

TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA CÔNG NGHỆ THÔNG TIN



യമാ



ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN

TÊN ĐỀ TÀI XÂY DỰNG ỨNG DỤNG QUẢN LÍ DANH SÁCH SINH VIÊN

Người hướng dẫn: ThS. NGUYỄN THỊ MINH HỶ

Sinh viên thực hiện:

Tên sinh viên 1: PHAM VĂN TIẾN TRƯỞNG

LÓP: 20TCLC-DT4

Tên sinh viên 2:PHAM CÔNG HUY

LÓP: 20TCLC-DT4



ớ Đà Nẵng, 04/2021 &



MỤC LỤC

MỤC LỤC 2

DANH MỤC HÌNH VỄ	3
LỜI NÓI ĐẦU	3
1. TỔNG QUAN ĐỀ TÀI:	1
2. CƠ SỞ LÝ THUYẾT	1
2.1. Ý tưởng:	1
2.2. Cơ sở lý thuyết:	2
3. TỔ CHÚC CẦU TRÚC DỮ LIỆU VÀ THUẬT TOÁN	6
3.1. Phát biểu bài toán	6
3.2. Cấu trúc dữ liệu:	7
3.3. Thuật toán	8
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ	18
4.1. Tổ chức chương trình:	18
4.2. Ngôn ngữ cài đặt:	20
4.3. Kết quả	20
4.3.1. Giao diện chính của chương trình	20
4.3.2. Kết quả thực thi của chương trình	24
4.3.3. Nhận xét đánh giá:	27
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	28
5.1. Kết luận:	28
5.2. Hướng phát triển:	28
- Tiếp thu ý kiến đóng góp và phản hồi của mọi người để phát triển chức năng và giao diện.	
TÀI LIỆU THAM KHẢO	29
PHU LUC	30

DANH MỤC HÌNH VỄ

1.	Hình 1: Minh Họa Kiêu dữ liệu struct	Trang 2
2.	Hình 2: Minh Họa Tìm Kiếm Tuần Tự	Trang 3
3.	Hình 3: Minh Họa Tìm Kiếm Nhị Phân.	Trang 4
4.	Hình 4: Minh Họa Sắp Xếp Quick Sort.	Trang 5
5.	Hình 5: Minh Họa Cánh Xóa Một Phần Tử Trong mảng	Trang 6
6.	Hình 6: Quá Trình Tìm Kiếm Một Sinh Viên	Trang 8
7.	Hình 7: Quá Trình Tìm Kiếm Sinh Viên theo Thuật Toán Tìm kiếm N	lhị Phân
		Trang 9
8.	Hình 8: Quá Trình Tìm Kiếm Tuần Tự	Trang 11
9.	Hình 9: Lưu Đồ Thuật Toán Phân Đoạn Mảng	Trang 14,15
10.	Hình 10: Menu chính của chương trình	Trang 20
11.	Hình 11: Menu lựa chọn cánh thêm sinh viên	Trang 21
12.	Hình 12: Màn hình nhập thông tin sinh viên trực tiếp	Trang 21
13.	Hình 13: Màn hình nhập thông tin từ file.	Trang 22

14.	Hình 14: Màn hình nhập thông tin để tìm kiếm hoặc xóa	Trang 22
15.	Hình 15: Màn hình hiện kết quả khi tìm thấy sinh viên	Trang 23
16.	Hình 16: màn hình lựa chọn xóa sinh viên	Trang 23
17.	Hình 17: Màn hình hiển thị danh sách sinh viên	Trang 24
18.	Hình 18: Dữ liệu đầu vào là danh sách sinh viên	Trang 24
19.	Hình 19: Kết quả tìm kiếm theo tên.	Trang 24
20.	Hình 20: Kết quả tìm kiếm theo mã sinh viên	Trang 25
21.	Hình 21: Kết quả xóa sinh viên theo tên	Trang 25
22.	Hình 22: Kết quả xóa sinh viên theo mã sinh viên	Trang 26
23.	Hình 23: Kết quả sau khi sắp xếp.	Trang 26
24.	Hình 24: Kết quả sau khi cấp mã sinh viên và email	Trang 26
25.	Hình 25: Kết quả khi in ra file.	Trang 27

LỜI NÓI ĐẦU

Ngày nay công nghệ thông tin đóng vai trò cực kỳ quan trọng và trở thành một phần không thể thiếu trong đời sống. Việc ứng dụng một cách rộng rãi vào mọi lĩnh vực đã đem lại hiệu quả, năng suất công việc khá cao. Điều đó đòi hỏi ngày càng cải tiến công nghệ, tối ưu hóa thuật toán để phát triển nhiều tính năng hơn nữa. Có rất nhiều công cụ, môn học để giải quyết vấn đề này. Một trong những môn học nền tảng quan trọng ảnh hưởng trực tiếp đến thuật toán đó là "*Cấu trúc dữ liệu*".

Để nghiên cứu kỹ hơn và xây dựng hợp lý cấu trúc dữ liệu và thuật toán, nhóm chúng em đã chọn đề tài về "Xây dựng ứng dụng quản lý danh sách sinh viên.".

Mục đích thực hiện đề tài: Áp dụng được kiến thức môn học cấu trúc dữ liệu vào đồ án, luyện tập khả năng làm việc nhóm.

Mục tiêu: Hiểu và trình bày được cấu trúc dữ liệu và phần code của đồ án.

Phạm vi và đối tượng nghiên cứu: Đối tượng là sinh viên, phạm vi nghiên cứu gồm: Quản lí thông tin sinh viên, sắp xếp danh sách, tìm kiếm (xóa) sinh viên, cấp mã sinh viên và tên email, cập nhật sinh viên.

Phương pháp nghiên cứu: Kết hợp kiến thức môn học cấu trúc dữ liệu, hiểu biết của bản thân và tham khảo thêm thông tin trên internet.

Ứng dụng quản lý học sinh – sinh viên đang là phần mềm mang lại nhiều lợi ích cho việc quản lý phù hợp sử dụng và hỗ trợ cho các giảng viên, giáo viên giúp quản lý các học sinh, học viên của mình được tốt hơn.

Trong quá trình thực hiện, mặc dù đã có nhiều cố gắng song không tránh khỏi những thiếu sót, chúng em rất mong nhận được sự chỉ dẫn, đóng góp của quý thầy cô để đề tài của chúng em ngày càng hoàn thiện hơn.

Đồng thời, chúng em cũng gửi lời cảm ơn chân thành đến

Cô Nguyễn Thị Minh Hỷ đã giúp đỡ chúng em hoàn thành đề tài này.

1. TỔNG QUAN ĐỀ TÀI:

1.1, Mục đích của đề tài

Phần mềm hỗ trợ phòng giáo vụ trong công tác quản lý dữ liệu sinh viên, môn học, giảng viên, dễ dàng thống kê dữ liệu khi cần thiết. Bài toán " Quản lý Sinh Viên" nhằm giải quyết và đáp ứng một cách hiệu quả các nhu cầu về mặt quản lý thông tin trong các trường Đại Học.

1. 2, Ý nghĩa của đề tài

Tin học hoá trong công tác quản lý nhằm giảm bớt sức lao động của con người, tiết kiệm được thời gian, độ chính xác cao, gọn nhẹ và tiện lợi hơn rất nhiều so với việc làm thủ công quản lý trên giấy tờ như trước đây. Tin học hoá giúp thu hẹp không gian lưu trữ, tránh được thất lạc dữ liệu, tự động hệ thống hoá và cụ thể hoá các thông tin theo nhu cầu của con người.

Bên cạnh đó, thông qua đề tài này giúp chúng em phát triển các kỹ năng: phân tích, thiết kế, lựa chọn cấu trúc dữ liệu và xây dựng thuật toán, lập trình,...

2. CƠ SỞ LÝ THUYẾT

2.1. Ý tưởng

- Dùng kiểu dữ liệu Struct để định nghĩa một cấu trúc sinh viên, và một mảng struct để lưu trữ danh sách sinh viên.
 - Cấu trúc sinh viên gồm:
 - + Họ Lót
 - + Tên
 - + Giới Tính
 - + Ngày Sinh
 - + Địa Chỉ
 - + Mã Sinh Viên
 - + Email
- Công việc mà chúng ta cần làm ở mỗi yêu cầu:
 - + Thêm sinh viên: Dùng các hàm nhập chuỗi để nhập thông tin sinh viên

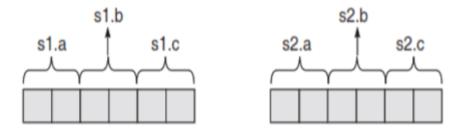
- + Tìm Kiếm Sinh Viên: Chúng ta sẽ tìm kiếm bằng các thuật toán tìm kiếm như: Nhị phân ,Tuần tự .
- + Xóa Sinh Viên: Chúng ta cần tìm kiếm sinh viên và xóa sinh viên đó bằng cách dịch chuyển mảng lùi 1 chỉ số phần phía sau nơi xóa.
- + Sắp xếp danh sách sinh viên: Đầu tiên chúng ta sẽ sắp xếp tên theo Alpha Beta bằng phương pháp quick sort sau đó sẽ sắp xếp các sinh viên có tên trùng nhau theo họ lót bằng phương pháp Quick Sort.
- + Xuất danh sách: Sẽ dùng vòng lặp để hiện thị lên màn hình thông tin của từng sinh viên.
- + Cấp mã sinh viên: Ban Đầu Mỗi Sinh Viên Sẽ Có Mã Sinh Viên Là 1022000, sau khi sắp xếp thì mã sinh viên sẽ tự động cấp 3 kí tự cuối theo thứ tự sau khi sắp xếp.
- + Cấp email: Sẽ dùng hàm xử lí chuỗi để ghép mã sinh viên với chuỗi "@gmail.com".

2.2. Cơ sở lý thuyết

2.2.1, Giới thiệu về kiểu dữ liệu struct

Structure trong C (cấu trúc trong C) là một kiểu dữ liệu người dùng tự định nghĩa (user defined datatype) cho phép bạn lưu trữ các loại phần tử khác nhau.

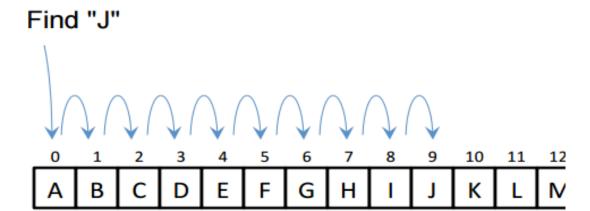
Mỗi phần tử của một cấu trúc được gọi là một thành viên (member).



Hình 1: Minh hoa kiểu dữ liêu struct.

2.2.2, Các phương pháp tìm kiếm

2.2.2.1, **Tìm kiếm tuyến tính**: Là kiểm tra tuần tự từng phần tử của mảng, đến khi nào giống thì thôi



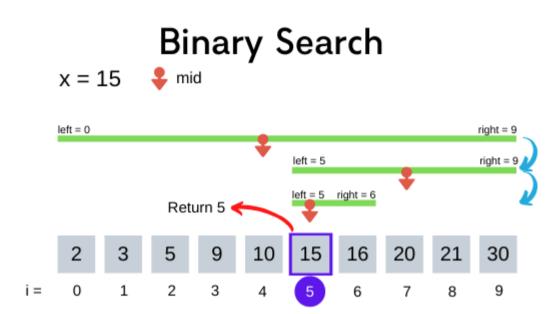
Hình 2: Minh Họa Tìm Kiếm Tuần Tự.

```
int \ linear Search \ (int \ a[], \ int \ n, \ int \ x) \ \{ for \ (int \ i = 0; \ i < n; \ i++) \ \{ if \ (a[i] == x) \ return \ i; else \ return \ -1; \}
```

2.2.2.2, **Tìm kiếm nhị phân:** Điều kiện của thuật toán này là mảng đã được sắp xếp tăng dần. So sánh x với giá trị của phần tử nằm ở giữa mảng (mid=(left+right)/2). Nếu x nhỏ hơn a[mid] thì nó chỉ có thể nằm ở nửa bên trái, ngược lại x lớn hơn a[mid] thì x nằm ở nửa bên phải. Xác định x nằm ở nửa nào thì ta lặp lại thuật toán với nửa đó. Như vậy số lần kiểm tra giảm đi nhiều do không phải mất công kiểm tra những phần tử thuộc nửa còn lại.

```
int binarySearch (int a[], int n, int x) {
    int left = 0, right = n - 1, mid;
    do {
        mid = (left + right) / 2;
        if (a[mid] == x) return mid;
        else if (a[mid] <= x) left = mid + 1;
        else right = mid - 1;
    } while (left <= right);
    return -1;</pre>
```

}

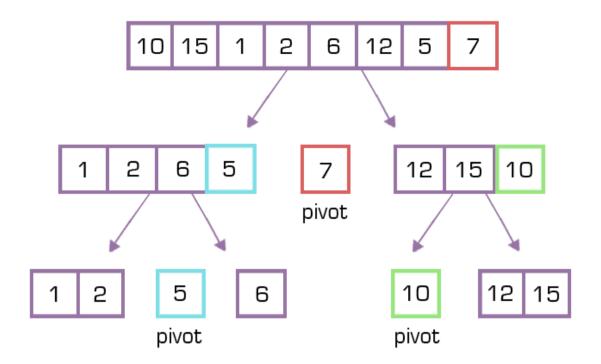


Hình 3: Minh Họa Tìm Kiếm Nhị Phân.

2.3 Thuật toán sắp xếp Quick Sort:

- Chọn một phần tử trong dãy và gọi nó là phần tử chốt p (pivot).
- Chia dãy đã cho ra thành hai dãy con: Dãy con trái (*L*) gồm những phần tử không lớn hơn phần tử chốt, còn dãy con phải (*R*) gồm các phần tử không nhỏ hơn phần tử chốt. Thao tác này được gọi là thao tác phân đoạn (Partition).

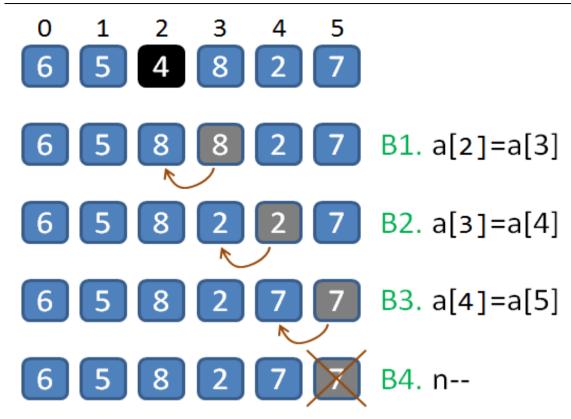
Lặp lại một cách đệ qui thuật toán đối với hai dãy con L và R.



Hình 4: Minh họa sắp xếp Quick Sort.

2.4, Thuật toán xóa một phần tử

- + Đầu tiên sẽ xác định vị trí của sinh viên cần xóa.
- + Dịch chuyển mảng lùi 1 chỉ số phần phía sau nơi xóa.
- + Giảm số lượng phần tử.



Hình 5: Minh họa cánh xóa một phần tử trong mảng

2.5 Các kiến thức đã học được ở học phần kỹ thuật lập trình và cấu trúc dữ liệu.

3. TỔ CHỨC CẦU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1. Phát biểu bài toán

- Input: Nhập thủ công hoặc đọc file chứa số sinh viên cần thêm vào và thông tin cá nhân từng sinh viên. Chương trình cho phép quản lý khoản 100 sinh viên/mỗi lớp.
- Output:
 - Cung cấp thông tin sinh viên mỗi một lớp, sẽ được lưu trên 1 file.
 - Chức năng thêm sinh viên: nhập họ lót, tên, ngày sinh, giới tính, địa chỉ.

- Chức năng Cấp Mã Sinh Viên chương trình tự động cấp sau khi sắp xếp danh sách theo thứ tự Alphabe của Tên và Họ lót. Nếu chưa sắp xếp chương trình sẽ thông báo lỗi.
- Chức năng Địa Chỉ Email chương trình tự động cấp sau khi cấp mã sinh viên. Nếu chưa cấp mã chương trình sẽ thông báo lỗi.
- Chức năng sắp xếp danh sách.
- Chức năng tìm kiếm và xóa sinh viên.

3.2. Cấu trúc dữ liệu

Dùng kiểu dữ liệu Struct để để định nghĩa một cấu trúc sinh viên, và một mảng Struct để lưu trữ danh sách sinh viên.

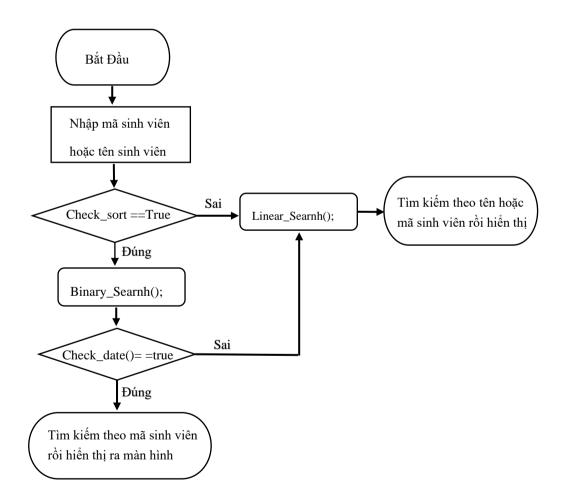
Tổ chức lưu trữ dữ liệu:

```
+ Định nghĩa một cấu trúc sinh viên:
    struct first_and_last_name{
        char first_name[25];
        char last_name[25];
    };
    struct student{
        first_and_last_name name;
        char gender[10];
        char Date_of_birth[15];
        char address[50];
        char student_code[10];
        char email[25];
};
```

- + Int total_number_of_students: Dùng để lưu tổng số sinh viên trong lớp.
- + Bool check_find , bool check_innit , bool check_sort , bool check_provide_student_code , bool check_provide_email: Lần lượt dùng để lưu trạng thái của các hoạt động tìm kiếm, khởi tạo lớp, sắp xếp, cấp mã sinh viên, cấp email.
 - + Int location_to_delete: Dùng để lưu vị trí sinh viên cần xóa đã tìm thấy.

3.3. Thuật toán

3.3.1 Tìm Kiếm Sinh Viên



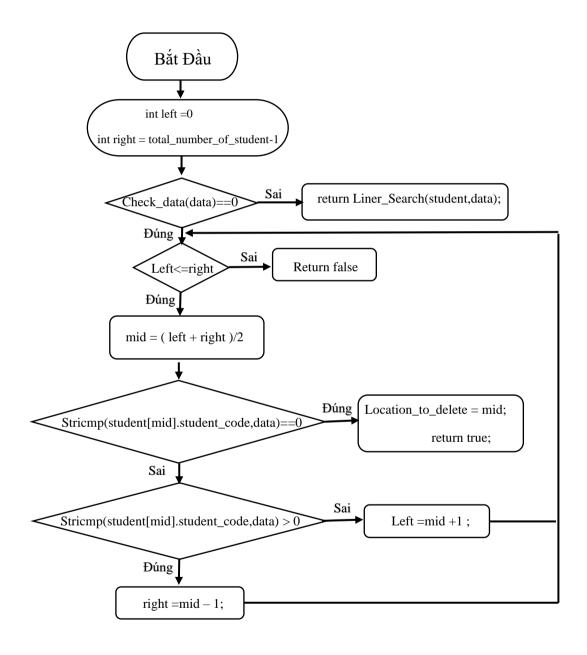
Hình 6: Quá Trình Tìm Kiếm Một Sinh Viên.

Step 1: Nhập dữ liệu.

Step 2 : Kiểm tra danh sách đã sắp xếp chưa.

Step 3: Tìm kiếm theo thuật toán thích hợp.

a, Tìm kiếm Nhị Phân (độ phức tạp: Log2(n))

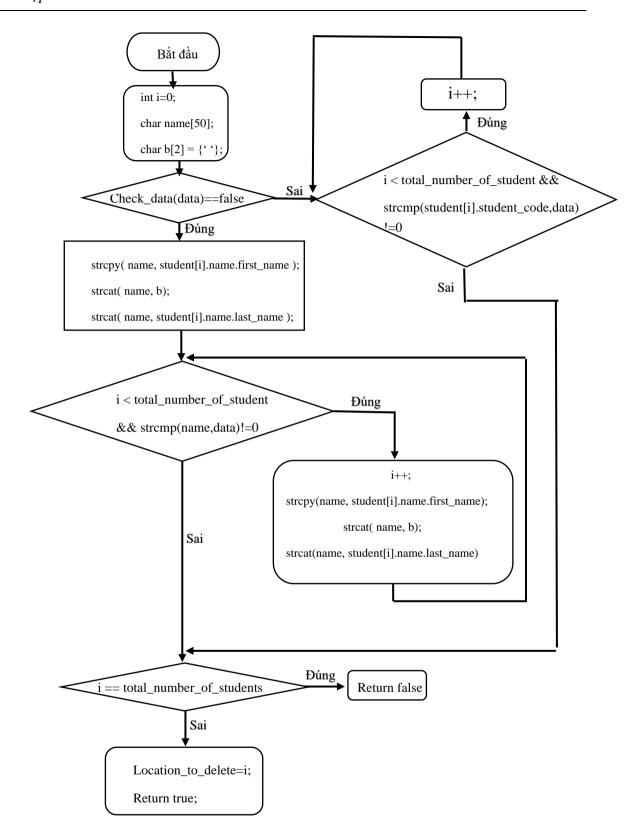


Hình 7: Quá trình tìm kiếm sinh viên theo thuật toán tìm kiếm nhị phân.

```
Step 1: Gán left = 0, right = total_number_of_students, khởi tạo mid
```

- Step 2: Kiểm tra dữ liệu là mã sinh viên hay là họ và tên.
- Step 3: Nếu là tên thì chuyển sang tìm kiếm tuần tự, ngược lại thì bắt đầu tìm kiếm theo mã sinh viên.

```
Step 4:
while ( left <= right ) {
    mid = ( left + right ) / 2;
    if ( stricmp(student[mid].student_code,data) == 0 ) {
        location_to_delete = mid;
        return true;
    }
    if ( stricmp(student[mid].student_code,data) > 0 )
        right = mid - 1;
        else if ( stricmp(student[mid].student_code,data) < 0 )
        left = mid + 1;
    }
    return false;
}</pre>
b, Tîm kiếm tuần tự (độ phức tạp: O(n))
```

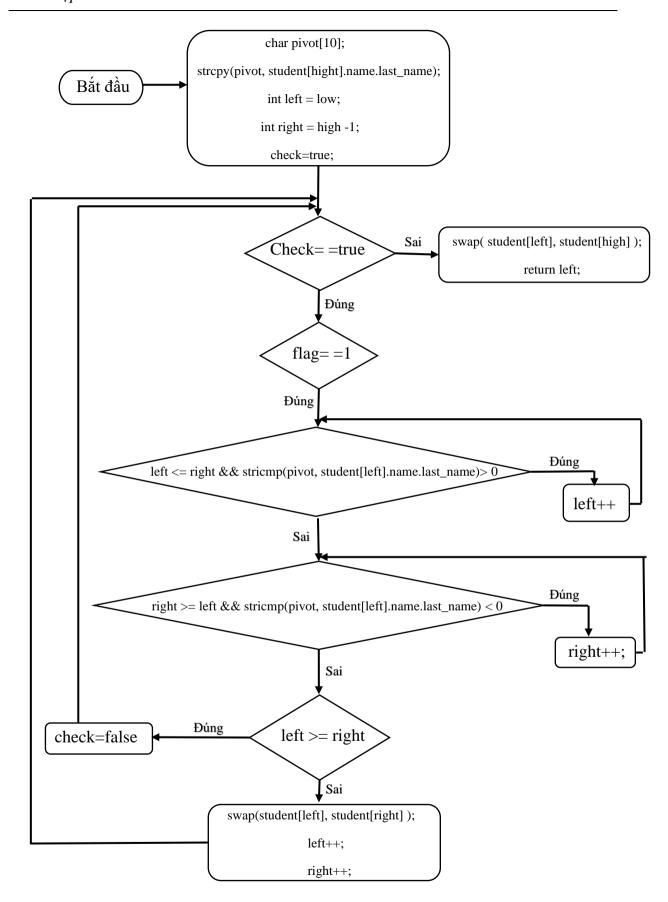


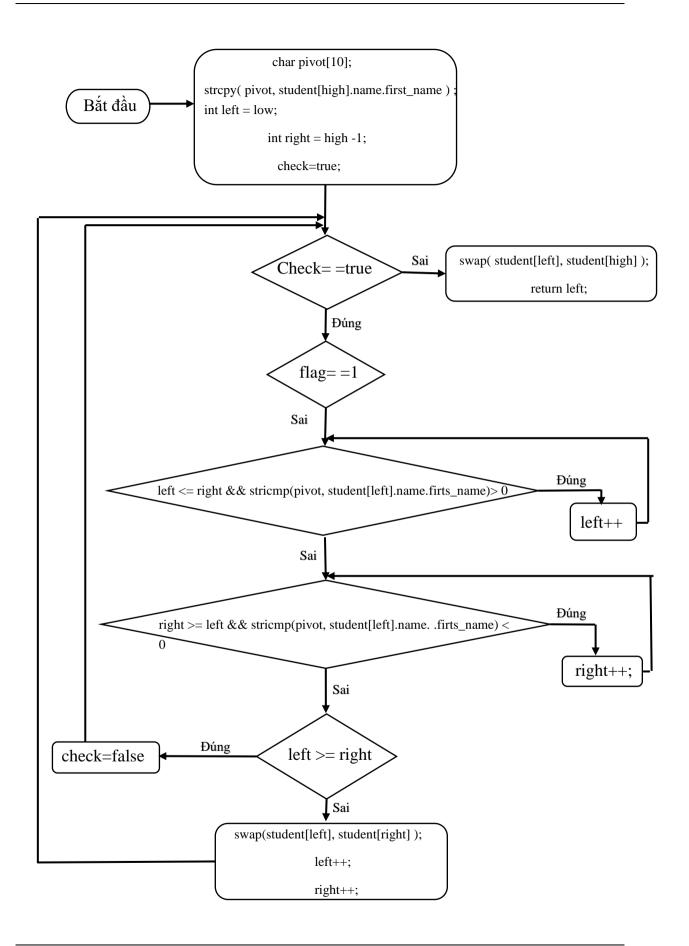
Hình 8: Quá trình tìm kiếm tuần tự.

```
Step 1:Khởi tạo các biến
      int i = 0;
      char name[50];
      char b[2] = {''};
Step 2: Kiểm tra dữ liệu là mã sinh viên hay là họ và tên.
Step 3: Nếu dữ liệu là họ và tên thì tìm kiếm theo Tên
      strcpy( name , student[i].name.last_name );
            strcat( name , b );
            strcat( name , student[i].name. first _name );
            while( i < total_number_of_students && stricmp(name,data) != 0 )
            { i++;
              strcpy( name , student[i].name.last_name );
              strcat( name , b );
              strcat( name , student[i].name. first _name );
            }
Step 4: Ngược lại nếu dữ liệu người dùng nhập là mã sinh viên.
       while( i < total_number_of_students &&
      strcmp(student[i].student_code,data) != 0)
      i++;
Step 5: Kiểm tra xem có sinh viên trong lớp không.
```

```
if(\ i == total\_number\_of\_students\ )\ return\ false; location\_to\_delete = i; return\ true;
```

- 3.3.2 Thuật toán sắp xếp Quick Sort (độ phức tạp O(nlog(n)))
- Thuật toán phân đoạn các sinh viên có tên (Họ lót) lớn hơn chốt sang trái, các sinh viên có tên (Họ lót) nhỏ hơn sang phải và trả về vị trị sau khi phân đoạn của phần tử chốt.





Hình 9: Lưu đồ thuật toán phân đoạn mảng.

```
+ Khởi tạo các biến:
         char pivot[10];
         if(flag==1)
         strcpy( pivot, student[high].name.last_name );
         else strcpy( pivot, student[high].name.first_name );
         int left = low;
         int right = high - 1;
+ Tiến hành phân đoạn danh sách sinh viên:
   while(true){
           if(flag==1){
           while( left <= right && stricmp(pivot,student[left].name.last_name) >0);
           left++;
           while(right >= left && stricmp(pivot,student[right].name.last_name) <0);
           right--;
            }
           else {
           while( left <= right && stricmp( pivot, student[left].name.first_name )> 0);
           left++;
           while(right >= left && stricmp( pivot, student[right].name.first_name ) <0);
           right--;
           }
           if (left >= right) break;
            swap( student[left] , student[right] );
```

```
left++;
     right--;
     swap( student[left] , student[high] );
  }
+ Trả về vị trí phần tử chốt
 return left;
   Đầu tiên sẽ tiến hành sắp xếp danh sách theo tên:
     Step 1: Tiến hành phân đoạn mảng.
     Step 2: Sau khi phân đoạn mảng ta sẽ có 2 mảng con: mảng bên trái của x
      và mảng bên phải của x. Tiếp tục công việc với mỗi mảng con (chọn
       pivot, phân đoạn) cho tới khi mảng được sắp xếp.
void quickSort_ first_name_step1( SV student[] , int low , int high ){
      if (low < high){
          int pi = partition ( student, low, high,0);
          quickSort_ first_name_step1( student , low , pi-1 );
          quickSort_ first_name_step1( student , pi+1 , high );
      }
}
   Tiếp theo sắp xếp các sinh viên có tên trùng nhau:
Step 1: Khởi tạo các biến
       int bien_trung_gian=0;
         char firstname[10];
```

```
Step 2: Tiến hành phân đoan các sinh viên có tên trùng nhau để tiến hành so
sánh. Sắp xếp theo ho lót các đoan sinh viên trùng nhau cho đến hết mảng
              strcpy(firstname,student[0].name. first_name);
             for(int i = 1; i < total_number_of_students; i++){
                  if(stricmp(firstname,student[i].name, first name) != 0 \parallel i ==
      total_number_of_students-1)
                   quickSort_last_name_step2(student,bien_trung_gian,i-1);
                   bien trung gian = i;
                   strcpy(firstname,student[i].name. first_name);
              }
3.3.3 Thuật toán xóa một sinh viên
 Step 1: Tìm kiếm sinh viên theo tên hoặc mã sinh viên.
 Step 2: Dịch chuyển mảng lùi 1 chỉ số – phần phía sau nơi xóa.
for (int j = location_to_delete+1; j < total_number_of_students; j++)
      student[j-1] = student[j];
      }
   Step 3: Giảm số lương sinh viên trong lớp đi 1
     total_number_of_students--;
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ
```

4.1. Tổ chức chương trình:

Gồm menu sau:

1-Thêm sinh viên.

- 2-Đọc file chứa thông tin sinh viên.
- 3-Sắp xếp danh sách.
- 4-Xóa sinh viên.
- 5-Tìm sinh viên.
- 6-Cấp mã sinh viên
- 7-Cấp email
- 8-In ra danh sách.
- 9-Thoát.

Sau khi nghiên cứu bài toán, nhóm chúng em chia chương trình thành các hàm nhỏ để dễ tổ chức, quản lí và bảo trì chương trình.

- void mainloop(SV student[]): Hàm làm vòng lặp cho chương trình, hiển thị menu.
- void add_students(SV student[]): Hàm dùng để thêm sinh viên.
- void Filter_the_data(char *p,char c): Hàm lọc dữ liệu từ file và dữ liệu nhận từ bàn phím.
- void Read_in_file(SV student[]): Hàm đọc thông tin từ file.
- void read_data_from_keyboard(char *p,int x,int y,int z,int w): Hàm đọc dữ liệu người dùng nhập từ bàn phím.
- int partition(SV student[] , int low , int high,int flag): Hàm Phân đoạn mảng sinh viên.
- void quickSort_last_name_step1(SV student[]): Hàm sắp xếp danh sách sinh viên theo họ lót bước 1.
- void quickSort_last_name_step2(SV student[] , int low , int high): Hàm sắp xếp danh sách sinh viên theo họ lót bước 2.
- void quickSort_ first_name_step1(SV student[] , int low , int high): Hàm sắp xếp danh sách sinh viên theo tên bước 1.
- void swap(SV &a , SV &b): Hàm hoán đổi vị trí hai sinh viên.
- void delete_student(SV student[]): Hàm xóa sinh viên.
- -void find_student(SV student[]): Hàm tìm kiếm sinh viên.

- bool Linear_Search(SV student[]): Hàm tìm kiếm sinh viên bằng phương pháp tìm kiếm tuần tự.
- bool Binary_Search(SV student[]): Hàm tìm kiếm sinh viên bằng phương pháp tìm kiếm nhị phân.
- void provide_student_code(SV student[]): Hàm cung cấp mã sinh viên.
- void provide_email(SV student[]): Hàm cung cấp email sinh viên.
- void print_list_step1(SV student[]),void print_list_step2(SV sv ,int x): Các hàm dùng để in danh sách ra màn hình.
- void output_file(SV student[]): Hàm xuất danh sách sinh viên ra file.
- bool check_data(char data[]): Hàm kiểm tra dữ liệu.

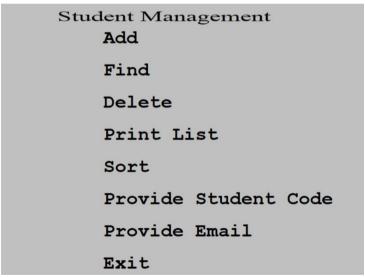
4.2. Ngôn ngữ cài đặt:

Ngôn ngữ lập trình C.

4.3. Kết quả

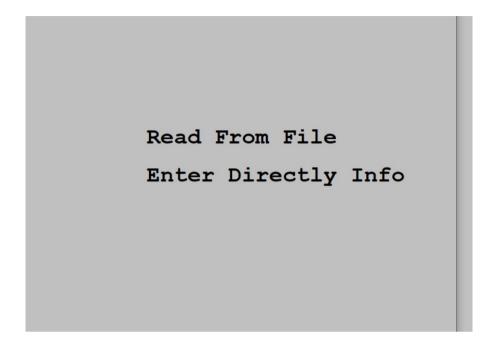
4.3.1. Giao diện chính của chương trình

- Menu chính của chương trình



Hình 10: Menu chính của chương trình.

- Menu lựa chọn cánh thêm sinh viên



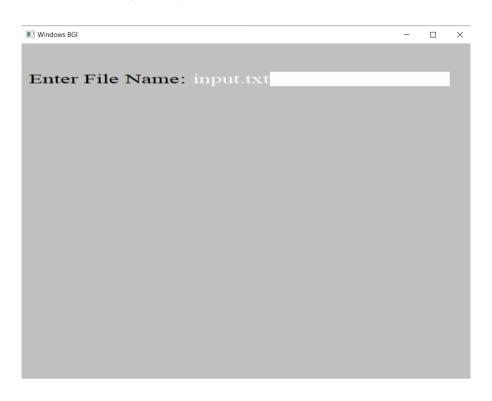
Hình 11: Menu lựa chọn cánh thêm sinh viên.

- Màn hình nhập thông tin sinh viên trực tiếp

■ Windows BGI	_	×
First Name: Truong		
Last Name: Pham Van Tien		
Gender: nam		
Date_of_birth: 18/11/2002		
Address: au co		
Back Add		

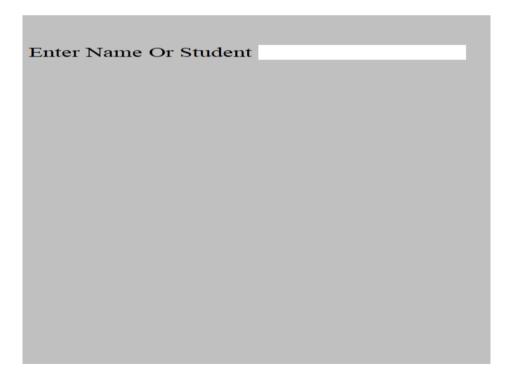
Hình 12: Màn hình nhập thông tin sinh viên trực tiếp.

- Màn hình nhập thông tin từ file



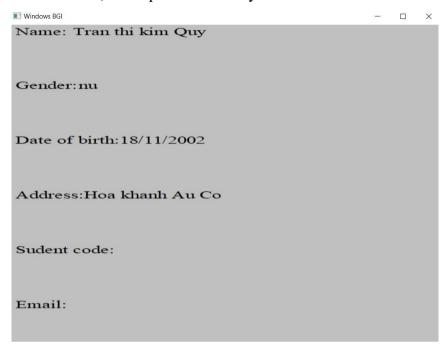
Hình 13: Màn hình nhập thông tin từ file.

Màn hình nhập thông tin để tìm kiếm hoặc xóa



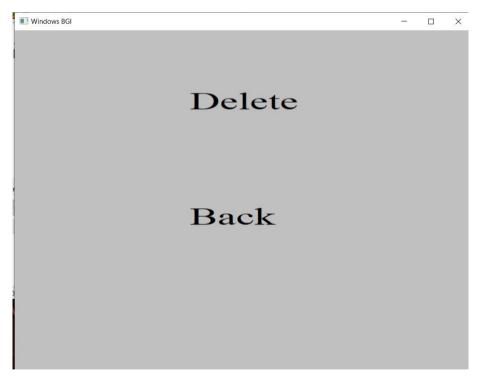
Hình 14: Màn hình nhập thông tin để tìm kiếm hoặc xóa.

- Màn hình hiện kết quả khi tìm thấy sinh viên



Hình 15: Màn hình hiện kết quả khi tìm thấy sinh viên.

- Màn hình lựa chọn xóa sinh viên



Hình 16: màn hình lựa chọn xóa sinh viên.

- Màn Hình hiển thị Danh sách sinh viên

Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email
0	dh djdj	djdj	djdjd	djdjd		
1	djdjdjdj djjdjd	djdjjddjdd	ddjdd	djdjjd		
2	Pham Van Tien Truong	Nam	18/11/2002	Hoa khanh Au Co		
3	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang		
4	Tran van Hai	nu	18/11/2002	Truong su pham da nang		
5	Nguyen hoang Yu	nu	18/11/2002	Hoa khanh Au Co		
6	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang		
7	Ho ky huy	Nam	18/11/2002	Hoa khanh Au Co		
8	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co		
9	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co		

Hình 17: Màn hình hiển thị danh sách sinh viên.

4.3.2. Kết quả thực thi của chương trình

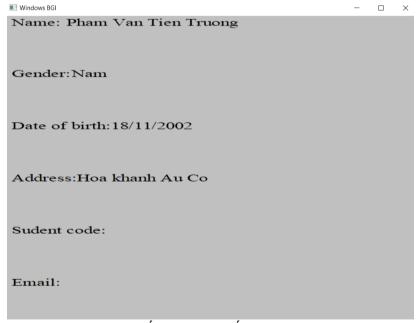
4.3.2.1 Dữ liệu đầu vào

Winde	ows BGI					- o
Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email
0	Pham Van Tien Truong	Nam	18/11/2002	Hoa khanh Au Co		
1	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang		
2	Tran van Hai	nu	18/11/2002	Truong su pham da nang		
3	Nguyen hoang Vu	nu	18/11/2002	Hoa khanh Au Co		
4	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang		
5	Ho ky huy	Nam	18/11/2002	Hoa khanh Au Co		
6	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co		
7	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co		
8	Tran thi kim Quy	nu	18/11/2002	Hoa khanh Au Co		
9	bui chi bao	nam	18/11/2002	ha tinh		
10	nguyen nhat hung	nam	18/11/2002	quang nam		

Hình 18: Dữ liệu đầu vào là danh sách sinh viên.

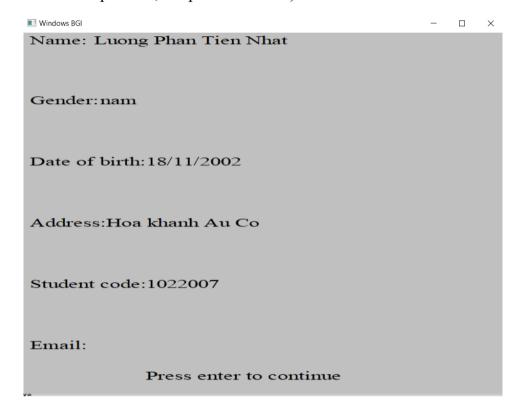
4.3.2.2 Kết quả tìm kiếm

- Tìm kiếm sinh viên có tên là :"Pham Van Tien Truong"



Hình 19: Kết quả tìm kiếm theo tên.

- Tìm kiếm sinh viên có mã số sinh viên là 1022007 (tìm kiếm sau khi danh sách lớp đã được cấp mã sinh viên)



Hình 20: Kết quả tìm kiếm theo mã sinh viên.

4.3.2.3 Kết quả sau khi xóa một sinh viên

- Xóa sinh viên có tên là: "Pham Van Tien Truong".

■ Wind	ows BGI					-
Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email
0	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang		
1	Tran van Hai	nu	18/11/2002	Truong su pham da nang		
2	Nguyen hoang Vu	nu	18/11/2002	Hoa khanh Au Co		
3	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang		
4	Ho ky huy	Nam	18/11/2002	Hoa khanh Au Co		
5	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co		
6	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co		
7	Tran thi kim Quy	nu	18/11/2002	Hoa khanh Au Co		
8	bui chi bao	nam	18/11/2002	ha tinh		
9	nguyen nhat hung	nam	18/11/2002	quang nam		

Hình 21: Kết quả xóa sinh viên theo tên.

- Xóa sinh viên có mã sinh viên là : 1022003 (Xóa sau khi danh sách lớp đã được cấp mã sinh viên). Đây là sinh viên có tên "Ho Ky Huy"

■ Wind	- D								
Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email			
0	Bui Chi Bao	Nam	18/11/2002	Ha Tinh	10022000	10022000@gmail.com			
1	Tran van Hai	nu	18/11/2002	Truong su pham da nang	10022001	10022001@gmail.com			
2	Nguyen Nhat Hung	Nam	18/11/2002	Quang Nam	10022002	10022002@gmail.com			
3	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang	10022004	10022004@gmail.com			
4	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang	10022005	10022005@gmail.com			
5	Tran thi kim Quy	nu	18/11/2002	Hoa khanh Au Co	10022006	10022006@gmail.com			
6	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co	10022007	10022007@gmail.com			
7	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co	10022008	10022008@gmail.com			
8	Nguyen hoang Vu	nu	18/11/2002	Hoa khanh Au Co	10022009	10022009@gmail.com			

Hình 22: Kết quả xóa sinh viên theo mã sinh viên.

4.3.2.4 Kết quả sau khi sắp xếp sinh viên

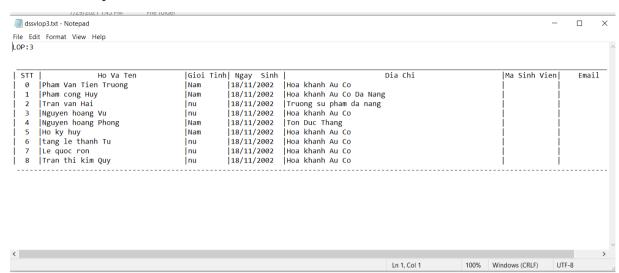
■ Windo	ows BGI					- 0
Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email
0	bui chi bao	nam	18/11/2002	ha tinh		
1	Tran van Hai	nu	18/11/2002	Truong su pham da nang		
2	nguyen nhat hung	nam	18/11/2002	quang nam		
3	Ho ky huy	Nam	18/11/2002	Hoa khanh Au Co		
4	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang		
5	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang		
6	Tran thi kim Quy	nu	18/11/2002	Hoa khanh Au Co		
7	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co		
8	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co		
9	Nguyen hoang Yu	nu	18/11/2002	Hoa khanh Au Co		

Hình 23: Kết quả sau khi sắp xếp.

4.3.2.5 Kết quả sau khi cung cấp mã sinh viên và email

■ Wind	dows BGI					- t
Stt	Ho Va Ten	Gioi Tinh	Ngay sinh	Dia Chi	Ma Sinh Vien	Email
0	bui chi bao	nam	18/11/2002	ha tinh	10022000	10022000@gmail.com
1	Tran van Hai	nu	18/11/2002	Truong su pham da nang	10022001	10022001@gmail.com
2	nguyen nhat hung	nam	18/11/2002	quang nam	10022002	10022002@gmail.com
3	Ho ky huy	Nam	18/11/2002	Hoa khanh Au Co	10022003	10022003@gmail.com
4	Pham cong Huy	Nam	18/11/2002	Hoa khanh Au Co Da Nang	10022004	10022004@gmail.com
5	Nguyen hoang Phong	Nam	18/11/2002	Ton Duc Thang	10022005	10022005@gmail.com
6	Tran thi kim Quy	nu	18/11/2002	Hoa khanh Au Co	10022006	10022006@gmail.com
7	Le quoc ron	nu	18/11/2002	Hoa khanh Au Co	10022007	10022007@gmail.com
8	tang le thanh Tu	nu	18/11/2002	Hoa khanh Au Co	10022008	10022008@gmail.com
9	Nguyen hoang Yu	nu	18/11/2002	Hoa khanh Au Co	10022009	10022009@gmail.com

Hình 24: Kết quả sau khi cấp mã sinh viên và email.



4.3.2.6 Kết quả khi in ra file

Hình 25: Kết quả khi in ra file.

4.3.2.7 Các lỗi người dùng gặp phải

- Lớp rỗng: "Empty class".
- Danh sách lớp đã được sắp xếp: "Arranged".
- Mã sinh viên đã được cấp: "Student Code Has Been Issued".
- Danh sách lớp chưa sắp xếp nên không thể cấp mã sinh viên: "Not Arranged".
 - Email đã được cấp: "Email Has Been Issued".
 - Chưa cấp mã sinh viên nên không thể cấp Email: "No Student Code Yet".

4.3.3. Nhận xét đánh giá:

Đáp ứng đầy đủ các yêu cầu về chức năng cần thiết.

Giao diện đơn giản dễ sử dụng nhưng vẫn cần được cải tiến thêm.

Thuật toán đơn giản nên thời gian chạy cho kết quả khá nhanh và ổn định.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN

5.1. Kết luận:

Thuật toán đáp ứng được nhu cầu của đề bài. Qua đồ án lần này, ta có thể hiểu được và sử dụng các mảng song song để lưu dữ liệu quan hệ, các mảng một chiều, các thuật toán thao tác với mảng, các hàm xử lý chuỗi, đồng thời có thể áp dụng nó vào bài toán thực tế. Hơn thế nữa, đồ án còn giúp sinh viên rèn luyện khả năng tư duy, sáng tạo, làm việc nhóm và khả năng thuyết trình.

5.2. Hướng phát triển:

- Tiếp thu ý kiến đóng góp và phản hồi của mọi người để phát triển thêm về chức năng và giao diện.
- Nghiên cứu, phát triển thuật toán tối ưu hơn nữa.

TÀI LIỆU THAM KHẢO

- [1]. Chương trình quản lý sinh viên sử dụng struct trong
- C,https://nguyenvanhieu.vn/chuong-trinh-quan-ly-sinh-vien-su-dung-struct/
- [2]. Các hàm trong chế độ đồ họa (phần 1)

https://tuhoclaptrinh.cachhoc.net/2017/02/26/cac-ham-trong-che-do-do-hoa-phan-1/

- [3]. 5 Thuật Toán Tìm Kiếm Mọi LTV Nên Biết https://codelearn.io/sharing/5-thuat-toan-tim-kiem-moi-ltv-nen-biet
- [4]. Đồ Án Game Rắn Săn Mồi, https://github.com/nguyenvanhieuvn/Snake

PHŲ LŲC

1. Nhập Thông Tin Sinh Viên Từ Bàn Phím

```
void add_students( SV student[]){
                                settextstyle(1,0,3);
                                setfillstyle(1,15);
                         showText(50,50,"First Name:");
                        showText(50,150,"Last Name:");
                          showText(50,250,"Gender:");
                       showText(50,350,"Date_of_birth:");
                          showText(50,450,"Address:");
                               bar(250,50,600,75);
                              bar(250,150,600,175);
                              bar(250,250,600,275);
                              bar(300,350,650,375);
                              bar(200,450,600,475);
read_data_from_keyboard(student[total_number_of_students].name.first_name,
250,50,600,75);
Filter_the_data(student[total_number_of_students].name.first_name,63);
read_data_from_keyboard(student[total_number_of_students].name.last_name,2
50,150,600,175);
Filter_the_data(student[total_number_of_students].name.last_name,63);
```

```
read_data_from_keyboard(student[total_number_of_students].gender,250
,250,600,275);
Filter_the_data(student[total_number_of_students].gender,63);
read_data_from_keyboard(student[total_number_of_students].Date_of_bi
rth,300,350,650,375);
Filter_the_data(student[total_number_of_students].Date_of_birth,63);
read_data_from_keyboard(student[total_number_of_students].address,20
0,450,600,475);
Filter_the_data(student[total_number_of_students].address,63);
total_number_of_students++;
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN) || mousex() < 100 ||
mousey() < 400 \parallel mousex() > 400){
showText(150,550,"Back");
 showText(300,550,"Add");
  }
  if(mousex() \le 275)
             showTextBackground(150,550,"Back",15);
 }else {
             showTextBackground(300,550,"Add",15);
             cleardevice();
             add_students(student); }
```

```
delay(500);
cleardevice();}
```

2. Đọc Dữ Liệu Từ Bàn Phím

```
void read_data_from_keyboard(char *p,int x,int y,int z,int w){
     char str[2];
      str[1] = 0;
      char key=0;
      int m=0;
      int r;
      bool check=false;
      bool check_full=false;
      while(key!=13){
                      key= getch();
                      if(key==8&&check_full==false){
                                p[m]='?';
                                m--;
                          char b[51]={""};
                           strncat(b,p,m+1);
                           bar(x,y,z,w);
```

```
outtextxy(x,y,b);
   if(p[0]=='?') check_full=true;
}
else str[0] = key;
if(key!=13&&key!=8){
     for(int i=0;i<strlen(p);i++){
            if(p[i]=='?') {
                  p[i]=key;
                  check=true;
                  r=i;
                  if(i==strlen(p)-1) m=strlen(p)-1;
                            check_full=false;
                            break;
                  }
              }
            if(check==true){
                    char b[51]={""};
                    strncat(b,p,r+1);
                    bar(x,y,z,w);
                     outtextxy(x,y,b);
                     check=false;
```

```
}
                                        else {
                                              str[0]=key;
                                              strcat(p,str);
                                              bar(x,y,z,w);
                                              outtextxy(x,y,p);
                                              m=strlen(p)-1;
                         }
        }
}
3. Hàm Lọc Dữ Liệu Đọc từ file Và bàn phím
void Filter_the_data(char *p,char c){
      for(int i=0;i<strlen(p);i++){
         if(p[i]==c) p[i]=' ';
}
      char string[70];
      int key;
      for(int i=strlen(p)-1;i>=0;i--){
              if(p[i]!=' ') {
```

```
key=i;
               break;}
      }
      if(key==strlen(p)-1) return;
      strncpy(string,p,key+1);
      strcpy(p,string);
}
4. Hàm Đọc Dữ Liệu Từ file
void Read_in_file(SV student[]){
    p=fopen("input.txt","r");
    cleardevice();
   if(p==NULL) {
          PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
            showTextBackground(250,200,"File Does Not Exist",15);
                                        }
  else {
     char first_name[50];
```

```
char last_name[50];
      char gender[50];
      char d[50];
      char ns[50];
    int n=total_number_of_students;
    while(fscanf(p, "Last_Name: %s First_Name: %s Gender: %s
Date_of_birth: %s Address: %s\n", student[n].name.last_name,student[n].name.
first _name,student[n].gender,student[n].Date_of_birth,student[n].address) !=
EOF){
         Filter_the_data(student[n].name.first_name,95);
         Filter_the_data(student[n].name.last_name,95);
         Filter_the_data(student[n].address,95);
       total_number_of_students++;
                 n=total_number_of_students;
     }
    showTextBackground(250,200,"Read Successfully",15);
    fclose(p);
  }
  delay(500);
}
```

5. Các Hàm dùng để in danh sách sinh viên

```
void print_list_step1(SV student[]){
               setbkcolor(0);
               cleardevice();
               setcolor(15);
               rectangle(5,5,1470,30+25*(total_number_of_students));
               rectangle(5,5,1470,30);
               line(50,5,50,30+25*(total_number_of_students));
               line(400,5,400,30+25*(total_number_of_students));
               line(500,5,500,30+25*(total_number_of_students));
               line(700,5,700,30+25*(total_number_of_students));
               line(1100,5,1100,30+25*(total_number_of_students));
               line(1250,5,1250,30+25*(total_number_of_students));
               settextstyle(2,0,7);
               outtextxy(7,7,"Stt");
              outtextxy(140,7,"Ho Va Ten");
              outtextxy(415,7,"Gioi Tinh");
              outtextxy(550,7,"Ngay sinh");
              outtextxy(840,7,"Dia Chi");
               outtextxy(1110,7,"Ma Sinh Vien");
               outtextxy(1280,7,"Email");
```

```
outtextxy(550,30+25*(total_number_of_students)+40,"Press
enter to continue");
               int x=5;
               char s[5]={""};
           for(int i = 0; i < total\_number\_of\_students; i++){
              print_list_step2( student[i],x);
              sprintf (s,"%d",i);
              outtextxy(7,x+25+2,s);
              line(5,x+25,1470,x+25);
              x + = 25;
           }
          clearmouseclick(WM_LBUTTONDOWN);
          char key=0;
         while(key!=13){
          key=getch();
                                            }
          char key=0;
          while(key!=13){
                     key=getch();
              }
                 setbkcolor(7);
```

int left = 0;

int right = total_number_of_students - 1;

```
setbkcolor(0);
                  setbkcolor(7);
}
void print_list_step2( SV sv ,int x){
         char name[50]={""};
          char b[2] = {''};
          strcpy( name , sv.name.first_name );
     strcat( name, b);
     strcat( name , sv.name.last_name );
     outtextxy(70,x+25+2,name);
     outtextxy(415,x+25+2,sv.gender);
     outtextxy(550,x+25+2,sv.Date_of_birth);
     outtextxy(710,x+25+2,sv.address);
     outtextxy(1110,x+25+2,sv.student_code);
     outtextxy(1260,x+25+2,sv.email);
}
6. Hàm tìm kiếm sinh viên bằng thuật toán nhị phân
bool Binary_Search( SV student[] , char data[] ){
```

```
int mid;
       if( check_data(data) ){
            while ( left <= right ){
                 mid = (left + right) / 2;
                 if (stricmp(student[mid].student_code,data) == 0){
                        location_to_delete = mid;
                        return true;
                  }
                  if (stricmp(student[mid].student_code,data) > 0)
                         right = mid - 1;
                  else if (stricmp(student[mid].student_code,data) < 0)
                         left = mid + 1;
             }
             return false;
      }
      else {
           return Linear_Search(student,data);
      }
}
7. Hàm tìm kiếm sinh viên bằng thuật toán tuần tự
bool Linear_Search( SV student[] , char data[] ){
       int i = 0;
       char name[50];
       char b[2] = {''};
       if( check_data(data) == false ){
         strcpy( name , student[i].name.last_name );
         strcat( name, b);
```

```
strcat( name , student[i].name. first _name );
         while( i < total_number_of_students && stricmp(name,data) != 0 ) {
           i++;
           strcpy( name , student[i].name.last_name );
           strcat( name, b);
           strcat( name , student[i].name. first_name );
         }
      }
      else {
         while(i < total_number_of_students && strcmp(student[i].student_code,data)
    !=0) i++;
      }
      if( i == total_number_of_students ) return false;
      location_to_delete = i;
      return true;
}
8. Hàm Xóa Một Sinh Viên
void delete_student( SV student[] ){
      find_student( student );
       if( check_find ){
              cleardevice();
             settextstyle(1,0,5);
             clearmouseclick(WM_LBUTTONDOWN);
             while(!ismouseclick(WM_LBUTTONDOWN) || mousex() < 200 ||
    mousey() > 600 \parallel mousey() < 110 \parallel mousex() > 580){
```

```
showText(300,100,"Delete");
              showText(300,300,"Back");
             if (mousey() \le 250){
                  showTextBackground(300,100,"Delete",15);
                  for(int j = location_to_delete+1; j < total_number_of_students; j+)
                  student[j-1] = student[j];
                  total_number_of_students--;
              }
              else {
                showTextBackground(300,300,"Back",15);
                cleardevice();
      }
}
9. Hàm Kiệm Tra dữ liệu
    bool check_data( char data[] ){
      if( data[0] >= '0' && data[0] <= '9') return true;
      return false;
}
10. Hàm Phân Đoạn mảng sinh viên
int partition( SV student[] , int low , int high ,int flag){
      char pivot[10];
      if(flag==1)
```

```
strcpy( pivot, student[high].name.last_name );
     else
            strcpy( pivot, student[high].name. first_name );
    int left = low;
    int right = high - 1;
    while(true){
         if(flag==1){
             while( left <= right && stricmp(pivot,student[left].name.last_name)>0)
             left++;
            while(right >= left && stricmp(pivot,student[right].name.last_name)<0)
             right--;
          }
         else {
            while( left<= right && stricmp( pivot, student[left].name.first_name)>0)
            left++;
            while(right>=left && stricmp( pivot, student[right].name.first_name)<0)</pre>
             right--;
         }
        if (left >= right) break;
                swap( student[left] , student[right] );
                left++;
                right--;
        }
      swap( student[left] , student[high] );
      return left;
}
```

11. Hàm Sắp Xếp Danh Sách sinh viên Theo tên

}

```
void quickSort_ first_name_step1( SV student[] , int low , int high )
{
    if (low < high)
    {
        int pi = partition( student, low, high,0);
        quickSort_ first_name_step1( student , low , pi-1 );
        quickSort_ first_name_step1( student , pi+1 , high );
    }
}</pre>
```

12. Hàm xác Định mảng Các Sinh viên có tên trùng nhau

```
void quickSort_last_name_step1(SV student[]){
    int bien_trung_gian=0;
    char firstname[10];
    strcpy(firstname,student[0].name. first_name);
    for(int i = 1; i < total_number_of_students; i++){
        if( stricmp(firstname,student[i].name.first_name) != 0 || i == total_number_of_students-1)
        {
            quickSort_last_name_step2(student,bien_trung_gian,i-1);
            bien_trung_gian = i;
            strcpy(firstname,student[i].name. first_name);
    }
}</pre>
```

}

13. Hàm sắp xếp sinh viên Theo họ lót

```
void quickSort_last_name_step2(SV student[],int low,int high){
    if(low<high){
        int pi=partition(student,low,high,1);
        quickSort_last_name_step2(student,low,pi-1);
        quickSort_last_name_step2(student,pi+1,high);
    }
}</pre>
```

14. Hàm Cấp Mã Sinh Viên

```
void provide_student_code(SV student[])
  int key=1022000;
  for(int i=0;i<total_number_of_students;i++){
      itoa(key,student[i].student_code,10);
      key++;
    }
}</pre>
```

15. Hàm Cấp Email

```
void provide_email(SV student[]){
```

```
char b[11]="@gmail.com";
for(int i=0;i<total_number_of_students;i++){
    strcpy(student[i].email,student[i].student_code);
    strcat(student[i].email, b);
}</pre>
```

16. Hàm in danh sách sinh viên ra file

```
void output_file(SV student[])
      FILE *fp;
      char filename[13]="dssvlop .txt";
      FILE *p;
      p=fopen("lop.txt","r");
      int n;
      char r='0';
      fscanf(p,"%d",&n);
      filename[7]=r+n+1;
      fclose(p);
      p=fopen("lop.txt","w");
      fprintf(p,"%d",n+1);
      fclose(p);
      fp=fopen(filename,"w");
      fprintf(fp,"%-4s%d\n\n","LOP:",n+1);
      fprintf(fp,"%-145s\n","
```

```
_");
     fprintf(fp,"%-145s\n","| STT |
                                       Ho Va Ten
                                                       |Gioi Tinh| Ngay Sinh |
   Dia Chi
                     |Ma Sinh Vien|
                                    Email
                                             |");
     for(int i=0;i<total_number_of_students;i++){</pre>
          fprintf(fp,"%c%2s%-3d%c","," ",i,");
          char name[34];
              char b[2] = {''};
               strcpy( name , student[i].name.first name );
               strcat( name, b);
               strcat( name ,student[i].name.last_name );
          fprintf(fp,"%-33s%c",name,");
          fprintf(fp,"%-9s%c",student[i].gender,'|');
          fprintf(fp,"%-12s%c",student[i].Date_of_birth,";
          fprintf(fp,"%-50s%c",student[i].address,";
          fprintf(fp,"%-12s%c",student[i].student_code,"|');
          fprintf(fp,"%-15s%c\n",student[i].email,"|');
        }
     fprintf(fp,"%-145s"," ------
        -----");
     fclose(fp);
}
17. Hàm Hiển Thị menu
void mainloop(SV student[]){
      initwindow (800,600);
      setbkcolor(7);
     int x = 0;
```

```
char *s = new char[5];
  PlaySound (TEXT ("start.wav"), NULL, SND_FILENAME | SND_ASYNC);
  while (x <= 100)
         cleardevice();
         setcolor (10);settextstyle(4,0,5);outtextxy (60,200," Loading...");
         sprintf (s,"%d",x);
         setcolor (4);
         outtextxy (550,200,strcat(s,"%"));
         if (x == 100)
                delay (500);
         delay (20);
         x++;
  }
  cleardevice();
  while(true){
         setbkcolor(0);
    setbkcolor(7);
         clearmouseclick(WM_LBUTTONDOWN);
         while (!ismouseclick(WM_LBUTTONDOWN) || mousex() < 200 ||
mousey() > 600 \parallel mousey() < 20 \parallel mousex() > 580 )
         settextstyle(1,0,4);
         showText(120,10,"Student Management");
         settextstyle(10,0,5);
         showText(220,50,"Add");
         showText(220,120,"Find");
         showText(220,190,"Delete");
```

```
showText(220,260,"Print List");
        showText(220,330,"Sort");
        showText(220,400,"Provide Student Code");
        showText(220,470,"Provide Email");
        showText(220,540,"Exit");
  }
        clearmouseclick(WM_LBUTTONDOWN);
    if (mousey() \le 95){
           PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
        showTextBackground(220,50,"Add",15);
           cleardevice();
           clearmouseclick(WM_LBUTTONDOWN);
           while (!ismouseclick(WM_LBUTTONDOWN) || mousex() < 200 ||
mousey() > 600 \parallel mousey() < 110 \parallel mousex() > 580 )
               showText(220,200,"Read From File");
             showText(220,270,"Enter Directly Info");
           if (mousey() \le 270){
             PlaySound (TEXT ("beep.wav"), NULL, SND FILENAME |
SND_ASYNC);
               showTextBackground(220,200,"Read From File",15);
               Read_in_file(student);
               }else {
                     PlaySound (TEXT ("beep.wav"), NULL,
SND_FILENAME | SND_ASYNC);
         showTextBackground(220,270,"Enter Directly Info",15);
```

```
cleardevice();
         add_students(student);
       check_innit = true;
       check_sort = false;
       check_provide_student_code = false;
       check_provide_email = false;
       cleardevice();
       clearmouseclick(WM_LBUTTONDOWN);
    } else if(mousey() <= 165){
         showTextBackground(220,120,"Find",15);
         PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
         setbkcolor(7);
           cleardevice();
         if(check_innit==true){
                 find_student( student );
          }
          else {
               delay(300);
                     PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
             showTextBackground(250,200,"Empty class",15);
          }
           delay(1000);
           cleardevice();
           clearmouseclick(WM_LBUTTONDOWN);
```

```
ext{less if}(mousey() \le 235){
               showTextBackground(220,190,"Delete",15);
               cleardevice();
               PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
               if(check_innit==true){
           delete_student(student);
           }
           else {
               delay(300);
               PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
             showTextBackground(250,200,"Empty class",15);
               }
               delay(1000);
           cleardevice();
           clearmouseclick(WM_LBUTTONDOWN);
         else if(mousey() \le 305)
               showTextBackground(220,260,"Print List",15);
               cleardevice();
               if(check_innit==true){
                           PlaySound (TEXT ("beep.wav"), NULL,
SND_FILENAME | SND_ASYNC);
                       print_list_step1(student);
           }else {
```

```
PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
                 showTextBackground(250,200,"Empty class",15);
               }
          cleardevice();
          clearmouseclick(WM_LBUTTONDOWN);
         else if(mousey() \le 375)
                 cleardevice();
                 showTextBackground(220,330,"Sort",15);
                 cleardevice();
                     PlaySound (TEXT ("beep.wav"), NULL,
SND_FILENAME | SND_ASYNC);
          if(check_innit==false){
            PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
            showTextBackground(250,200,"Empty class",15);
               }
               else if(check_sort==false){
  quickSort_last_name_step1(student,0,total_number_of_students-1);
                     quickSort_first_name_step1(student);
                     showTextBackground(250,200,"Sort Success",15);
                 check_sort=true;
            }
          else if(check_sort==true){
```

```
PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
              showTextBackground(250,200,"Arranged",15);
              }
            delay(1000);
            cleardevice();
            clearmouseclick(WM_LBUTTONDOWN);
    }
    else if(mousey() \leq 445){
        PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
        showTextBackground(220,400,"Provide Student Code",15);
        cleardevice();
        if(check innit==false){
                    PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
             showTextBackground(250,200,"Empty class",15);
           }
        else if(check_provide_student_code == true){
              PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
              showTextBackground(100,200,"Student Code Has Been
Issued",15);
              }
              else if(check sort == false){
              PlaySound (TEXT ("error.wav"), NULL, SND_FILENAME |
SND_ASYNC);
                    showTextBackground(100,200,"Not Arranged",15);
```

```
}else {
                     showTextBackground(50,100,"Student Code has been
issued successfully",15);
                     provide_student_code( student );
                     check_provide_student_code = true;
               }
               delay(1000);
               cleardevice();
          clearmouseclick(WM_LBUTTONDOWN);
        }
        else if(mousey() \leq 515){
               PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
               cleardevice();
               showTextBackground(220,470,"Provide Email",15);
               cleardevice();
              if(check_innit==false){
                     PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
                     PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
             showTextBackground(250,200,"Empty class",15);
               }else if(check_provide_email==true){
                     PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
                     showTextBackground(100,200,"Email Has Been
Issued",15);
```

```
}
               else if(check_provide_student_code == false){
                     PlaySound (TEXT ("error.wav"), NULL,
SND_FILENAME | SND_ASYNC);
                     showTextBackground(100,200,"No Student Code
Yet",15);
               }
               else {
                     showTextBackground(100,200,"Email Issued
Successfully",15);
                     provide_email( student );
                     check_provide_email = true;
               }
               delay(1000);
               cleardevice();
           clearmouseclick(WM_LBUTTONDOWN);
         }
        else {
               PlaySound (TEXT ("beep.wav"), NULL, SND_FILENAME |
SND_ASYNC);
               showTextBackground(220,540,"Exit",15);
           setbkcolor(7);
           cleardevice();
           clearmouseclick(WM_LBUTTONDOWN);
           output_file( student );
           exit(0);
```

```
}
}
/* Do An Quan Li Hoc Sinh */
```