảng một chiều để quản lý các hàng của ma trận, mỗi phần tử của mảng trỏ tới Node đầu Stack chứa hàng ma trận tương ứng.

Sử dụng phương pháp lặp đơn để tính toán nghiệm của phương trình.

## Cơ sở lý thuyết.

[1] **Phương pháp lặp đơn**:



Biến đổi hệ phương trình về dạng: x  Bx g



x  (x1 , x2 ,., xn) ; g  (g1 , g2 ,..., gn ) ; B = {bij}n

Cách biến đổi:

|  |
| --- |
| a11x1 +a12x2 + + a1nxn = a1n+1 |
| a21x1 +a22x2 + + a2nxn = a2n+1 |
| ....... |
| an1x1 +an2x2 + + annxn = ann+1 |

n

x1  (a1n1  a1 jx j ) / a11 ( j  1)

j1

….

n

xn  (ann1  an jx j ) / ann ( j  n)

j1

\*Tổng quát:

n

xi  (ain 1  aij x j ) / aii ( j  i) (\*) (i = 1n)

j1

Cho hệ phương trình xấp xỉ nghiệm ban đầu: x 0  (x0 1, x0 2,..., x0 n)



Thay x0 vào (\*) để tính: x 1  (x1 1, x1 2,..., x1 n)

n

x1i = (ain + 1 - aij x0i)/aii (j  i) j1



Tương tự, tính x2, x3, …

n

Tổng quát: xk+1i = (ain + 1 - aij xki)/aii (j  i) j1

Quá trình lặp sẽ dừng khi thỏa mãn **tiêu chuẩn hội tụ** tuyệt đối:

xk+1i xki   (i  1, n)



Khi đó xk  (xk1 , xk 2 ,., xkn )

Điều kiện hội tụ:

Hệ phương trình có ma trận lặp B thỏa mãn:

n

r1  max  bij  1

i j1

n

hoặc r2  max  bij  1

j i1

n n

hoặc r3  bij)2  1

i1 j=1

thì quá trình sẽ hội tụ đến nghiệm.

# THUẬT TOÁN

## Phát biểu bài toán.

Mô tả đầu vào (Input) và đầu ra (Output) của bài toán.

**Input:**

* File DAYSO.IN gồm một ma trận (nxm) có n dòng và m cột, mỗi phần tử tách nhau bởi một dấu cách trắng.
* Mảng một chiều B được nhập từ bàn phím.

**Output:**

* File RESULT1.OUT chứa ma trận đã thêm một phần tử vào mỗi hàng.
* File RESULT2.OUT chứa nghiệm của hệ phương trình.

## Cấu trúc dữ liệu.

**Mảng một chiều:**  là tập hợp tuần tự các phần tử **có cùng kiểu dữ liệu** và các phần tử được lưu trữ trong **một dãy các ô nhớ liên tục** trên bộ nhớ.

[2]**Danh sách liên kết đơn:** là một tập hợp các Node được phân bố động, được sắp xếp theo cách sao cho mỗi Node chứa *một giá trị* (**Data**) và *một con trỏ* (**Next**). Con trỏ sẽ trỏ đến phần tử kế tiếp của danh sách liên kết đó. Phần tử cuối cùng của danh sách liên kết được chỉ tới NULL.(Hình 3.1)



*Hình 1. Hình minh họa cấu trúc của một danh sách liên kết đơn.*

*(Nguồn: https://blog.luyencode.net/danh-sach-lien-ket-don/)*

[3] **Stack (Ngăn xếp):** là 1 dạng đặc biệt của danh sách liên kết mà việc bổ sung hay loại bỏ 1 phần tử đều thực hiện ở 1 đầu của danh sách gọi là đỉnh. Ngăn xếp có 2 thao tác cơ bản: thêm phần tử vào được gọi là push và loại bỏ phần tử được gọi là pop. Việc loại bỏ phần tử sẽ tiến hành loại bỏ phần tử mới nhất được đưa vào danh sách, chính vì tính chất này mà ngăn xếp còn được gọi là kiểu dữ liệu LIFO( last in first out – Vào sau ra trước).(Hình 3.2)

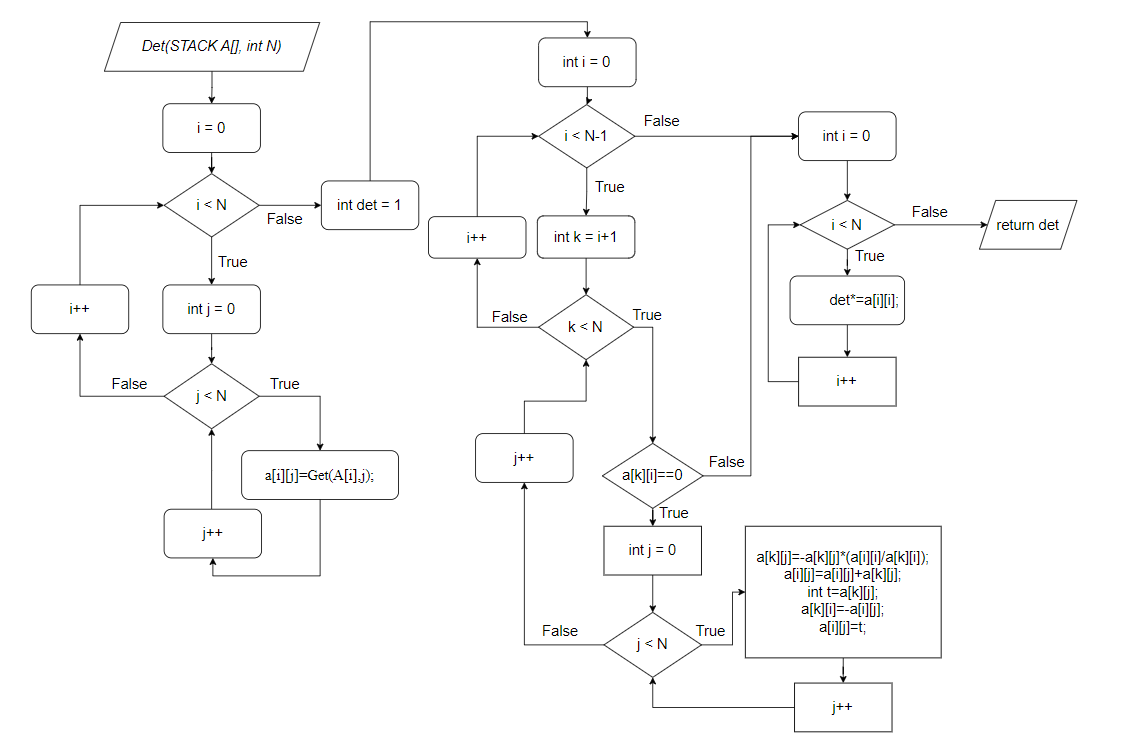


*Hình 2. Hinh minh họa cấu trúc của một stack.*

*(Nguồn: https://gochocit.com/ky-thuat-lap-trinh/ngan-xep-stack-la-gi-cach-xay-dung-ngan-xep)*

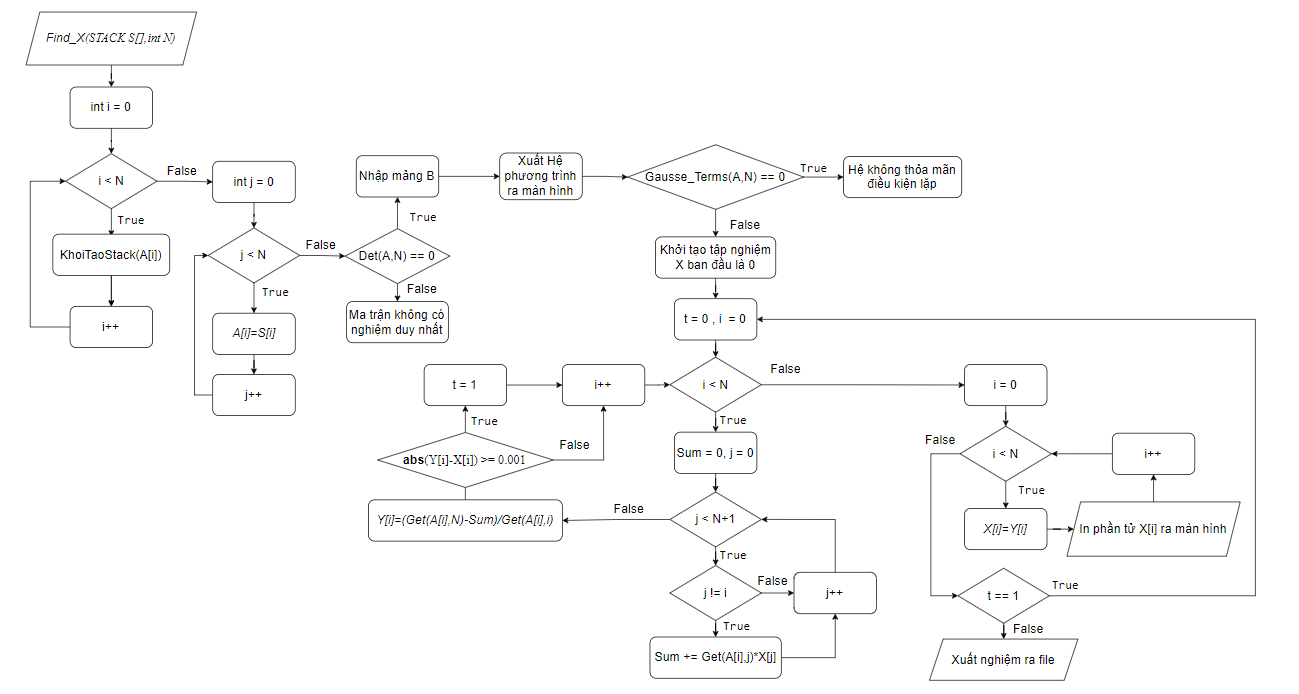
# CÁC HÀM XỬ LÝ CHÍNH

**Hàm Det:** Tính định thức của ma trận.

**

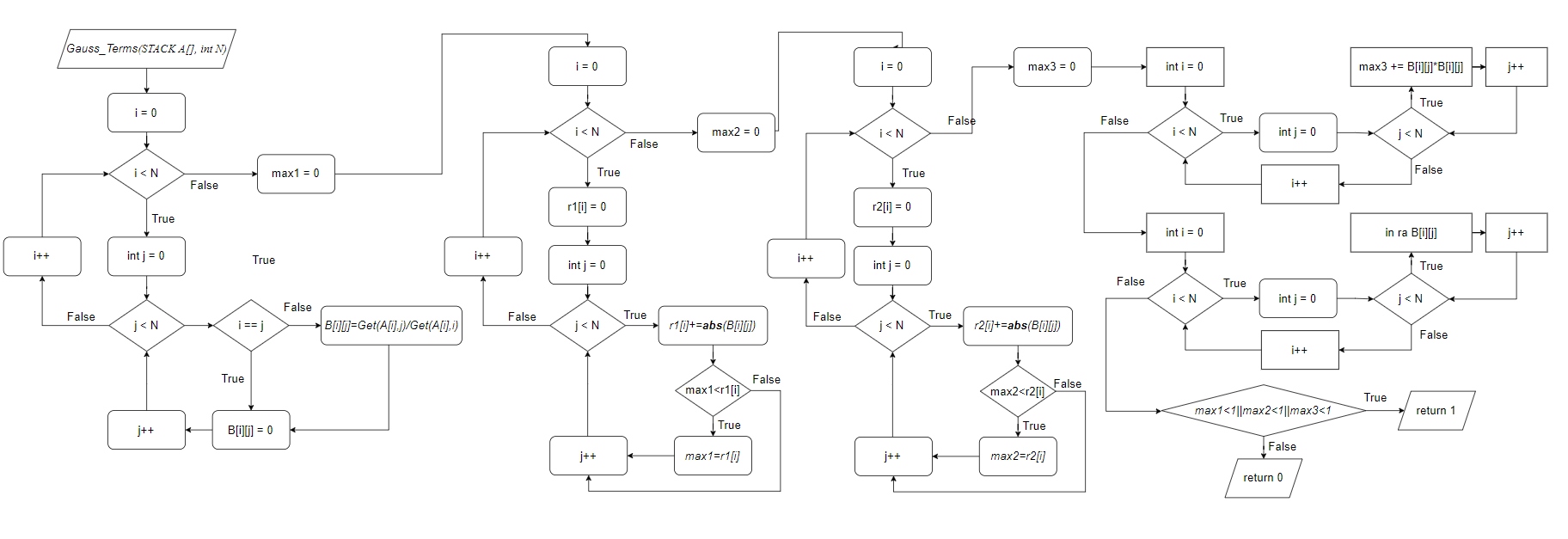
*Hình 3. Biểu đồ khối thuật toán hàm tính định thức ma trận.*

**Hàm Find\_X()**: Tìm tập nghiệm X của hệ phương trình AX=B với B là mảng nhập vào và xuất nghiệm ra file.

**

*Hình 4. Biểu đồ khối thuật toán hàm tìm nghiệm cho hệ phương trình theo phương pháp lặp đơn.*

**Hàm Gauss\_Terms:** Kiểm tra điều kiện hội tụ về ngiệm của phương pháp lặp.



*Hình 5.* *Biểu đồ khối thuật toán hàm kiểm tra điều kiện hội tụ của phương pháp lặp đơn.*

# KẾT QUẢ CHƯƠNG TRÌNH

## Tổ chức chương trình.

Chương trình khởi tạo một cấu trúc node để lưu trữ các phần tử trong ma trận, sau đó sử dụng các hàm bao gồm:

* KhoitaonNode: Để khởi tạo một node.
* IsEmpty: Để kiểm tra stack có rỗng hay không.
* Push: Thêm một phần tử vào đầu stack.
* Pop: Lấy ra phần tử đầu trong stack.
* Top: Lấy thông tin phần tử đầu stack.
* Reverse: Đảo ngược giá trị của stack.
* Pushback: Thêm phần tử vào cuối stack.
* Get truy vấn: Đến một giá trị của node có trong stack.
* Print\_Stack: Xuất ra một dòng trong ma trận.
* Print\_Matrix: In ma trận ra màn hình.
* Write\_File1: Ghi ma trận vào file sau khi thêm vào mỗi hàng một phần tử.
* Write\_File2: Ghi các nghiệm của phương trình vào file.
* Make\_Matrix : Quản lý ma trận bằng một mảng, mỗi phần tử của mảng là một stack chứa một hàng của ma trận.
* Read\_File: Đọc ma trận từ file.
* Det: Tính định thức của ma trận.
* Find\_X: Tìm nghiệm của biểu thức AX = B.
* Menu: tạo bảng chọn.

## Ngôn ngữ cài đặt.

Ngôn ngữ lập trình C : là một ngôn ngữ [mệnh lệnh](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_ki%E1%BB%83u_m%E1%BB%87nh_l%E1%BB%87nh) được phát triển từ đầu [thập](https://vi.wikipedia.org/wiki/Th%E1%BA%ADp_ni%C3%AAn_1970) [niên 1970](https://vi.wikipedia.org/wiki/Th%E1%BA%ADp_ni%C3%AAn_1970) bởi [Dennis Ritchie](https://vi.wikipedia.org/wiki/Dennis_Ritchie) để dùng trong [hệ điều hành UNIX](https://vi.wikipedia.org/wiki/Unix). Từ đó, ngôn ngữ này đã lan rộng ra nhiều hệ điều hành khác và trở thành một những ngôn ngữ phổ dụng nhất. C là ngôn ngữ rất có hiệu quả và được ưa chuộng nhất để viết các [phần mềm hệ](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_h%E1%BB%87_th%E1%BB%91ng) [thống](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_h%E1%BB%87_th%E1%BB%91ng), mặc dù nó cũng được dùng cho việc viết các [ứng dụng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_%E1%BB%A9ng_d%E1%BB%A5ng). Ngoài ra, C cũng thường được dùng làm phương tiện giảng dạy trong [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh) mặc dù ngôn ngữ này không được thiết kế dành cho người nhập môn.

## Giao diện chính.



*Hình 6. Giao diện chính của chương trình.*

## Kết quả thực thi



*Hình 7. Hiển thị ma trận ra màn hình.*



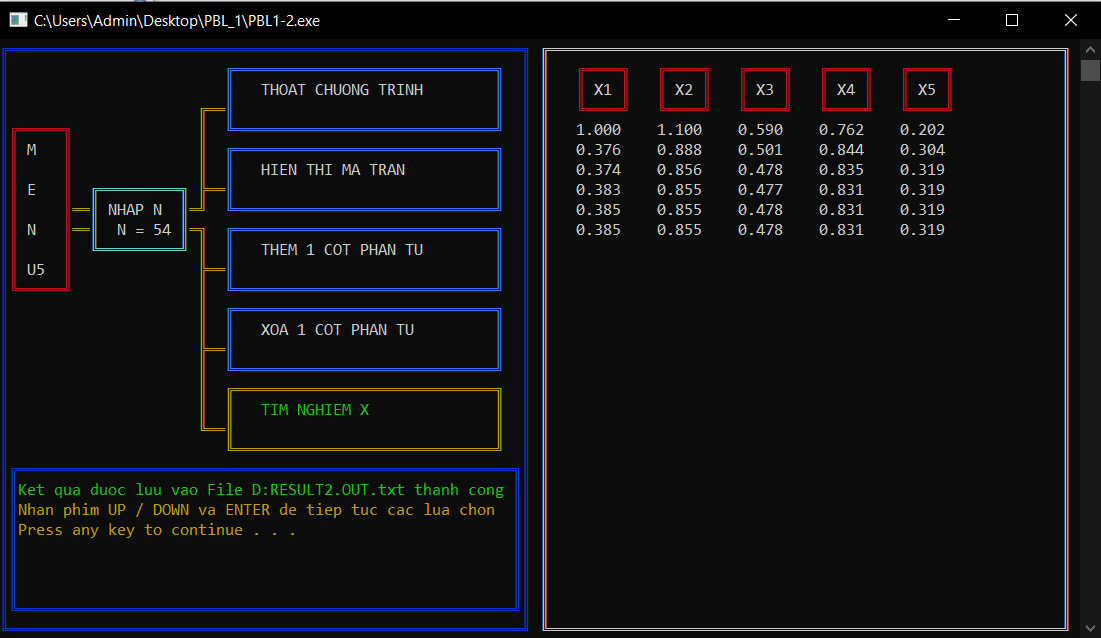
*Hình 8. Thực hiện thêm vào mỗi hàng ma trận một phần tử.*



*Hình 9. Thực hiện xóa mỗi hàng của ma trận một phần tử.*

**

*Hình 10. Giao diện chương trình khi tìm tập nghiệm X của hệ phương trình.*

**

*Hình 11. Bảng kết quả tập nghiệm qua các bước.*

## Nhận xét

Về tính năng, chương trình đã đảm bảo đầy đủ các yêu cầu mà đề bài đặt ra, thực hiện và giải đúng hệ phương trình bằng phương pháp lặp đơn, tuy nhiên, vẫn cần phải tối ưu nhiều nơi để tốc độ xử lý được nhanh hơn.

# KẾT LUẬN

## Kết quả đạt được

* Sau một thời gian nghiên cứu, thực nghiệm và hoàn thiện báo cáo đồ án, nhóm chúng em đã thực hiện đầy đủ những yêu cầu mà bài toán đề ra.
* Biết cách ứng dụng cấu trúc dữ liệu ngăn xếp, cũng như các kiến thức về phương pháp tính ứng dụng trong thực hành.
* Biết cách sử dụng đồ họa để trình bày chương trình một cách gọn gàng hơn.

## Hạn chế

* Phần chương trình được viết khá dài và còn chưa tối ưu về mặt thuật toán.
* Phần đồ họa vẫn chưa được hoàn chỉnh.

## Hướng phát triển

* Tiếp tục tìm hiểu để hoàn thiện những hạn chế bên trên, giảm độ phức tạp thuật toán.
* Tiếp cận các ngôn ngữ bậc cao hơn để tối ưu hóa việc sử dụng các tài nguyên thư viện.
* Chỉnh sửa và hoàn thiện phần đồ họa để có giao diện tốt hơn.

# TÀI LIỆU THAM KHẢO

[1] Đỗ Thị Tuyết Hoa, Bài giảng môn phương pháp tính (Dành cho sinh viên khoa Công nghệ thông tin), 2007.

[2] Danh sách liên kết, URL: <https://blog.luyencode.net/danh-sach-lien-ket-don/>

(Truy cập: 15/4/2023).

[3] Cấu trúc ngăn xếp, URL: <https://viblo.asia/p/cau-truc-du-lieu-va-giai-thuat-ngan-xep-stack-DbmemoJqvAg>

(Truy cập: 12/4/2023).

[4] Tạo giao diện đồ họa, URL: <https://www.youtube.com/watch?v=UjQIwlr_DqI>

( Truy cập: 1/6/2023).

# PHỤ LỤC

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

#include <string>

#define MAX 1000

//Khai Báo Cấu Trúc 1 Node

struct node{

double data;

struct node \*pNext;

};

typedef struct node NODE;

//Khai báo cấu trúc 1 stack

struct stack{

NODE \*pTop;//dùng con trỏ đầu để quản lí stack

};

typedef struct stack STACK;

//Khởi Tạo Stack

void KhoiTaoStack(STACK &s){

s.pTop = NULL;

}

//Khởi tạo 1 node

NODE \*KhoiTaoNode(double x){

NODE \*p = new NODE;

p -> data = x;

p -> pNext = NULL;

return p;

}

//Kiểm tra stack có rỗng hay không

bool IsEmpty(STACK s){

if(s.pTop == NULL) return true;

else return false;

}

//Thêm 1 phần tử vào đầu stack

bool Push(STACK &s, NODE \*p){

if(p == NULL) return false;

if(IsEmpty(s) == true){

s.pTop = p;

}

else{

p -> pNext = s.pTop;

s.pTop = p;

}

return true;

}

//Lấy phần tử đầu stack và trả về giá trị của nó đồng thời hủy nó đi

bool Pop(STACK &s, double &x ){

if(IsEmpty(s) == true) return false;

else{

NODE \*p = s.pTop;

x = p->data;

s.pTop = s.pTop->pNext;

delete p;

}

return true;

}

//Xem thông tin phần tử đầu danh sách mà không bị mất giá trị

bool Top(STACK &s, double &x){

if(IsEmpty(s) == true) return false;

else{

x = s.pTop->data;

}

return true;

}

//Đảo ngược giá trị của stack

void Reverse(STACK &s){

STACK temp;

KhoiTaoStack(temp);

for(NODE \*k = s.pTop; k != NULL; k = k -> pNext){

NODE \*p = KhoiTaoNode(k -> data);

Push(temp,p);

}

s = temp;

}

//Thêm 1 phần tử vào cuối

void PushBack(STACK &s, NODE \*p){

Reverse(s);

Push(s, p);

Reverse(s);

}

//Hàm truy đến giá trị của 1 node trong Stack

double Get(STACK &s, int index){

int x = 0;

NODE \*k = s.pTop;

while(x != index){

k = k -> pNext;

x++;

}

return k -> data;

}

//Hàm Ẩn Hiện Dấu Nháy Trên Màn Hình Console

void ShowCur(bool CursorVisibility)

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursor = { 1, CursorVisibility };

SetConsoleCursorInfo(handle, &cursor);

}

//Hàm Đưa Con Nháy Đến Địa Chỉ(x,y) Trên Màn Hình Console

void gotoxy(int x, int y)

{

static HANDLE h = NULL;

if(!h)

h = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD c = { x, y };

SetConsoleCursorPosition(h,c);

}

//Hàm Thay Đổi Màu Sắc Văn Bản

void SetColor ( int code ) {

HANDLE color = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute( color , code );

}

//Hamf Thay Đổi Màu Sắc BackGround

void TextColor(int x)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE) , x);

}

//Hàm Tạo 1 Hình Chữ Nhật Có Nội Dung

void Box(int x, int y, int w, int h, int t, int b, char \*nd){

TextColor(b);

for(int iy = y+1; iy <= y + h-1; iy++){

for(int ix = x + 1; ix <= x + w - 1; ix++){

gotoxy(ix, iy); printf(" ");

}

}

SetColor(b);

gotoxy(x + 1, y + 1); printf("%s", nd);

TextColor(1);

SetColor(t);

if(h <= 1|| w <= 1) return;

for(int ix = x; ix <= x + w; ix++){

gotoxy(ix, y); printf("%c", char(205));

gotoxy(ix, y + h); printf("%c", char(205));

}

for(int iy = y; iy <= y + h;iy++){

gotoxy(x, iy);printf("%c", char(186));

gotoxy(x + w, iy);printf("%c", char(186));

}

gotoxy(x, y); printf("%c", char(201));

gotoxy(x, y + h); printf("%c", char(200));

gotoxy(x + w, y); printf("%c", char(187));

gotoxy(x + w, y +h);printf("%c", char(188));

}

//Hàm Tạo n Hình Chữ Nhật Chứa Nội Dung Của Menu

void n\_Box(int x, int y, int w, int h, int t, int b){

char \*nd;

SetColor(7);

for(int i = 0;i < 5; i++){

if(i == 0) nd = " THOAT CHUONG TRINH";

else if(i == 1) nd = " HIEN THI MA TRAN";

else if(i == 2) {

nd = " THEM 1 COT PHAN TU";

}

else if(i == 4) nd = " TIM NGHIEM X";

else nd = " XOA 1 COT PHAN TU";

Box(x, y + (i \* 4), w, h, t, b, nd);

}

}

//Hàm In Ma Trân

void Print\_Matrix(STACK s[], int N, int y){

int v;

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

gotoxy(62, y + i);

v = 1;

for(NODE \*k = s[i].pTop; k != NULL; k = k -> pNext){

printf("%.3lf", k -> data);

v += 8;

gotoxy(62 + v, i + y);

}

}

}

//Hàm ghi File Ma Trận Sau Khi Thêm Mỗi Hàng 1 phần tử

int Write\_File1(char \*filename, STACK s[], int N){

FILE \*fp;

fopen\_s(&fp, filename, "w");

char c = '\n';

if(fp == NULL) return 0;

else{

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

for(NODE \*k = s[i].pTop; k != NULL; k = k->pNext){

fprintf( fp, "%.3lf ", k -> data);

}

fprintf( fp, "%c", c);

}

}

fclose(fp);

return 1;

}

//Hàm Ghi File Các nghiệm X của Hệ Phương Trình

int Write\_File2(char \*filename,double X[],int N){

FILE \*fp;

fopen\_s( &fp, filename, "w");

if(fp == NULL) return 0;

else{

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

fprintf(fp, "X%d=%.3lf\n", i+1, X[i]);

}

}

fclose(fp);

return 1;

}

//Hàm Đọc Ma Trận Từ File

int Read\_File(char \*filename, STACK s[], int N){

STACK k;

KhoiTaoStack(k);

FILE \*fp = NULL;

double x;

fopen\_s(&fp,filename,"r");

if(fp == NULL) return 0;

else{ while( fscanf( fp , "%lf " , &x) != EOF){//Đọc Hết Các Giá Trị trong File vào Stack k

NODE \*p = KhoiTaoNode(x);

Push( k , p );

}Reverse(k);//Hoán đổi ngược lại các vị trí

}fclose( fp );

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

for(int j = 0; j < (N-1); j++){

double v;

Pop(k, v);

NODE \*p = KhoiTaoNode(v);

Push(s[i], p);

}Reverse(s[i]);

}

return 1;

}

//Hàm Kiểm Tra Ma Trận Có Phải Ma Trận Vuông Hay Không

int CheckMatrix(STACK s[],int N){

int dem=0;

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

for(NODE \*k=s[i].pTop;k!=NULL;k=k->pNext){

dem++;

}

}

if(dem==N\*N) return 1;

else return 0;

}

// Hàm tính Định Thức Ma Trận Vuông Cấp N

double Det(STACK A[], int N){

double a[50][50];

//Gán Các Giá Trị Của Stack A cho mảng hai chiều

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

for(int j = 0; j < N; j++){

a[i][j] = Get(A[i],j);

}

}

int det = 1;

for(int i = 0; i < N-1; i++){

for(int k = i+1; k < N; k++){

if(a[k][i] == 0) continue;

if(a[k][i] != 0){

for(int j = 0; j < N; j++){

a[k][j] = - a[k][j] \* (a[i][i] / a[k][i]);

a[i][j] = a[i][j] + a[k][j];

int t = a[k][j];

a[k][i] = - a[i][j];

a[i][j] = t;

}

}

}

}

for(int i = 0; i < N; i++) det \*= a[i][i];

return det;

}

//Hàm Kiểm Tra Điều Kiện Hội Tụ Của Phương Pháp Lặp Gauss\_Siedel

int Gauss\_Terms(STACK A[], int N){

double r1[N], r2[N];

double max1, max2, max3;

double B[10][10];

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

for(int j = 0; j < N; j++){

if(i == j) B[i][j] = 0;

else{

B[i][j] = Get(A[i], j) / Get(A[i], i);

}

}

}

max1 = 0;

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

r1[i] = 0;

for(int j = 0; j < N; j++){

r1[i] += abs(B[i][j]);

}

if(max1 < r1[i] ) max1 = r1[i];

}

max2 = 0;

for(int j = 0; j < N; j++){

r2[j] = 0;

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

r2[j] += abs(B[i][j]);

}

if(max2 < r2[j]) max2 = r2[j];

}

max3 = 0;

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

for(int j = 0; j < N; j++){

max3 += B[i][j] \* B[i][j];

}

}

if(max1 < 1 || max2 < 1 || max3 < 1) return 1;

else return 0;

}

//Giao dien cua Menu

void Interface(){

Box(0,0,58,29,1,1," ");

Box(1,21,56,7,1,1," ");

Box(60,0,58,29,7,1," ");

n\_Box(25,1,30,3,9,7);

Box(10,7,10,3,11,7," NHAP N ");

SetColor(7);

gotoxy(12,9); printf(" N =");

Box(1,4,6,8,4,7," M");

SetColor(7);

gotoxy(2,7); printf(" E");

gotoxy(2,9);printf(" N");

gotoxy(2,11);printf(" U");

SetColor(6);

gotoxy(8,8); printf("%c%c",char(205),char(205));

gotoxy(8,9); printf("%c%c",char(205),char(205));

gotoxy(21,8); printf("%c", char(205));

gotoxy(21,9); printf("%c", char(205));

gotoxy(22,8);printf("%c",char(188));

gotoxy(22,9);printf("%c",char(187));

for(int i=3;i<=19;i++){

if (i==9||i==8) continue;

gotoxy(22,i); printf("%c",char(186));

}

gotoxy(22,3);printf("%c%c%c",char(201),char(205),char(205));

gotoxy(22,7);printf("%c%c%c",char(204),char(205),char(205));

gotoxy(22,11);printf("%c%c%c",char(204),char(205),char(205));

gotoxy(22,15);printf("%c%c%c",char(204),char(205),char(205));

gotoxy(22,19);printf("%c%c%c",char(200),char(205),char(205));

}

//Hàm tìm nghiệm X khi biết AX=B với A là ma trận đọc từ file và B là mảng nhập từ bàn phím

void Find\_X(STACK S[], int N){

STACK A[N]; //Sử Dụng 1 Stack Thay Thế

double X[N], Y[N], B[N];

int i, j;

for( i = 0; i < N; i++){ //Gán Stack Đã Đọc Từ File Qua Stack Thay Thế

KhoiTaoStack(A[i]);

A[i] = S[i];

}

//Kiểm Tra Điều Kiện Của Hệ Phương Trình

if(Det(A,N) == 0){

Box(1,21,56,7,1,1," ");

SetColor(4);

gotoxy(2,22); printf("MA TRAN KHONG CO NGHIEM DUY NHAT");

gotoxy(3,23);system("pause");

}

else{

Box(62,1,6,6,9,7,"NHAP");

SetColor(7);

gotoxy(63,4); printf("MANG");

gotoxy(63,6);printf("B");

for( i = 1; i <= N; i++){

Box(72,i\*3,15,2,11,1," ");

SetColor(11);

if(i!=1){

gotoxy(79,3\*i); printf("%c",char(202));

}

if(i!=N){

gotoxy(79,3\*i+2); printf("%c",char(203));

gotoxy(79,3\*i+3);printf("%c",char(186));

}

SetColor(7);

gotoxy(73,i\*3+1); printf("B[%d]=",i-1);

fflush( stdin);

gotoxy(80,i\*3+1);

scanf("%lf", &B[i-1]);

NODE \*p = KhoiTaoNode(B[i-1]);

PushBack(A[i-1],p);

}

system("cls");

Interface();

SetColor(7);

gotoxy(4,11);printf("%d",N);

gotoxy(18,9); printf("4");

Box(63,1,50,2,5,7,"He Phuong Trinh Co Dang Ma Tran");

gotoxy(88,3);printf("%c",char(203));

Box(61,4,56,N+1,2,1," ");

gotoxy(88,4);printf("%c",char(202));

SetColor(7);

Print\_Matrix(A,N,5); //Xuất Hệ Phương Trình

//Kiểm Tra Điều Kiện Hội Tụ

if(Gauss\_Terms(A, N) == 0){

Box(1,21,56,7,1,1," ");

SetColor(4);

gotoxy(2,22);

printf("ERROR: Khong Thoa Man Dieu Kien Hoi Tu");

gotoxy(2,23);

system("pause");

}

else{

for( i = 0; i < N; i++) X[i] = 0; //Cho tập nghiệm x ban đầu là 0 hết

int t;

double Sum;

Box(63,11,40,2,3,7,"Tap nghiem cua he phuong trinh");

for( i = 1; i <= N; i++){

if(i==1) SetColor(3);

else SetColor(4);

gotoxy(75,10+3\*i); printf("%c",char(203));

Box(68,N+6+3\*i,20,2,4,7," ");

SetColor(4);

gotoxy(75,11+3\*i); printf("%c",char(202));

SetColor(7);

gotoxy(69,N+7+3\*i);printf(" X %d = ", i);

}

do{

t = 0;

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

Sum = 0;

for(j = 0; j < (N+1); j++){

if(j != i) Sum = Sum + Get(A[i], j)\*X[j];

}

Y[i] = (Get(A[i],N) - Sum) / Get(A[i] , i);

if(abs(Y[i] - X[i]) >= 0.001) t = 1;

}

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

X[i] = Y[i];

}

}

while(t == 1);

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

gotoxy(77,N+10+3\*i);

SetColor(7);

printf("%.3lf", X[i]);

}

}

Box(1,21,56,7,1,1," ");

if(Write\_File2("RESULT2.OUT.txt", X, N) == 0){

Box(1,21,56,7,1,1," ");

SetColor(4);

gotoxy(2,22); printf("Error: Khong ton tai File RESULT2.OUT.txt");

}

else {

Box(1,21,56,7,1,1," ");

SetColor(10);

gotoxy(2,22); printf("Ket qua duoc luu vao File RESULT2.OUT.txt thanh cong");

}

SetColor(6);

gotoxy(2,23); printf("Nhan phim UP / DOWN va ENTER de tiep tuc cac lua chon");

gotoxy(2,24);

}

}

// Hàm Menu Chứa Và Thực Hiện Các Động Lệnh

void Menu(){

int N;

int x = 25;

int y = 1;

do{

Interface();

SetColor(10);

gotoxy(2,22);

printf("Doc ma tran gom N (N>=5) hang tu File D:\DAYSO.IN.txt");

gotoxy(17,9);

scanf("%d", &N);

if(N < 5) {

Box(1,21,56,7,1,1," ");

gotoxy(2,22);

SetColor(4);

printf("ERROR: N khong hop le");

gotoxy(2,23);

system("pause");

system("cls");

}

} while(N < 5);

STACK s[N];

for(int i = 0; i < N; i++) KhoiTaoStack(s[i]);

if( Read\_File("DAYSO.IN.txt", s, N) == 0){

Box(1,21,56,7,1,1," ");

gotoxy(2,22);

SetColor(4);

printf("ERROR: Loi file D:\DAYSO.IN.txt");

gotoxy(2,23);

system("pause");

}

else{

Interface();

Box(1,21,56,7,1,1," ");

SetColor(10);

gotoxy(2,22);printf("Ma tran doc tu File thanh cong");

SetColor(6);

gotoxy(2,22);printf("Nhan phim UP / DOWN va ENTER de tiep tuc cac lua chon");

SetColor(7);

gotoxy(17,9);printf("%d",N);

while(true){

ShowCur(0);

int e = 0;

int yp = y;

int ycu = yp;

n\_Box(25,1,30,3,9,7);

char \*nd =" THOAT CHUONG TRINH";

bool kt = true;

while(true){

if(kt == true){

Box(x,ycu,30,3,9,7,nd);

ycu = yp;

if(yp == 5) nd = " HIEN THI MA TRAN";

else if(yp == 9) nd = " THEM 1 COT PHAN TU";

else if(yp == 13) nd =" XOA 1 COT PHAN TU";

else if(yp == 17) nd =" TIM NGHIEM X";

else if(yp == 1) nd =" THOAT CHUONG TRINH";

Box(x,yp,30,3,6,10,nd);

kt=false;

}

if(kbhit()){

char c = getch();

if(c == 13) break;

else if(c ==-32 ){

kt = true;

c = getch();

if(c == 72 && yp != y){

if(yp != y) yp -= 4;

else{

yp = y +16;

}

}

else if(c == 80 && yp != y + 16){

if(yp != y + 9) yp += 4;

else{

yp = y;

}

}

}

}

}

ShowCur(1);

if(yp == 1){

gotoxy(2,29);

break;

}

else if(yp == 5) {

Box(62,2,15,2,6,7," MA TRAN");

gotoxy(67,4); printf("%c",char(203));

gotoxy(67,5); printf("%c",char(186));

gotoxy(68,4); printf("%c",char(203));

gotoxy(68,5); printf("%c",char(186));

Box(61,6,56,N+1,2,1," ");

SetColor(10);

gotoxy(67,6); printf("%c",char(202));

gotoxy(68,6); printf("%c",char(202));

SetColor(7);

Print\_Matrix(s,N,7);

Box(1,21,56,7,1,1," ");

SetColor(6);

gotoxy(2,22);

printf("Nhan phim UP / DOWN va ENTER de tiep tuc cac lua chon");

gotoxy(2,23);

fflush(stdin);

system("pause");

}

else if(yp == 9) {

Box(1,21,56,7,1,1," ");

SetColor(4);

gotoxy(2,22);

printf("VUI LONG NHAP SO THUC");

Box(1,21,56,7,1,1," ");

SetColor(10);

gotoxy(2,22);

printf("Cac phan tu se duoc them vao dau moi hang cua ma tran");

gotoxy(60,1);

SetColor(3);

Box(62,1,40,2,5,7,"Nhap cac phan tu can them vao ma tran");

SetColor(6);

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

double value;

for( i = 1; i <= N; i++){

if(i==1) SetColor(5);

else SetColor(6);

gotoxy(70,3\*i); printf("%c",char(203));

Box(62,1+i\*3,28,2,6,7," ");

gotoxy(70,3\*i+1);printf("%c",char(202));

SetColor(7);

gotoxy(64,2+3\*i); printf("Phan Tu Thu %d = ",i);

fflush( stdin);

gotoxy(80,2+3\*i);

scanf("%lf", &value);

NODE \*p = KhoiTaoNode(value);

Push(s[i-1], p);

}

}

Box(1,21,56,7,1,1," ");

SetColor(6);

gotoxy(2,22);

printf("Nhan phim UP / DOWN va ENTER de tiep tuc cac lua chon");

gotoxy(2,23);

}

else if(yp==13){

Box(1,21,56,7,1,1," ");

SetColor(10);

gotoxy(2,22); printf("Xoa phan tu dau tien moi hang cua ma tran");

double z;

for(int i=0;i<N;i++) Pop(s[i],z);

SetColor(6);

gotoxy(2,23);

printf("Nhan phim UP / DOWN va ENTER de tiep tuc cac lua chon");

gotoxy(2,24);

system("pause");

}

else if(yp==17){

if(CheckMatrix(s,N)==0){

Box(1,21,56,7,1,1," ");

gotoxy(2,22);

SetColor(4);

printf("ERROR:Ma tran hien khong phai ma tran vuong");

gotoxy(2,23);

}

else{

Box(1,21,56,7,1,1," ");

SetColor(10);

gotoxy(2,22); printf("He phuong trinh co dang AX=B voi B nhap tu ban phim");

gotoxy(60,1);

Find\_X(s, N);

}

system("pause");

}

Box(60,0,58,29,7,1," ");

}

}

}

int main(){

Menu();

}