BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**THỰC TẬP CƠ SỞ**

**VIẾT CHƯƠNG TRÌNH MINH HỌA CÔNG VIỆC ĐỂ ĐẠT SỐ LƯỢNG CÔNG VIỆC TỐI ĐA ĐƯỢC THỰC HIỆN BỞI MỘT NGƯỜI**

**Giảng viên phụ trách: Hà Thị Thanh Ngà**

**Sinh viên thực hiện: Trần Ngọc Duy Long**

**Lớp: 61CNTT-2**

**Mã số sinh viên: 61133256**

**LỜI CẢM ƠN**

Để hoàn thành bài báo cáo thực tập cơ sở này, trước hết em xin gửi đến quý thầy, cô Khoa Công nghệ Thông tin - Trường Đại học Nha Trang lời cảm ơn chân thành.

Em xin gửi đến cô Hà Thị Thanh Ngà, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành bài báo cáo này lời cảm ơn sâu sắc nhất.

Trong quá trình nghiên cứu thực hiện đề tài, cũng như là trong quá trình làm bài báo cáo thực tập cơ sở, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do kiến thức cũng như kinh nghiệm thực tiễn của bản thân còn hạn chế nên đề tài, bài báo cáo cơ sở này khó thể không tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp từ quý thầy, cô để em học hỏi thêm được nhiều kinh nghiệm, cũng như kỹ năng cần thiết.

Em xin chân thành cảm ơn!

MỤC LỤC

[Chương 1. TỎNG QUAN VẤN ĐỀ THỰC TẬP 4](#_Toc92525202)

[1.1 CƠ SỞ LÍ THUYẾT 4](#_Toc92525203)

[1.1.1 Tổng quan về phần mềm lập trình Visual Studio 4](#_Toc92525204)

[1.1.2 Một số tính năng của phần mềm Visual Studio 4](#_Toc92525205)

[1.2 CÁC NGÔN NGỮ ĐƯỢC SỬ DỤNG 7](#_Toc92525206)

[1.2.1 Ngôn ngữ C/C++ 7](#_Toc92525207)

[Chương 2. Chương 2. ĐẶC TẢ BÀI TOÁN 8](#_Toc92525208)

[2.1 Bài toán 8](#_Toc92525209)

[2.2 Hướng xử lí 8](#_Toc92525210)

[Chương 3. Chương 3. CÀI ĐẶT CHƯƠNG TRÌNH 11](#_Toc92525211)

[3.1 Chương trình code 11](#_Toc92525212)

[3.1.1 Mức 1: 11](#_Toc92525213)

[3.1.2 Mức 2.1 18](#_Toc92525214)

[3.1.3 Mức 2.2 20](#_Toc92525215)

# TỎNG QUAN VẤN ĐỀ THỰC TẬP

# **CƠ SỞ LÍ THUYẾT**

### Tổng quan về phần mềm lập trình Visual Studio

Phần mềm lập trình Visual Studio là một phần mềm hỗ trợ đắc lực hỗ trợ công việc lập trình website rất nổi tiếng hiện nay của Microsoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một cách dễ dàng và nhanh chóng nhất thông qua Visual Studio.

Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ Microsoft. Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sử mã, trình thiết kế, gỡ lỗi. Người dùng có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với phần mềm Visual Studio. Người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend.

### **Một số tính năng của phần mềm Visual Studio**

#### Biên tập mã

Giống như bất kỳ một IDE khác, Visual Studio gồm có một trình soạn thảo mã hỗ trợ tô sáng cú pháp và hoàn thiện mã bằng cách sử dụng IntellSense không chỉ cho các hàm, biến và các phương pháp mà còn sử dụng cho các cấu trúc ngôn ngữ như: Truy vấn hoặc vòng điều khiển.

Các trình điều khiển biên tập mã Visual Studio cũng chỗ trợ cái đặt dấu trang trong mã để có thể điều hướng một cách nhanh chóng và dễ dàng. Hỗ trợ các điều hướng như: Thu hẹp các khối mã lệnh, tìm kiếm gia tăng,…

Visual còn có tính năng biên dịch nền tức là khi mã đang được viết thì phần mềm này sẽ biên dịch nó trong nền để nhằm cung câp thông tin phản hồi về cú pháp cũng như biên dịch lỗi và được đánh dấu bằng các gạch gợn sóng màu đỏ.

**1.1.2.2 Trình gỡ lỗi**

Visual Studio có một trình gỡ lỗi có tính năng vừa lập trình gỡ lỗi cấp máy và gỡ lỗi cấp mã nguồn. Tính năng này hoạt động với cả hai mã quản lý giống như ngôn ngữ máy và có thể sử dụng để gỡ lỗi các ứng dụng được viết bằng các ngôn ngữ được hỗ trợ bởi Visual Studio.

**1.1.2.3 Thiết kế**

* **Windows Forms Designer**

Được sử dụng với mục đích xây dựng GUI sử dụng Windows Forms, được bố trí dùng để xây dựng các nút điều khiển bên trong hoặc cũng có thể khóa chúng vào bên cạnh mẫu. Điều khiển trình bày dữ liệu có thể được liên kết với các nguồn dữ liệu như: Cơ sở dữ liệu hoặc truy vấn.

#### WPF Designer

Tính năng này cũng giống như Windows Forms Designer có công dụng hỗ trợ kéo và thả ẩn dụ. Sử dụng tương tác giữa người và máy tính nhằm mục tiêu vào Windows Presentation Foundation.

#### Thiết kế, phát triển Web

Visual Studio cũng có một trình soạn thảo và thiết kế website cho phép các trang web được thiết kế theo tính năng kéo và thả đối tượng. Mục đích là để hỗ trợ người dùng tạo trang web dễ dàng hơn, những yêu cầu đơn giản như thiết kế web du lịch hay các trang giới thiệu của công ty có thể sử dụng tính năng này vì nó vẫn đảm bảo cho bạn sở hữu được một website hoàn chỉnh.

#### Mảng hai chiều

Trong lập trình, có những trường hợp cần lưu trữ dữ liệu dưới dạng ma trận gồm nhiều dòng và nhiều cột. Do đó, cần một cấu trúc dữ liệu để lưu trữ ma trận. Đó là mảng hai chiều. Mảng hai chiều lưu trữ các phần tử theo dạng bảng gồm dòng và cột.

#### Cách khai báo và khởi tạo mảng hai chiều

Hình 1. Hình minh họa mảng hai chiều

* **Cú pháp khai báo**

<kiểu dữ liệu> <tến biến mảng> [<Số dòng>] [<Số cột>]

* **Khởi tạo mảng hai chiều**

Khởi tạo như mảng một chiều nhưng có khai báo số dòng và số cột.

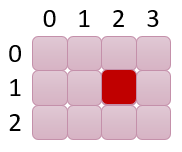
|  |
| --- |
| //Khởi tạo  Int temp[4][3] = {50, 70, 60, 48, 75, 62, 51, 69, 60}  //Kết quả  50 70 60  48 75 62  51 69 60 |

* Truy xuất đến một phần tử trong mảng hai chiều

Thông qua chỉ số hàng và chỉ số cột trong mảng. Cú pháp:

<tên biến mảng> [<cs1>] [<cs2>]

Ví dụ: Truy xuất phần tử dòng 1 cột 2 là temp[1][2]



Hình 2. Vị trí giá trị temp[1][2] trong mảng hai chiều

# **CÁC NGÔN NGỮ ĐƯỢC SỬ DỤNG**

### Ngôn ngữ C/C++

Ngôn ngữ lập trình C++ được phát triển từ ngôn ngữ C. Nó chính là mở rộng của ngôn ngữ C, có nghĩa là nói được bổ sung các đặc tính mới của ngôn ngữ lập trình hướng đối tượng, Trước đó ngôn ngữ C là ngôn ngữ bậc cao được biết đến với khả năng chạy được trên hầu hết các máy tính và độc lập với phần cứng.

* **Các thành phần trong C++**

<fstream>: Định nghĩa các lớp hỗ trợ các phép toán cho dòng xuất nhập (iostream) trên các chuỗi chưa trong các tập tin bên ngoài.

<ios>: Định nghĩa nhiều kiểu và hàm cơ bản cho các xử lý của các dòng xuất nhập. Tiêu đề này thường được bao gồm nhiều tiêu đề dòng cuất nhập khác mà ít khi được khai báo qua lệnh #include trực tiếp.

<iostream>: Định nghĩa nhận đầu vào từ bàn phím và xuất dữ liệu ra console. Phần io của iostream là viết tắt của input / output.

# Chương 2. ĐẶC TẢ BÀI TOÁN

# **Bài toán**

Viết một chương trình minh họa việc chọn lựa công việc để đạt số lượng công việc tối đa được thực hiện bởi một người. Giả sử mỗi người chỉ có thể làm một việc tại một thời điểm.

Mục đích: Thể hiện giải thuật dưới dạng chương trình hoàn chỉnh

Yêu cầu:

Mức 1:

1. Trình bày giải thuật thực hiện chương trình

2. Sử dụng ngôn ngữ: Pascal/C/C++ để thể hiện

- Input: Nhập n (n<=20) số hoạt động và mỗi hoạt động gồm thời gian bắt đầu và thời gian kết thúc.

- Cấu trúc dữ liệu: Mảng hai chiều

- Output: Các hoạt động cùng thời gian bắt đầu và thời gian kết thúc.

Mức 2:

Đọc số hoạt động và mỗi hoạt động gồm thời gian bắt đầu và thời gian kết thúc từ file văn bản chứa input.

Xuất kết quả ra màn hình và lưu vào file.

# **Hướng xử lí**

Mức 1: Áp dụng Mảng hai chiều được miêu tả ở mục 1.1.3

* **Ý tưởng:**

Với việc minh họa công việc để đạt số lượng công việc tối đa được thực hiện bởi một người. Đầu vào là một chuỗi các hoạt động ngẫu nhiên gồm thời gian bát đầu và thời gian kết thúc với điều kiện :

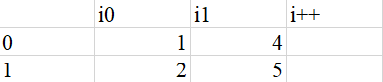
* Số lượng hoạt động không quá 20.
* Thời gian kết thúc phải lớn hơn thời gian bắt đầu.
* Thời gian bắt đầu phải nhỏ hơn hoặc bằng 23 và thời gian kết thúc phải nhỏ hơn hoặc bằng 24.

Đầu ra là chuỗi các hoạt động phù hợp có thời gian nối tiếp liên tục.

* **Thực hiện:**

Nhập một chuỗi các hoạt động n với thời gian bát đầu a và thời gian kết thúc b ngẫu nhiên. Trong đó:

* n <= 20, nếu sai nhập lại n.
* a <= 23, nếu sai nhập lại a.
* b <= 24, nếu sai nhập lại b.

 Lưu thời gian bắt đầu va thời gian kết thúc vào mảng arr (mảng hai chiều). a tương ứng với hàng 0, b tương ứng với hàng 1. Ví dụ arr[0][i0] = 1, arr[1][i0] = 3, arr[0][i1] = 4, arr[1][i1] = 5.

Hình 3. Minh họa lưu giá trị vào mảng arr

Đề sắp xếp các hoạt động, em sử dụng điều kiện arr[0][i1] > arr[1][i0]. Nói cách khác, thời gian bắt đầu của hoạt động sau phải lớn hơn thời gian bắt đầu của hoạt động trước. Khi thỏa điều kiện hoạt động sau sẽ nối tiếp với hoạt động trước và lấy chính hoạt động đó xét điều kiện cho hoạt động tiếp theo theo một vòng lặp đến khi kết thúc. Nếu không thỏa điều kiện thì hoạt động sau sẽ được loại bỏ và hoạt động trước tiếp tục xét điều kiện cho các hoạt dộng tiếp theo.

Mức 2: Sử dụng thư viện fstream được miêu tả ở mục 1.2.1 đề lấy dữ liệu từ file hoặc xuất kết quả từ dữ liệu đã nhập ra file.

* Ý tưởng:
* Lấy dữ liệu trong file text đã nhập sẵn để sắp xếp.
* Xuất kết quả từ dữ liệu nhập ra file text.
* Thực hiện:
* Nhập dữ liệu trong file text bao gồm số lượng hoạt động và thời gian bắt đầu, kết thúc của mỗi hoạt động. Lấy dữ liệu vừa nhập từ trong file text bằng tính năng <ifstream> của thư viện <fstream>.
* Xuất kết quả các hoạt động ra file text từ dữ liệu nhập trong bảng console bằng tính năng <ofstream> của thư viện <fstream>.

# Chương 3. CÀI ĐẶT CHƯƠNG TRÌNH

# **Chương trình code**

### Mức 1:

Khai báo thư viện C/C++, thư viện <fstream> để lấy dữ liệu từ file và xuất kết quả ra file.

|  |
| --- |
| *#include<stdio.h>*  *#include<fstream>*  *#include<iostream>*  *using namespace std;* |

Hàm nhaphoatdong() có chức năng nhận số lượng hoạt động (n) nhập vào từ bảng console. Vòng lặp while diễn ra nếu số lượng hoạt động (n) nhập vào lớn hơn 20.

|  |
| --- |
| *void nhaphoatdong(int &n)*  *{*  *printf("Nhap so luong hoat dong\n");*  *scanf\_s("%d",&n);*  *while (n > 20) {*  *printf("Nhap so luong hoat dong be hon 20\n");*  *scanf\_s("%d", &n);*  *}*  *}* |

Hàm nhapthoigian() có chức năng nhận giá trị thời gian bắt đầu (a) và thời gian kết thúc (b) của mỗi hoạt động được lặp lại trong vòng lặp for tăng dần theo i. Vòng lặp while diễn ra nếu thời gian bắt đầu lớn hơn 23 vì thời gian kết thúc lớn hơn thời gian bắt đầu nên giá trị của thởi gian bắt đầu không thể bằng 24. Vòng lặp while tiếp theo sẽ diễn ra khi thời gian kết thúc lớn hơn 24. Vòng lặp while cuối cùng sẽ diễn ra khi thời gian kết thúc nhỏ hơn hoặc bằng thời gian bắt đầu. Sau đó hệ thống sẽ lưu giá trị thời gian bắt đầu và thời gian kết thúc của mỗi hoạt động vào mảng arr (mảng hai chiều).

|  |
| --- |
| *void nhapthoigian(int arr[3][21], int n, int &a, int &b)*  *{*  *for (int i = 0; i < n;i++)*  *{*  *printf("\nHoat dong %d\n", i + 1);*  *printf("Nhap thoi gian bat dau: ");*  *scanf\_s("%d", &a);*  *while (a > 23)*  *{*  *printf("Thoi gian bat dau phai nho hon 24: ");*  *scanf\_s("%d", &a);*  *}*  *printf("Nhap thoi gian ket thuc: ");*  *scanf\_s("%d", &b);*  *while (b > 24)*  *{*  *printf("Thoi gian ket thuc phai nho hon 25: ");*  *scanf\_s("%d", &b);*  *}*  *while (a >= b)*  *{*  *printf("Thoi gian ket thuc phai lon hon > thoi gian bat dau: ");*  *scanf\_s("%d", &b);*  *}*  *arr[0][i] = a;*  *arr[1][i] = b;*  *}*  *}* |

Hàm sapxep1() có chức năng xuất kết quả ra màn hình console và file File.text. Điều kiện sắp xếp các hoạt động là thời gian bắt đầu của hoạt động sau phải lớn hơn thời gian kết thúc của hoạt động trước bằng điều kiện if. Nếu thỏa điều kiện hoạt động sau tức hoạt động arr[0][j], arr[1][j] sẽ nối tiếp với hoạt động trước tức hoạt động arr[0][i], arr[1][i]. Nếu không thỏa hệ thống sẽ loại bỏ hoạt động đó và xét tiếp hoạt động tiếp theo. Cuối cùng gán i = j để xét các hoạt động tiếp theo bằng vòng lặp for.

|  |
| --- |
| *void sapxep1(int arr[3][21], int &n, ofstream &File)*  *{*  *printf("(%d,%d)", arr[0][0], arr[1][0]);*  *File<<"("<<arr[0][0]<<"," << arr[1][0]<<")"<<endl;*  *for (int i = 0;i < n;i++)*  *{*  *for (int j = i;j < n;j++)*  *{*  *if (arr[0][j] > arr[1][i] )*  *{*  *printf("(%d,%d)", arr[0][j], arr[1][j]);*  *File << "(" << arr[0][j] << "," << arr[1][j] << ")"<<endl;*  *i = j;*  *}*  *}*  *}*  *}* |

Hàm sapxep2() có chức năng xuất kết quả ra màn hình console. Điều kiện sắp xếp các hoạt động là thời gian bắt đầu của hoạt động sau phải lớn hơn thời gian kết thúc của hoạt động trước bằng điều kiện if. Nếu thỏa điều kiện hoạt động sau tức hoạt động arr[0][j], arr[1][j] sẽ nối tiếp với hoạt động trước tức hoạt động arr[0][i], arr[1][i]. Nếu không thỏa hệ thống sẽ loại bỏ hoạt động đó và xét tiếp hoạt động tiếp theo. Cuối cùng gán i = j để xét các hoạt động tiếp theo bằng vòng lặp for.

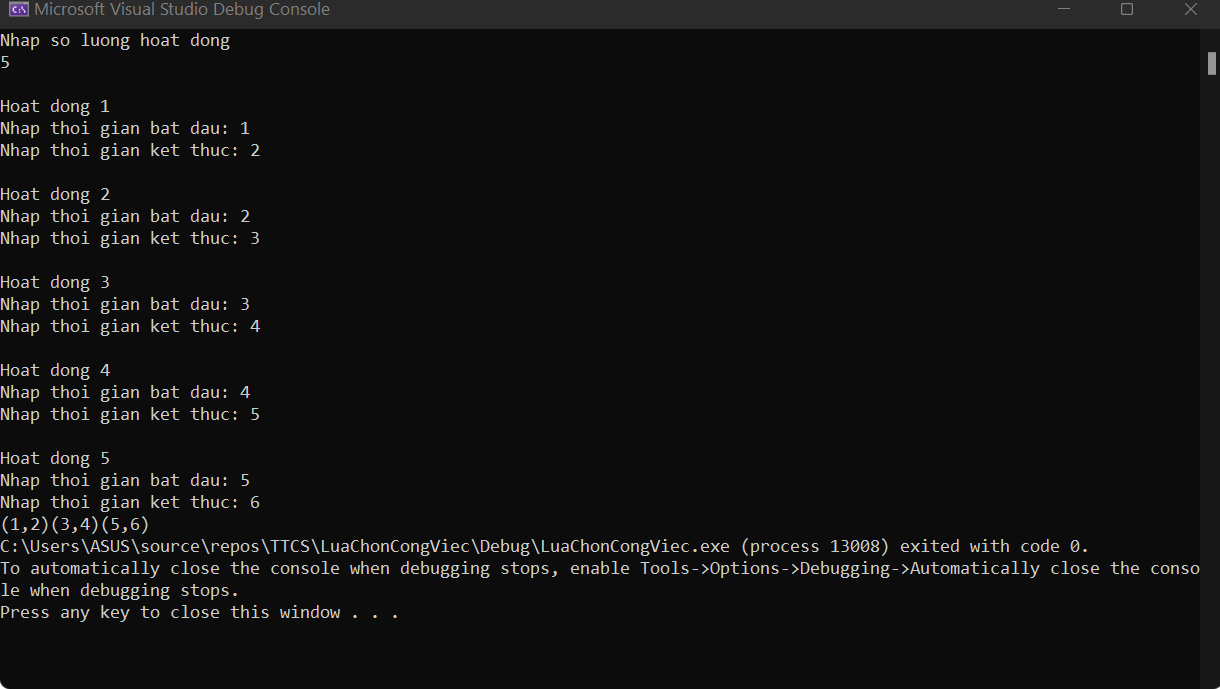
|  |
| --- |
| *void sapxep2(int arr[3][21], int& n, ifstream& File)*  *{*  *printf("(%d,%d)", arr[0][0], arr[1][0]);*  *for (int i = 0;i < n;i++)*  *{*  *for (int j = i;j < n;j++)*  *{*  *if (arr[0][j] > arr[1][i])*  *{*  *printf("(%d,%d)", arr[0][j], arr[1][j]);*  *i = j;*  *}*  *}*  *}*  *}* |

Hàm chương trình chính:

|  |
| --- |
| *int main()*  *{*  *int n, a, b;*  *int arr[3][21];*  *int i, j;*  *nhaphoatdong(n);*  *nhapthoigian(arr, n, a, b);*  *sapxep1(arr,n,File);*  *File.close();*  *}* |

Code chương trình hoàn chỉnh:

|  |
| --- |
| *#include<stdio.h>*  *#include<fstream>*  *#include<iostream>*  *using namespace std;*  *void nhaphoatdong(int &n)*  *{*  *printf("Nhap so luong hoat dong\n");*  *scanf\_s("%d",&n);*  *while (n > 20) {*  *printf("Nhap so luong hoat dong be hon 20\n");*  *scanf\_s("%d", &n);*  *}*  *}*  *void nhapthoigian(int arr[3][21], int n, int &a, int &b)*  *{*  *for (int i = 0; i < n;i++)*  *{*  *printf("\nHoat dong %d\n", i + 1);*  *printf("Nhap thoi gian bat dau: ");*  *scanf\_s("%d", &a);*  *while (a > 23)*  *{*  *printf("Thoi gian bat dau phai nho hon 24: ");*  *scanf\_s("%d", &a);*  *}*  *printf("Nhap thoi gian ket thuc: ");*  *scanf\_s("%d", &b);*  *while (b > 24)*  *{*  *printf("Thoi gian ket thuc phai nho hon 25: ");*  *scanf\_s("%d", &b);*  *}*  *while (a >= b)*  *{*  *printf("Thoi gian ket thuc phai lon hon > thoi gian bat dau: ");*  *scanf\_s("%d", &b);*  *}*  *arr[0][i] = a;*  *arr[1][i] = b;*  *}*  *}*  *void sapxep1(int arr[3][21], int &n, ofstream &File)*  *{*  *printf("(%d,%d)", arr[0][0], arr[1][0]);*  *File<<"("<<arr[0][0]<<"," << arr[1][0]<<")"<<endl;*  *for (int i = 0;i < n;i++)*  *{*  *for (int j = i;j < n;j++)*  *{*  *if (arr[0][j] > arr[1][i] )*  *{*  *printf("(%d,%d)", arr[0][j], arr[1][j]);*  *File << "(" << arr[0][j] << "," << arr[1][j] << ")"<<endl;*  *i = j;*  *}*  *}*  *}*  *}*  *void sapxep2(int arr[3][21], int& n, ifstream& File)*  *{*  *printf("(%d,%d)", arr[0][0], arr[1][0]);*  *for (int i = 0;i < n;i++)*  *{*  *for (int j = i;j < n;j++)*  *{*  *if (arr[0][j] > arr[1][i])*  *{*  *printf("(%d,%d)", arr[0][j], arr[1][j]);*  *i = j;*  *}*  *}*  *}*  *}*  *int main()*  *{*  *int n, a, b;*  *int arr[3][21];*  *int i, j;*  *nhaphoatdong(n);*  *nhapthoigian(arr, n, a, b);*  *sapxep2(arr,n,File);*  *File.close();*  *}* |



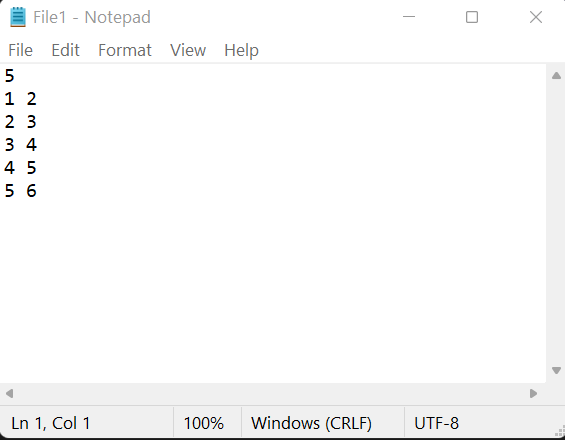
Hình 4. Kết quả bảng console mức 1

### Mức 2.1

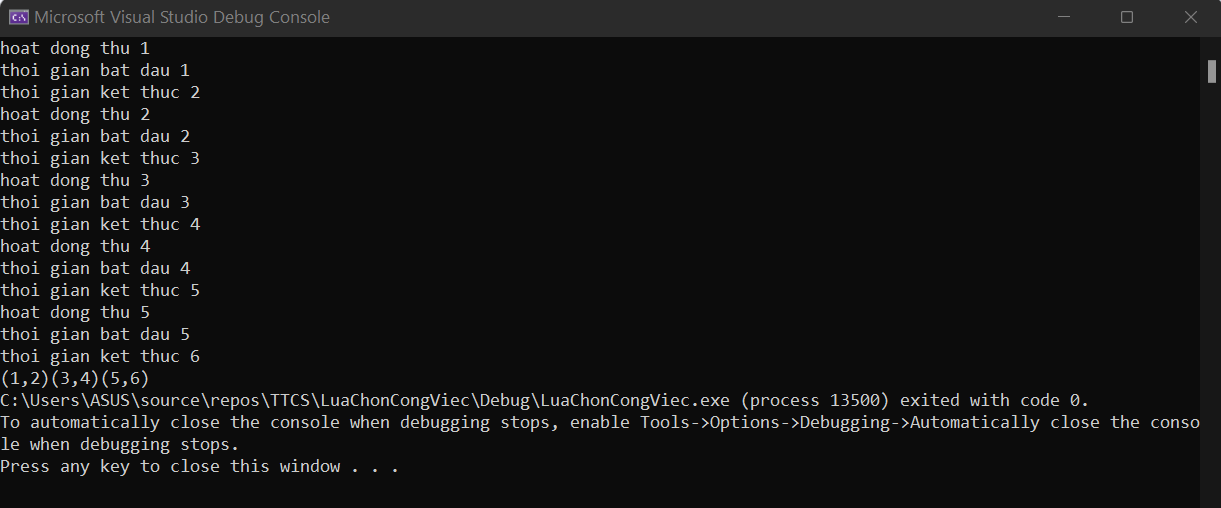
Đọc số lượng hoạt động gồm thời gian bắt đầu và thời gian kết thúc từ file File1.text

Hàm chương trình chính:

|  |
| --- |
| *int main() {*  *ifstream File1;*  *File1.open("File1.txt");*  *int n, a, b;*  *int arr[3][21];*  *int i, j;*  *File1 >> n;*    *for (i = 0; i < n; i++)*  *{*  *File1 >> a, File1 >> b;*  *cout << "hoat dong thu " << i + 1 << endl;*  *cout << "thoi gian bat dau " << a << endl;*  *cout << "thoi gian ket thuc " << b << endl;*  *arr[0][i] = a;*  *arr[1][i] = b;*  *}*  *sapxep2(arr, n, File1);*  *File1.close();*  *}* |



Hình 5. Lấy dữ liệu từ file text



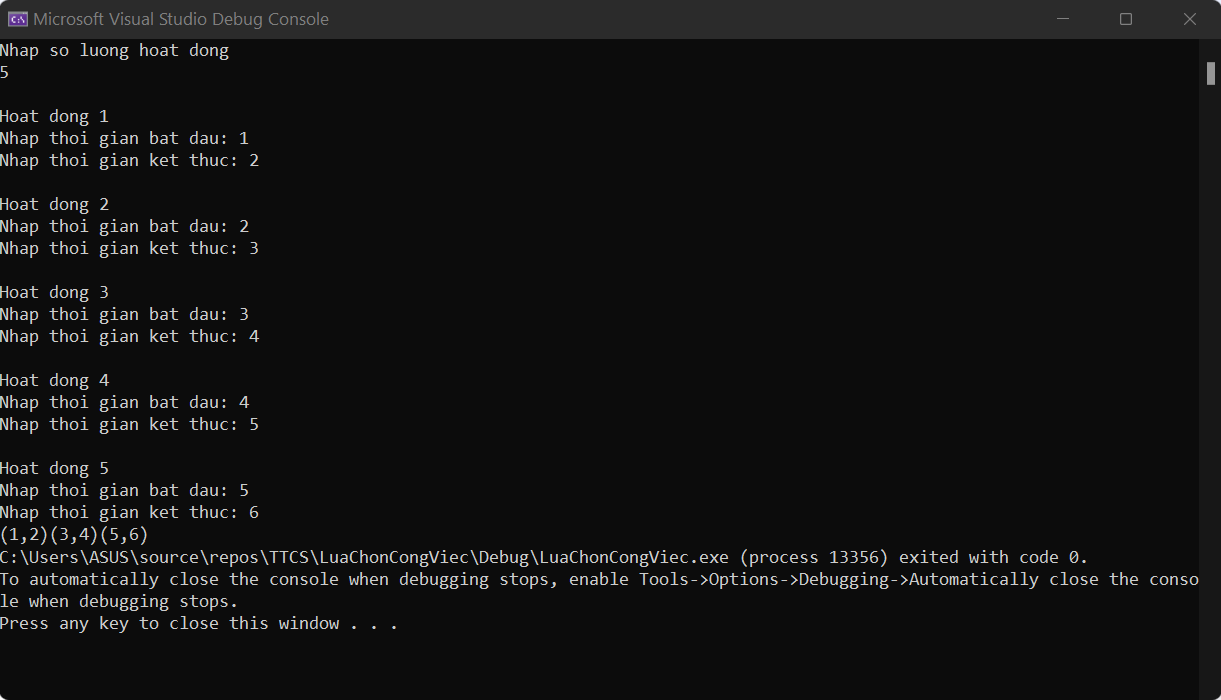
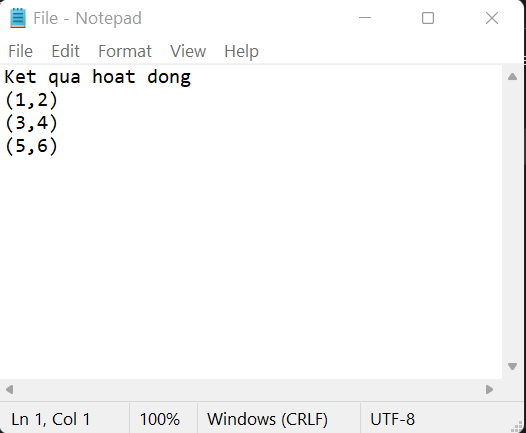
Hình 6. Kết quả bảng console sau khi lấy dữ liệu từ file text

### Mức 2.2

Xuất kết quả từ bảng console ra file File.txt.

Hàm chương trình chính:

|  |
| --- |
| *int main()*  *{*  *int n, a, b;*  *int arr[3][21];*  *int i, j;*  *nhaphoatdong(n);*  *nhapthoigian(arr, n, a, b);*  *ofstream File("File.txt");*  *File << "Ket qua hoat dong\n";*  *sapxep1(arr,n,File);*  *File.close();*  *}* |



Hình 7. Dữ liệu trong file text sau khi xuất kết quả từ bảng console

Hình 8. Kết quả bảng console