**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**ĐỀ TÀI: NHẬP/XUẤT VÀ HIỂN THỊ THÔNG TIN THÀNH PHỐ THEO YÊU CẦU**

Giảng viên hướng dẫn: TRẦN THỊ DUNG

Sinh viên thực hiện: TRẦN HOÀNG NAM

TRƯƠNG CÔNG THẠCH

LÊ MINH TUẤN

Lớp: CQ.60.CNTT

Khoá:60

Tp. Hồ Chí Minh, năm 2020

CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

**1.1 Lý do chọn đề tài**

Trong xã hội ngày càng phát triển hiện nay, khoa học công nghệ là thứ không thể thiếu đối với mỗi quốc gia, doanh nghiệp, trường học hay mỗi cá nhân, đặc biệt là công nghệ thông tin. Với sự phát triển nhanh một cách không ngừng nghĩ như vậy của công nghệ thông tin đã giúp giải quyết các công việc học tập, nguyên cứu, quản lý thông tin,… một cách dễ dàng và tiện lợi. Thấy được tiềm năng đó các quốc gia, doanh nghiệp, trường học, các cá nhân, … đã ứng dụng nó vào thực tiển cuộc sống để giải quyết công việc, học tập, giải trí với những chiếc điện thoại thông minh nhỏ gọn.

Trong những năm gần đây nhu cầu về các phần mềm, ứng dụng học tập, giải trí càng nhiều hơn do nhu cầu sử dụng điện thoại các thiết bị gọn nhẹ ngày càng cao và thuận tiện cho người sử dụng. Các thiết bị thông minh đã và đang được sử dụng rộng rãi trên toàn thế giới từ trẻ em cho đến cụ già ai cũng có ít nhất là một chiếc điện thoại thông minh. Nắm bắt được tình hình đó hàng loạt nhà sản xuất đã đánh vào thị trường ứng dụng mobile để phát triển.

Đặc biệt trong học tập và giải trí, sự tiện lợi của những chiếc điện thoại thông minh lại chứng minh ưu thế của mình. Người sử dụng có thể học tập và giải trí ở bất kì đâu mà họ muốn chỉ thông qua một chiếc điện thoại di động có kết nối với Internet. Đã có rất nhiều ứng dụng đã ra đời và đem lại nhiều lợi ích cho người sử dụng, bạn không cần phải đến thư viện để đọc sách hoặc tốn nhiều thời gian cho việc tìm kiếm thông tin, hoặc là bạn không cần đến trường, trung tâm để học chỉ cần cài được ứng dụng bạn có thể gói gọn mọi thứ trong chiếc điện thoại của bạn. Và ngay cả việc theo dõi tên, cũng như các thành phố mà bạn cần nghiêng cứu, đều rất đơn giản.

**1.2 Mục tiêu nghiên cứu:**

**-** Xây dựng ứng dụng nhập xuất dữ liệu các thành phố bằng ngôn ngữ C.

- Ứng dụng phải đáp ứng được giao diện thân thiện bắt mắt với người dùng, đầy đủ các chức năng,dễ sử dụng.

**1.3 Phạm vi nghiên cứu:**

-Nghiên cứu về các thuật toán sắp xếp, các cấu trúc dữ liệu và các hàm, các dữ liệu nhập và xuất.

**1.4 Khảo sát thực tế:**

Theo dõi dữ liệu các thành phố 1 cách trực quan và dễ hiểu, dễ tìm kiếm theo sở thích.

**1.5 Cách tiếp cận, phương pháp nguyên cứu:**

-Tìm hiểu các thành phố, các thông tin cần thiết, các hàm cần thiết, nhập xuất dữ liệu, và sắp xếp.

**1.6 Cấu trúc báo cáo đồ án thực tập chuyên môn:**

**1.6.1 Chương 1: Tổng quan về đề tài**

- Lý do chọn đề tài.

- Mục tiêu nghiên cứu.

- Phạm vi nghiên cứu.

- Khảo sát thực tế.

- Cách tiếp cận và phương pháp nghiên cứu.

**1.6.2 Chương 2: Bài toán nhập xuất thành phố theo yêu cầu**

**-** Giới thiệu bài toán

- Cở sở lý thuyết cho bài toán

- Quy trình

**1.6.3 Chương 3: Phân tích bài toán và thiết kế hệ thống**

- Công nghệ sử dụng

- Phân tích thiết kế hệ thống

**1.6.4 Kết luận và kiến nghị**

**1.6.5 Tài liệu tham khảo**

CHƯƠNG 2. BÀI TOÁN VỀ NHẬP XUẤT THÀNH PHỐ THEO YÊU CẦU

2.1. Giới thiệu bài toán:

Làm theo yêu cầu đã được giao, qua đó nhập vào và xuất ra, đồng thời sắp xếp các thành phố theo như ý muốn của đề bài.

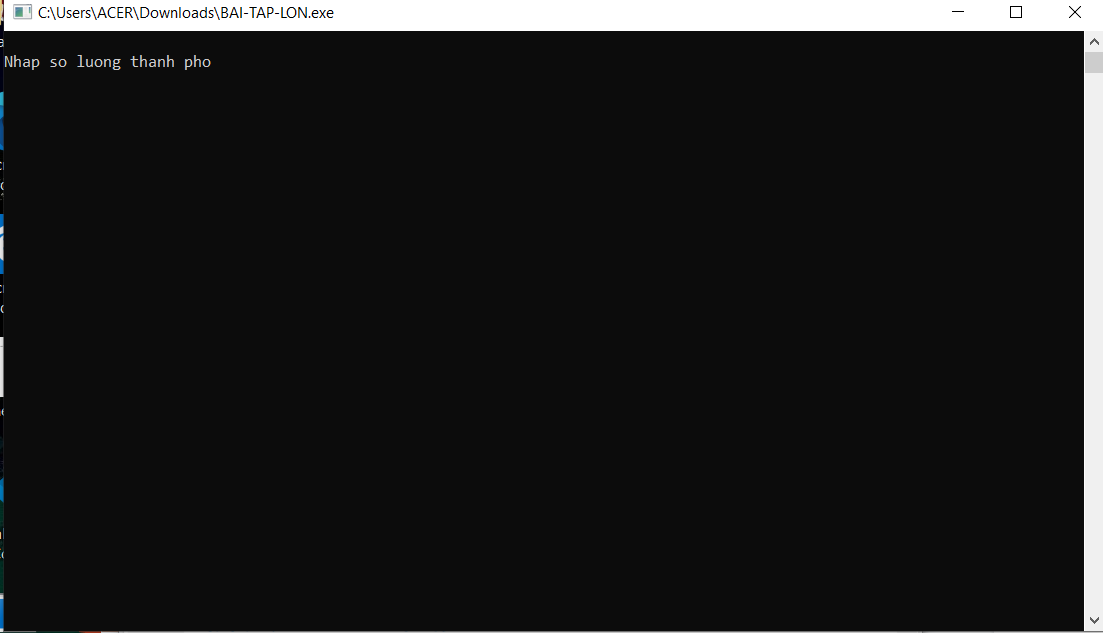
2.2 Mô tả và cài đặt giải thuật cho bài toán

-Sử dụng hàm struct, từ đó khai báo và áp dụng vào lập danh sách các thành phố được nhập vào, xây dựng menu từ các case, và từ đó tiếp tục xây dựng các yêu cầu của bài toán.

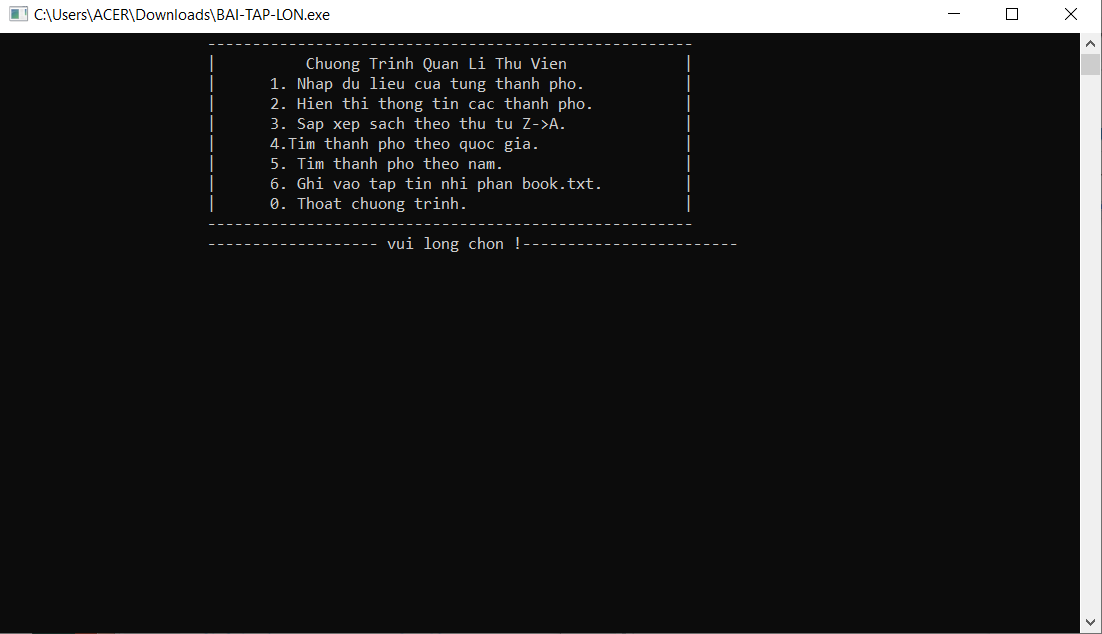
-Sử dụng void để nhập, xuất các thông tin của thành phố theo yêu cầu, đồng thời làm việc với tệp.

2.3 Hướng dẫn sử dụng

-Sau khi chạy chương trình sẽ được yêu cầu nhập vào số thành số cần



-Sau đó sẽ có menu hiện ra như ảnh, qua đó nhập yêu cầu cần làm theo số ghi trên menu.



CHƯƠNG 3. LÝ THUYẾT

3.1 Làm việc với tệp

Khi làm việc với file, cần khai báo 1 con trỏ kiểu FILE.

FILE \*fptr;

Để đọc ghi file trong C cũng như trong mọi ngôn ngữ lập trình, việc đầu tiên cần làm là mở file mà bạn muốn làm việc. Trong ngôn ngữ lập trình C, chúng ta có thể mở file bằng cách sử dụng hàm fopen() trong thư viện stdio.h như sau:

fptr = fopen("fileopen","mode")

Trong đó mode là một tham số chúng ta cần chỉ định.

Ví dụ:

fptr = fopen("E:\\cprogram\\newprogram.txt","w");

// hoặc

fptr = fopen("E:\\cprogram\\oldprogram.bin","rb");

Giả sử tập tin newprogram.txt chưa có trong thư mục E:\cprogram. Ví dụ đầu tiên với mode = "w" sẽ cho phép chương trình tự động tạo ra file newprogram.txt nếu nó chưa có. Và sau đó mở file này lên nhưng chương trình chỉ có thể ghi dữ liệu vào mà không thể đọc.

Mode là w chỉ cho phép chương trình ghi(nếu đã có dữ liệu thì ghi đè) nội dung của file.

Với ví dụ thứ 2, mode là rb cho phép chương trình mở 1 file nhị phân đã có sẵn oldprogram.bin. Với trường hợp này, chương trình của bạn chỉ có thể đọc file và không thể ghi nội dung vào file.

Các tham số của “mode”

Dưới đây là các giá trị có thể có của tham số mode  nói trên:

|  |  |  |
| --- | --- | --- |
| **Mode** | **Ý nghĩa** | **Nếu file không tồn tại** |
| r | Mở file chỉ cho phép đọc | Nếu file không tồn tại, fopen() trả về NULL. |
| rb | Mở file chỉ cho phép đọc dưới dạng nhị phân. | Nếu file không tồn tại, fopen() trả về NULL. |
| w | Mở file chỉ cho phép ghi. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| wb | Open for writing in binary mode. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| a | Mở file ở chế độ ghi “append”. Tức là sẽ ghi vào cuối của nội dung đã có. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| ab | Mở file ở chế độ ghi nhị phân “append”. Tức là sẽ ghi vào cuối của nội dung đã có. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| r+ | Mở file cho phép cả đọc và ghi. | Nếu file không tồn tại, fopen() trả về NULL. |
| rb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. | Nếu file không tồn tại, fopen() trả về NULL. |
| w+ | Mở file cho phép cả đọc và ghi. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| wb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| a+ | Mở file cho phép đọc và ghi “append”. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| ab+ | Mở file cho phép đọc và ghi “append” ở dạng nhị phân. | Nếu file không tồn tại, nó sẽ được tạo tự động. |

Thao tác đóng file:

fclose(fptr);

Đọc/ghi file văn bản:

sử dụng fprintf() và fscanf().

Ví dụ sử dụng fprintf()

#include <stdio.h>

#include <stdlib.h>

int main()

{

   int num;

   FILE \*fptr;

   fptr = fopen("C:\\program.txt","w");

   if(fptr == NULL)

   {

      printf("Error!");

      exit(1);

   }

   printf("Enter num: ");

   scanf("%d",&num);

   fprintf(fptr,"%d",num);

   fclose(fptr);

   return 0;

}

Đọc file sử dụng fscanf()

#include <stdio.h>

#include <stdlib.h>

int main()

{

   int num;

   FILE \*fptr;

   if ((fptr = fopen("C:\\program.txt","r")) == NULL){

       printf("Error! opening file");

       // Program exits if the file pointer returns NULL.

       exit(1);

   }

   fscanf(fptr,"%d", &num);

   printf("Value of n=%d", num);

   fclose(fptr);

   return 0;

}

Đọc/Ghi file nhị phân

Các hàm fread() và fwrite() trong C được sử dụng để đọc và ghi file trong C ở dạng nhị phân.

Ghi file nhị phân sử dụng fwrite()

#include <stdio.h>

#include <stdlib.h>

struct threeNum

{

   int n1, n2, n3;

};

int main()

{

   int n;

   struct threeNum num;

   FILE \*fptr;

   if ((fptr = fopen("C:\\program.bin","wb")) == NULL){

       printf("Error! opening file");

       // Program exits if the file pointer returns NULL.

       exit(1);

   }

   for(n = 1; n < 5; ++n)

   {

      num.n1 = n;

      num.n2 = 5\*n;

      num.n3 = 5\*n + 1;

      fwrite(&num, sizeof(struct threeNum), 1, fptr);

   }

   fclose(fptr);

   return 0;

}

Ví dụ đọc file nhị phân sử dụng fread()

#include <stdio.h>

#include <stdlib.h>

struct threeNum

{

   int n1, n2, n3;

};

int main()

{

   int n;

   struct threeNum num;

   FILE \*fptr;

   if ((fptr = fopen("C:\\program.bin","rb")) == NULL){

       printf("Error! opening file");

       // Program exits if the file pointer returns NULL.

       exit(1);

   }

   for(n = 1; n < 5; ++n)

   {

      fread(&num, sizeof(struct threeNum), 1, fptr);

      printf("n1: %d\tn2: %d\tn3: %d", num.n1, num.n2, num.n3);

   }

   fclose(fptr);

   return 0;

}

**Các giải thuật tìm kiếm , sắp xếp :** **\*Bài toán Tìm kiếm:**

Ví dụ cho dãy n số nguyên a0,a1,a2,..a(n-1) và một số nguyên x.Hãy tìm xem x có thuộc vào dãy số trên hay không. Nếu tìm được ở vị trí i thì xuất kết quả là I ; ngược lại nếu hông tìm thấy thì xuất kết quả là -1(chú ý dãy bắt đầu từ chỉ số 0, nếu dãy có nhiều số bằng x thì xuất vị trí I nhỏ nhất nếu cần )

Sau đây là thuật toán tìm kiếm thường được sử dụng nhất có tên gọi là tìm kiếm tuyến tính .Ý tưởng chính của thuật toán tìm kiếm như sau : Bắt đầu từ phần tử thứ a[0] ,ta lần lượt so sánh x với các giá trị a[i].Nếu có a[i] bằng x thì I chính là kết quả cần tìm và kết thúc thuật toán .Nếu trong dãy không có số a[i] nào bằng x thì xuất kết quả là -1 và cũng kết thúc thuật toán.

Đoạn code thể hiện ý tưởng tìm kiếm tuyến tính như hàm sau :

**int timkiem(int a[], int n ,int x)**

**{**

**int i=0;**

**while(i<n &&a[i]!=x)**

**i++;**

**if(i==n)**

**return -1;//tìm kiếm nhưng không có x**

**return i;//tìm thấy a[i] là phần tử có khóa ( giá trị )x**

**}**

**Bài toán Sắp xếp :**

**-**Trong các thuật toán sắp xếp , thuật toán đơn gỉn nhất đó là phương pháp sắp xếp đổi chỗ trực tiếp .

-Xét một mảng n số a0 ,a1,a(n-1). Nếu i<j và ai>aj,thì ta gọi đó là một nghịch thế .Mảng chưa sắp xếp sẽ có nghịch thế và ngược lại mảng đã có thứ tự sẽ không còn nghịch thế . Để sắp xếp một mảng ,ta có thể tìm cách làm giảm số các nghịch thế trong mảng này bằng cách hoán vị các phần tử ai,aj nếu có i<j,ai>aj (theo thuật toán hoán vị )

Đoạn code thê hiện ý tưởng sắp xếp đổi chỗ trực tiếp như hàm sau :

//đầu tiên cho viết hàm hoán vị

**void hoanvi(int &a,int &b)**

**{**

**int temp=0;**

**a=b;**

**b=temp;**

**}**

**void sapxep(int a[],int n)**

**{**

**for(int i=0;i<n;i++)**

**for(int j=i+1;j<n;j++)**

**if(a[i]>a[j] hoanvi(a[i],a[j])**

**}**

+Trong hàm hoanvi này các biến a,b phải được truyền theo kiểu tham biến .Hàm sắp xếp trên sẽ sắp xếp các phần tử theo chìu tăng dần , nếu cần sắp xếp giảm dần thì phép so sánh trong câu lệnh if(a[i]>a[j] sẽ được đổi lại là if(a[i]<a[j]).