TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO

MÔN: KỸ THUẬT LẬP TRÌNH

ĐỀ TÀI: BÀI TẬP LỚN

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN VŨ TẤN PHÚC

NGUYỄN CÔNG MINH QUÂN

TRẦN HUỲNH HOÀ PHÚC

Lớp : CQ.65.CNTT

Khoá : 65

Tp. Hồ Chí Minh, tháng 5 năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO

MÔN: KỸ THUẬT LẬP TRÌNH

ĐỀ TÀI: BÀI TẬP LỚN

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN VŨ TẤN PHÚC

NGUYỄN CÔNG MINH QUÂN

TRẦN HUỲNH HOÀ PHÚC

Lớp : CQ.65.CNTT

Khoá : 65

Tp. Hồ Chí Minh, tháng 5 năm 2025

LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi tới Quý Thầy Cô Bộ môn Công nghệ Thông tin Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh lời chúc sức khỏe và lòng biết ơn sâu sắc.

Em xin chân thành cảm ơn quý thầy cô đã giúp đỡ tạo điều kiện để em hoàn thành báo cáo về phần bài tập lớn. Đặc biệt em xin cảm ơn thầy Trần Phong Nhã đã nhiệt tình giúp đỡ, hướng dẫn cho em kiến thức, định hướng và kỹ năng để có thể hoàn thành bài báo cáo này.

Tuy đã cố gắng trong quá trình nghiên cứu tìm hiểu tuy nhiên do kiến thức còn hạn chế nên vẫn còn tồn tại quá nhiều thiếu xót. Vì vậy em rất mong nhận được sự đóng góp ý kiến của Quý thầy cô bộ môn để đề tài của em có thể hoàn thiện hơn.

Lời sau cùng, em xin gửi lời chúc tới Quý Thầy Cô Bộ môn Công nghệ thông tin và hơn hết là thầy Trần Phong Nhã có thật nhiều sức khỏe, có nhiều thành công trong công việc. Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chính Minh, ngày …… tháng …… năm …….

Giảng viên hướng dẫn

ThS. Trần Phong Nhã

MỤC LỤC

DANH MỤC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú

DANH MỤC HÌNH ẢNH

A. LÝ THUYẾT

1. Hàm

*Khái niệm về hàm:

- Một hàm trong C được hiểu theo nghĩa là một "Routine" hoặc "subprogram".
- Hàm là một đơn vị độc lập trong C
- Không được xây dựng hàm bên trong 1 hàm khác
- Mỗi hàm có thể có các biến, hằng, mảng riêng
- Một chương trình viết bằng C gồm 1 hoặc nhiều hàm, Trong đó có 1 hàm chính là hàm "main()"
- Hàm có thể có giá trị trả về (kết quả của hàm) hoặc không có giá trị trả về (chỉ đơn thuần thực hiện 1 công việc nào đó)
- Hàm có thể có hoặc không có tham số

* Khai báo hàm:

- Nguyên mẫu hàm (prototype của hàm)
  - Prototype hàm chỉ rõ các đặc điểm chính
  - Tên của hàm
  - Số lượng và kiểu của từng tham số hàm sẽ nhận + Giá trị trả về sau khi hàm kết húc.
  - Phải khai báo prototype của hàm trước khi sử dụng hàm ->thường khai báo nguyên mẫu ở đầu chương trình.

-Prototype hàm không cho thấy hàm sẽ làm những gì

*Công thức khai báo:

Kiểu_hàm Tên_hàm (Kiểu_tham_số_1, Kiểu_tham_số_2, ...);

Ví dụ: #include <stdio.h>

```
Int tong(int a, int b) {
   Return a + b;}
Int main() {
   Int x = 5, y = 7;
   Printf("Tong = %d\n", tong(x, y));
   Return 0;}
```

2. Con trỏ

- Khái niệm về con trỏ: Một con trỏ - pointer là một biến mà trong đó giá trị của nó là địa chỉ của biến khác. Ví dụ như địa chỉ của vùng nhớ. Giống như các biến và hằng số, bạn phải khai báo con trỏ trước khi bạn có thể sử dụng nó để lưu trữ bất kì địa chỉ của biến nào. Dạng tổng quát của việc khai báo con trỏ như sau: Kieu_du_lieu*ten_bien
- Ở đây, kieu_du_lieu là kiểu dữ liệu cơ bản con trỏ, nó là kiểu hợp lệ trong ngôn ngữ C và var-ten_bien là tên giá trị của con trỏ. Phần ký tự * sử dụng trong khai báo con trỏ giống như việc bạn sử dụng cho phép nhân. Mặc dù vậy, trong khai báo này, ký tự * được thiết kế để sử dụng các biến của con trỏ. Dưới đây là một số cách khai báo hợp lệ của con trỏ:

  Int *contro; /* con tro tro toi mot so nguyen */

  Double *phithuebao; /* con tro tro toi mot so double */

  Float *hocphi; /* con tro tro toi mot so float */

  Char *ho, *ten; /* con tro tro toi mot ky tu */

- Kiểu dữ liệu thực sự của giá trị của tất cả các con trỏ, có thể là số nguyên, float, ký tự, hoặc kiểu khác như một số thập lục phân dài – Long hexa biểu diễn một địa chỉ bộ nhớ. Điểm khác nhau duy nhất của các con trỏ của các kiểu dữ liệu khác nhau là kiểu dữ liệu của biến hoặc hằng số mà con trỏ chỉ tới.

Ví dụ:
```
#include <stdio.h>
    int main() {
        int a = 10;
        int *p = &a;
        printf("Gia tri cua a: %d\n", *p);
        return 0;}
```

3. Con trỏ mảng
   - Con trỏ mảng (Array Pointer) là con trỏ trỏ đến một mảng, tức là địa chỉ của cả mảng, chứ không chỉ là phần tử đầu tiên.

   Ví dụ:
```
#include <stdio.h>
int main() {
    int arr[] = {1, 2, 3};
    int *p = arr;
    for (int i = 0; i < 3; i++) {
        printf("%d ", *(p + i));}
    return 0;}
```

4. Mảng con trỏ

- Mảng con trỏ là sự mở rộng khái niệm con trỏ. Mảng con trỏ là một mảng mà mỗi phần tử của nó có thể chứa được một địa chỉ nào đó. Cũng giống như con trỏ, mảng con trỏ có nhiều kiểu: Mỗi phần tử của mảng con trỏ kiểu int sẽ chứa được địa chỉ kiểu int. Tương tự, ta suy ra ý nghĩa của các mảng con trỏ kiểu char, float, ... Mảng con trỏ được khai báo theo mẫu "type *namearr[N]"; trong đó type có thể là int, float, double, char, ..., namearr là tên của mảng, N là một hằng số nguyên xác định độ lớn của mảng.
- Khi gặp khai báo trên máy sẽ cấp phát N khoảng nhớ liên tiếp cho N phần tử của mảng namearr. Ví dụ, câu lệnhDouble *pa[100];
- Khai báo một mảng con trỏ kiểu double gồm một trăm phần tử. Mỗi phần tử pa[i] có thể dùng để lưu trữ một địa chỉ kiểu double. Cũng như đối với các mảng khác, ta có thể khởi đầu cho các mảng con trỏ ngoài và tĩnh theo cách như đã trình bày ở chương 2.
- Chú ý rằng bản thân mảng con trỏ không thể dùng để lưu trữ số liệu. Tuy nhiên mảng con trỏ cho phép sử dụng các mảng khác để lưu trữ số liệu một cách có hiệu qủa hơn theo cách: Chia mảng thành các phần và ghi nhớ địa chỉ đầu của mỗi phần vào một phần tử của mảng con trỏ .
- Cũng nên lưu ý rằng trước khi sử dụng một mảng con trỏ cần gán cho mỗi phần tử của nó một giá trị. Giá trị này phải là địa chỉ của một biến hoặc của một phần tử mảng. Các phần tử của mảng con trỏ kiểu char có thể được khởi đầu bằng các xâu ký tự.

Ví dụ:
```
#include <stdio.h>
```

```
int main() {
char *ds[] = {"Pho", "Bun", "Com"};
for (int i = 0; i < 3; i++) {
printf("%s\n", ds[i]);}
return 0;}
```

5. Con trỏ hàm

*Cách khai báo con trỏ hàm và mảng con trỏ hàm:

- Ta trình bầy quy tắc khai báo thông qua các ví dụ.
- Tác dụng của câu lệnh: Float (*f)(float), (*mf[50])(int);

Là khai báo:

- F là con trỏ hàm kiểu float có đối float,
- Mf là mảng con trỏ hàm kiểu float có đối int (mảng có 50 phần tử).
- Tác dụng câu lệnh: Double (*g)(int, double), (*mg[30])(double, float);

Là khai báo:

- G là con trỏ hàm kiểu double có các đối int và double,
- Mg là mảng con trỏ hàm kiểu double có các đối double và float (mảng có 30 phần tử).

* Tác dụng của con trỏ hàm:

- Con trỏ hàm dùng để chứa địa chỉ của hàm. Muốn vậy ta thực hiện phép gán tên hàm cho con trỏ hàm. Để phép gán có nghĩa thì kiểu hàm và kiểu con trỏ phải tương thích. Sau phép gán, ta có thể dùng tên con trỏ hàm thay cho tên hàm. C cho phép thiết kế các hàm mà tham số thực trong lời gọi tới nó lại là tên của một hàm khác. Khi đó tham số hình thức tương ứng phải là một con trỏ hàm.

Ví dụ:
```
#include <stdio.h>
Int cong(int a, int b) {
   Return a + b;
}
Int main() {
```

```
    Int (*ptr)(int, int) = cong;
    Printf("Tong: %d\n", ptr(2, 3));
    Return 0;
}
```

6. Cấp phát động

- Biến dùng để lưu trữ dữ liệu trong chương trình. Biến cần được khai báo trước khi sử dụng. Thực chất của việc khai báo biến là xin cấp phát một vùng bộ nhớ dể lưu trữ dữ liệu. Việc khai báo biến đã đề cập trong các chương trước chính là việc cấp phát tĩnh bộ nhớ. Tuy nhiên, không phải khi nào số lượng và kích thước của các biến cũng xác định được ngay khi biên dịch, chẳng hạn nó có thể phụ thuộc vào các thông tin người dùng cung cấp khi thực hiện chương trình. Trong trường hợp này, bộ nhớ cần được cấp phát động.

- Để cấp phát động bộ nhớ, sử dụng hàm malloc trong thư viện stdlib.h: " Void *malloc(size_t size)"

- Trong đó:

  - Size là kích thước vùng nhớ cần cấp.
  - Size_t là một kiểu dữ liệu định sẵn trong thư viện stdlib.h (thông thường có thể khai báo kiểu unsigned int).
  - Hàm trả về con trỏ kiểu void chỉ đến ô nhớ đầu tiên của vùng nhớ được cấp. Nếu bộ nhớ không còn đủ, giá trị con trỏ là NULL.

- Để giải phóng vùng nhớ được trỏ bởi con trỏ p, sử dụng hàm free trong thư viện stdlib.h: "Void free(void *p)"

- Lưu ý: nếu giá trị p bằng NULL thì hàm free sẽ không làm gì.

Ví dụ:
```
#include <stdio.h>
#include <stdlib.h>
Int main() {
    Int *a, n;
    Printf("Nhap n: ");
    Scanf("%d", &n);
```

```
A = (int*)malloc(n * sizeof(int));
For (int i = 0; i < n; i++) {
   Printf("a[%d] = ", i);
   Scanf("%d", &a[i]);}
Free(a);
Return 0;}
```

7. Xử lý tệp

- Hàm fopen
  - Dạng hàm: FILE *fopen(const char *tên_tệp, const char *kiểu);
  - Công dụng: Hàm dùng để mở tệp. Nếu thành công hàm cho con trỏ kiểu FILE ứng với tệp vừa mở. Các hàm cấp 2 sẽ làm việc với tệp thông qua con trỏ này. Nếu có lỗi hàm trả về giá trị NULL.

  Chú ý: Trong các kiểu đọc/ghi, cần làm sạch vùng đệm trước khi chuyển từ đọc sang ghi hoặc từ ghi sang đọc. Các hàm fflush và hàm di chuyển đầu từ đều làm được chuyện này.

- Hàm fclose
  - Dạng hàm: Int fclose(FILE *fp);
  - Công dụng: Hàm dùng để đóng tệp. Nội dung đóng tệp gồm: Đẩy dữ liệu còn trong vùng đệm lên đĩa (khi đang ghi), xoá vùng đệm (khi đang đọc) và giải phóng biến fp để nó có thể dùng cho tệp khác. Nếu thành công hàm cho giá trị 0, trái lại hàm cho EOF.
  - Đối: fp là con trỏ tương ứng với tệp cần đóng.
- Hàm fcloseall
  - Dạng hàm: Int fcloseall(void);
  - Công dụng: Hàm dùng để đóng tất cả các tệp đang mở. Nếu thành công hàm cho giá trị nguyên bằng số tệp đóng được, trái lại hàm cho EOF.
- Hàm fflush

Dạng hàm: Int fflush(FILE *fp);

  - Công dụng: Hàm dùng làm sạch vùng đệm của tệp fp. Nếu thành công hàm cho giá trị 0, trái lại hàm cho EOF.
  - Đối: fp là con trỏ tệp.
- Hàm fflushall

- Dạng hàm: Int fflushall(void);
- Công dụng: Hàm dùng làm sạch vùng đệm của các tệp đang mở. Nếu thành công hàm cho giá trị nguyên bằng số tệp đang mở, trái lại hàm cho EOF.

- Hàm feof
  - Dạng hàm: Int feof(FILE *fp);
  - Công dụng: Hàm dùng để kiểm tra cuối tệp. Hàm cho giá trị khác 0 nếu gặp cuối tệp khi đọc, trái lại hàm cho giá trị 0.
  - Đối: fp là con trỏ tệp.

Ví dụ: #include <stdio.h>

```
Int main() {
   FILE *f = fopen("data.txt", "w");
   Fprintf(f, "Xin chao!\n");
   Fclose(f);
   F = fopen("data.txt", "r");
   Char str[50];
   Fgets(str, 50, f);
   Printf("Doc file: %s", str);
   Fclose(f);
   Return 0;}
```

8. Kiểu cấu trúc
   - Để lưu trữ và xử lý thông tin trong máy tính ta có các biến và các mảng. Mỗi biến chứa được một giá trị. Mảng có thể xem là tập hợp nhiều biến có cùng một kiểu giá trị và được biểu thị bằng một tên. Cấu trúc có thể xem như một sự mở rộng của các khái niệm biến và mảng, nó cho phép lưu trữ và xử lý các dạng thông tin phức tạp hơn. Cấu trúc là một tập hợp các biến, các mảng và được biểu thị bởi một tên duy nhất.
   - Để định nghĩa một cấu trúc, sử dụng từ khóa struct theo mẫu sau:

```
Struct tên_cấu_trúc{
Khai báo các thành phần};
```

- Lưu ý: thành phần của một cấu trúc có thể là một biến kiểu cơ bản hoặc có thể là một cấu trúc khác.
- Để định nghĩa một kiểu cấu trúc, sử dụng từ khóa typedef theo mẫu sau:

```
Typedef struct{
Khai báo các thành phần} kiểu_cấu_trúc;
```

- Sau khi xây dựng được kiểu_cấu_trúc, nó có thể được dùng để khai báo các biến cấu trúc.
- Các thành phần cơ bản của một cấu trúc là biến và mảng, nên một lẽ tự nhiên và cũng là một quy tắc cần ghi nhớ là việc xử lý một cấu trúc bao giờ cũng phải được thực hiện thông qua các thành phần cơ bản của nó.
- Để truy nhập tới các thành phần của cấu trúc cần thông qua tên của biến cấu trúc theo mẫu sau: " biến_cấu_trúc.tên_thành_phần ".

Ví dụ:

```
#include <stdio.h>
Struct MonAn {
  Char ten[30];
  Int gia;
};
Int main() {
  Struct MonAn ma = {"Pho", 30000};
  Printf("Mon: %s – Gia: %d\n", ma.ten, ma.gia);
  Return 0;}
```

9. Danh sách liên kết
   - Một Danh sách liên kết là một dãy các cấu trúc dữ liệu được kết nối với nhau thông qua các liên kết. Hiểu một cách đơn giản thì Danh sách liên kết là một cấu trúc dữ liệu bao gồm một nhóm các nút tạo thành một chuỗi. Mỗi nút gồm dữ liệu ở nút đó và tham chiếu đến nút kế tiếp trong chuỗi.
   - Danh sách liên kết là Cấu trúc có ít nhất một thành phần là con trỏ kiểu cấu trúc đang định nghĩa gọi là cấu trúc tự trỏ.
   - Cấu trúc tự trỏ được dùng để xây dựng danh sách liên kết (móc nối), đó là một nhóm các cấu trúc có tính chất sau:

- Biết địa chỉ cấu trúc đầu đang được lưu trữ trong một con trỏ nào đó (giả sử pdau).
- Trong mỗi cấu trúc (trừ cấu trúc cuối) chứa địa chỉ của cấu trúc tiếp theo của danh sách.
- Cấu trúc cuối chứa hằng NULL.

- Danh sách có 3 tính chất trên gọi là danh sách móc nối theo chiều thuận. Với danh sách này, ta có thể lần lượt truy nhập từ cấu trúc đầu tới cấu trúc cuối theo chiếu từ trên xuống dưới.
- Tương tự, danh sách liên kết theo chiều ngược cũng có 3 tính chất trên nhưng theo chiều ngược lai:

- Biết địa chỉ cấu trúc cuối.
- Trong mỗi cấu trúc (trừ cấu trúc đầu) chứa địa chỉ của cấu trúc trước.
- Cấu trúc đầu chứa hằng NULL.

- Với danh sách này, ta có thể lần lượt truy nhập từ cấu trúc cuối tới cấu trúc đầu theo chiều từ dưới lên trên. Ngoài ra có thể xây dựng các danh sách mà mỗi phần tử chứa hai địa chỉ: địa chỉ cấu trúc trước và địa chỉ cấu trúc sau. Với loại danh sách này, ta có thể truy nhập từ trên xuống dưới theo chiều thuận hoặc từ dưới lên trên theo chiều ngược.

Ví dụ:
```
#include<stdio.h>
#include<stdlib.h>
typedef struct MonAn{
    char ten[50];
    char ma[50];
    float gia;
    struct MonAn*tiep;
}Monan;
void hienthidanhsach(Monan*MA){
    if(MA==NULL){
        printf("CHUA CO MON AN!");
        return;}
    Monan*ma= MA;
```

```
        printf("\n==== DANH SACH MON AN ====\n");
        printf("STT\tTEN MON\t\t MA\tGIA\n");
        while(ma != NULL){
                printf("%s\t\t| %s\t|%.2f\n",ma->ten,ma->ma,ma->gia);
        ma=ma->tiep;}}
int main(){
        Monan*MA = NULL;
        hienthidanhsach(MA);}
```

B. ỨNG DỤNG

1. Ứng dụng con trỏ cấp phát động để lấy ví dụ minh hoạ về cấu trúc mảng

BÀI LÀM

ỨNG DỤNG QUẢN LÝ SINH VIÊN

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
Typedef struct SinhVien{
    Char ten[50];
    Char mssv[20];
    Int namsinh;
    Float diem;
}SinhVien;
Void nhap(int n, SinhVien *SV){
    Getchar();
    For(int i=0; i<n; i++){
        Printf("\n===Sinh vien thu %d===\n",i+1);
        Printf("Nhap ma so sinh vien: ");
        Fgets(SV[i].mssv, sizeof(SV[i].mssv),stdin);
  SV[i].mssv[strcspn(SV[i].mssv, "\n")] = 0;
        Printf("Nhap ten sinh vien: ");
        Fgets(SV[i].ten, sizeof(SV[i].ten),stdin);
  SV[i].ten[strcspn(SV[i].ten, "\n")] = 0;
    Printf("Nhap nam sinh: ");
    Scanf("%d",&SV[i].namsinh);
```

```
            Getchar();
            Printf(“Nhap diem trung binh: “);
            Scanf(“%f”,&SV[i].diem);
            Getchar();
            }
      }
      Void xuat(int n,SinhVien *SV){
            Printf(“\n===DANH SACH SINH VIEN===\n”);
            Printf(“STT |\t HO TEN \t\t|MSSV |\t NAM SINH |\tDIEM\n”);
                  For(int i=0 ; i<n; i++){
                        Printf(“%-4d|\t%-8s\t\t|%-5s|\t%-
10d|\t%.2f\n”,i+1,SV[i].ten,SV[i].mssv,SV[i].namsinh,SV[i].diem);
            }
      }
      Void timkiemtheoma(int n, SinhVien*SV){
            Char mssv[20];
            Int kiemtra=0;
            Printf(“Nhap ma sinh vien can tim: “);
            Getchar();
            Fgets(mssv, sizeof(mssv),stdin);
         Mssv[strcspn(mssv, “\n”)] = 0;
            For(int i=0; i<n; i++){
            If(strcmp(SV[i].mssv,mssv)==0){
                  Printf(“Co tim duoc sinh vien \n”,mssv);
                        Printf(“HO TEN \t\t|MSSV |\t NAM SINH
|\tDIEM\n”);
```

```
                Printf("%-8s\t\t|%-5s|\t%-
10d|\t%.2f\n",SV[i].ten,SV[i].mssv,SV[i].namsinh,SV[i].diem);
                Kiemtra=1;}}
        If(kiemtra==0){
                Printf("Khong tim thay sinh vien\n");
        }
        }
    Void timkiemtheoten(int n, SinhVien*SV){
        Char ten[50];
        Int kiemtra=0;
        Printf("Nhap ten sinh vien can tim: ");
        Getchar();
        Fgets(ten, sizeof(ten),stdin);
       Ten[strcspn(ten, "\n")] = 0;
        For(int i=0; i<n; i++){
        If(strcmp(SV[i].ten,ten)==0){
                Printf("Co tim duoc sinh vien \n");
                Printf("HO TEN \t\t|MSSV |\t NAM SINH
|\tDIEM\n");
                Printf("%-8s\t\t|%-5s|\t%-
10d|\t%.2f\n",SV[i].ten,SV[i].mssv,SV[i].namsinh,SV[i].diem);
                Kiemtra=1;
        }}
        If(kiemtra==0){
                Printf("Khong tim thay sinh vien\n");
        }
        }
```

```
Void themsinhvien(int n, SinhVien*SV){
SV = (SinhVien*)realloc(SV,(n+1)* sizeof(SinhVien));
    Printf("Nhap thong tin sinh vien can them: \n");
    Getchar();
    Printf("Nhap ma so sinh vien: ");
        Fgets(SV[n].mssv, sizeof(SV[n].mssv),stdin);
  SV[n].mssv[strcspn(SV[n].mssv, "\n")] = 0;
        Printf("Nhap ten sinh vien: ");
        Fgets(SV[n].ten, sizeof(SV[n].ten),stdin);
  SV[n].ten[strcspn(SV[n].ten, "\n")] = 0;
    Printf("Nhap nam sinh: ");
    Scanf("%d",&SV[n].namsinh);
    Getchar();
    Printf("Nhap diem trung binh: ");
    Scanf("%f",&SV[n].diem);
    Getchar();
    N++;
    }
Void timsinhvientheodiem(int n,SinhVien*SV){
    Float diem;
    Int kiemtra=0;
    Printf("Nhap diem can tim: ");
    Scanf("%f",&diem);
    Getchar();
    For(int i=0; i<n; i++){
    If(SV[i].diem==diem){
        Printf("Co tim duoc sinh vien \n");
```

```
                    Printf("HO TEN \t\t|MSSV |\t NAM SINH
|\tDIEM\n");
                    Printf("%-8s\t\t|%-5s|\t%-
10d|\t%.2f\n",SV[i].ten,SV[i].mssv,SV[i].namsinh,SV[i].diem);
                Kiemtra=1;}}
        If(kiemtra==0){
                    Printf("Khong tim thay sinh vien\n");
        }
                }
    Void timsinhvientheonam(int n,SinhVien*SV){
        Int ns;
        Int kiemtra=0;
        Printf("Nhap nam sinh can tim: ");
        Scanf("%d",&ns);
        Getchar();
        For(int i=0; i<n; i++){
        If(SV[i].namsinh==ns){
            Printf("Co tim duoc sinh vien \n");
                    Printf("HO TEN \t\t|MSSV |\t NAM SINH
|\tDIEM\n");
                    Printf("%-8s\t\t|%-5s|\t%-
10d|\t%.2f\n",SV[i].ten,SV[i].mssv,SV[i].namsinh,SV[i].diem);
                Kiemtra=1;}}
        If(kiemtra==0){
                    Printf("Khong tim thay sinh vien\n");
        }
                }
```

```
Void sinhvienmax(int n, SinhVien*SV){
    Float max=SV[0].diem;
    For(int i=0; i<n; i++){
        If(SV[i].diem>max){
            Max=SV[i].diem;
            }}
    Printf("Sinh vien co diem cao nhat bang: %.2f",max);
}
Void sinhvienmin(int n, SinhVien*SV){
    Float min=SV[0].diem;
    For(int i=0; i<n; i++){
        If(SV[i].diem<min){
            Min=SV[i].diem;}}
    Printf("Sinh vien co diem thap nhat bang: %.2f",min);
    }
Void xoasinhvien(int n, SinhVien*SV){
    Char mssv[50];
    Int vitri=-1;
    Printf("Nhap ma so sinh vien can xoa: ");
    Getchar();
    Fgets(mssv, sizeof(mssv),stdin);
  Mssv[strcspn(mssv, "\n")] = 0;
  For(int i=0; i<n; i++){
    If(strcmp(SV[i].mssv,mssv)==0){
        Vitri=i;}}
    If(vitri==-1){
        Printf("Khong tim thay sinh vien can xoa\n") ;
```

```
                Return;}
            For(int i =vitri; i<n-1;i++){
                SV[i]=SV[i+1];}
        n--;
        SV = (SinhVien*)realloc(SV,(n)* sizeof(SinhVien));
         Printf("Da xoa");
         }
    Int main(){
        Int luachon;
        Int n=0;
        SinhVien *SV=NULL;
        While (1) {
            Printf("\n===== QUAN LY SINH VIEN =====\n");
            Printf("1. NHAP THONG TIN SINH VIEN\n");
            Printf("2. XEM DANH SACH SINH VIEN\n");
            Printf("3. TIM KIEM SINH VIEN THEO MA\n");
            Printf("4. TIM KIEM SINH VIEN THEO TEN\n");
            Printf("5. THEM SINH VIEN VAO DANH SACH\n");
            Printf("6. TIM KIEM SINH VIEN THEO DIEM\n");
            Printf("7. TIM KIEM SINH VIEN THEO NAM SINH\n");
            Printf("8. SINH VIEN CO DIEM CAO NHAT\n");
            Printf("9. SINH VIEN CO DIEM THAP NHAT\n");
            Printf("10. XOA SINH VIEN THEO MA KHOI DANH
SACH\n");
            Printf("11. THOAT\n");
            Printf("Chọn chức năng: ");
            Scanf("%d", &luachon);
```

```
Switch (luachon) {
      Case 1:
      If (SV != NULL){
         Free(SV);
      }
         Do{
          Printf(“Nhap so luong sinh vien: “);
          Scanf(“%d”,&n);
   }while(n<=0 || n>100);
   SV = (SinhVien*)malloc(n* sizeof(SinhVien));
   Nhap(n,SV);
         Break;
       Case 2:
       Xuat(n,SV);
       Break;
       Case 3:
       Timkiemtheoma(n,SV);
       Break;
       Case 4:
       Timkiemtheoten(n,SV);
       Break;
       Case 5:
       SV = (SinhVien*)realloc(SV,(n+1)* sizeof(SinhVien));
       Themsinhvien(n++,SV);
       Break;
       Case 6:
       Timsinhvientheodiem(n,SV);
```

```
            Break;
            Case 7:
            Timsinhvientheonam(n,SV);
            Break;
            Case 8:
            Sinhvienmax(n,SV);
            Break;
            Case 9:
            Sinhvienmin(n,SV);
            Break;
            Case 10:
            Xoasinhvien(n--,SV);
            Break;
          Case 11:
          Free(SV);
               Return 0;
}
}
}
```

Hình B.1: Hình ảnh khi chạy ứng dụng quản lý sinh viên

2. Ứng dụng danh sách liên kết lấy tối thiểu 1 ví dụ

BÀI LÀM

ỨNG DỤNG QUẢN LÝ DANH SÁCH CÁC MÓN ĂN

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
Typedef struct MonAn
{
        Char ten[50];
        Char ma[50];
        Float gia;
        Struct MonAn*tiep;
}Monan;
Monan*themmonan(){
        Monan*moi=(Monan*)malloc(sizeof(Monan));
        Printf("Nhap ten mon an: ");
        Getchar();
        Fgets(moi->ten,sizeof(moi->ten),stdin);
        Moi->ten[strcspn(moi->ten, "\n")]=0;
        Printf("Nhap ma mon an: ");
        Fgets(moi->ma,sizeof(moi->ma),stdin);
        Moi->ma[strcspn(moi->ma, "\n")]=0;
        Printf("Nhap gia mon an: ");
        Scanf("%f",&moi->gia);
        Return moi;
}
```

```
Void hienthidanhsach(Monan*MA){
        Int stt=1;
        If(MA==NULL){
                Printf(“CHUA CO MON AN!”);
                Return;}
        Monan*ma= MA;
        Printf(“\n==== DANH SACH MON AN ====\n”);
        Printf(“STT\tTEN MON\t\t MA\tGIA\n”);
        While(ma != NULL){
                Printf(“%d\t|%s\t\t| %s\t|%.2f\n”,stt,ma->ten,ma->ma,ma->gia);
        Ma=ma->tiep;
        Stt++;
}
}
Monan*timmonantheoma(Monan*MA){
        Char ma1[50];
        Int kiemtra=0;
        Printf(“Nhap ma mon an can tim: “);
        Getchar();
        Fgets(ma1,sizeof(ma1),stdin);
        Ma1[strcspn(ma1, “\n”)]=0;
        Monan*ma=MA;
        While(ma != NULL){
                If(strcmp(ma->ma,ma1)==0){
                        Printf(“CO TIM THAY MON AN: \n”);
                        Printf(“TEN MON\t\t MA\tGIA\n”);
```

```
                Printf("%s\t\t| %s\t|%.2f\n",ma->ten,ma->ma,ma->gia);
                Kiemtra++;}
            Ma=ma->tiep;
        }
        If(kiemtra==0){
        Printf("Khong tim thay mon an!\n");
    Return NULL;}
}
Monan*timmonantheogia(Monan*MA){
        Float giamon;
        Int kiemtra=0;
        Printf("Nhap gia mon an can tim: ");
        Scanf("%f",&giamon);
        Getchar();
        Monan*ma=MA;
        While(ma != NULL){
            If(ma->gia==giamon){
                Printf("CO TIM THAY MON AN: \n");
                Printf("TEN MON\t\t MA\tGIA\n");
                Printf("%s\t\t| %s\t|%.2f\n",ma->ten,ma->ma,ma->gia);
                Kiemtra++;}
            Ma=ma->tiep;
        }
        If(kiemtra==0){
        Printf("Khong tim thay mon an!\n");
    Return NULL;}
```

```
}
Monan*timmonantheokgia(Monan*MA){
        Float giamonmin,giamonmax;
        Int kiemtra=0;
        Do{
        Printf(“Nhap gia mon an tu:  “);
        Scanf(“%f”,&giamonmin);
        Printf(“Den: “);
        Scanf(“%f”,&giamonmax);
        }while(giamonmin>=giamonmax);
        Monan*ma=MA;
        While(ma != NULL){
                If(ma->gia>=giamonmin && ma->gia<=giamonmax){
                        Printf(“CO TIM THAY MON AN: \n”);
                        Printf(“TEN MON\t\t MA\tGIA\n”);
                        Printf(“%s\t\t| %s\t|%.2f\n”,ma->ten,ma->ma,ma->gia);
                        Kiemtra++;}
                Ma=ma->tiep;
        }
        If(kiemtra==0){
        Printf(“Khong tim thay mon an!\n”);
   Return NULL;}
}
Monan*timmonantheoten(Monan*MA){
        Char ten1[50];
        Int kiemtra=0;
```

```
        Printf(“Nhap ma mon an can tim: “);
        Getchar();
        Fgets(ten1,sizeof(ten1),stdin);
        Ten1[strcspn(ten1, “\n”)]=0;
        Monan*ma=MA;
        While(ma != NULL){
                If(strcmp(ma->ten,ten1)==0){
                        Printf(“CO TIM THAY MON AN: \n”);
                        Printf(“TEN MON\t\t MA\tGIA\n”);
                        Printf(“%s\t\t| %s\t|%.2f\n”,ma->ten,ma->ma,ma->gia);
                        Kiemtra++;}
                Ma=ma->tiep;
        }
        If(kiemtra==0){
        Printf(“Khong tim thay mon an!\n”);
   Return NULL;}
}
Monan*monanmax(Monan*MA){
        Float max= MA->gia;
        Monan*datnhat=MA;
        Monan*ma= MA->tiep;
        While(ma != NULL){
                If(ma->gia>max){
                        Max=ma->gia;
                        Datnhat=ma;}
                        Ma=ma->tiep;
```

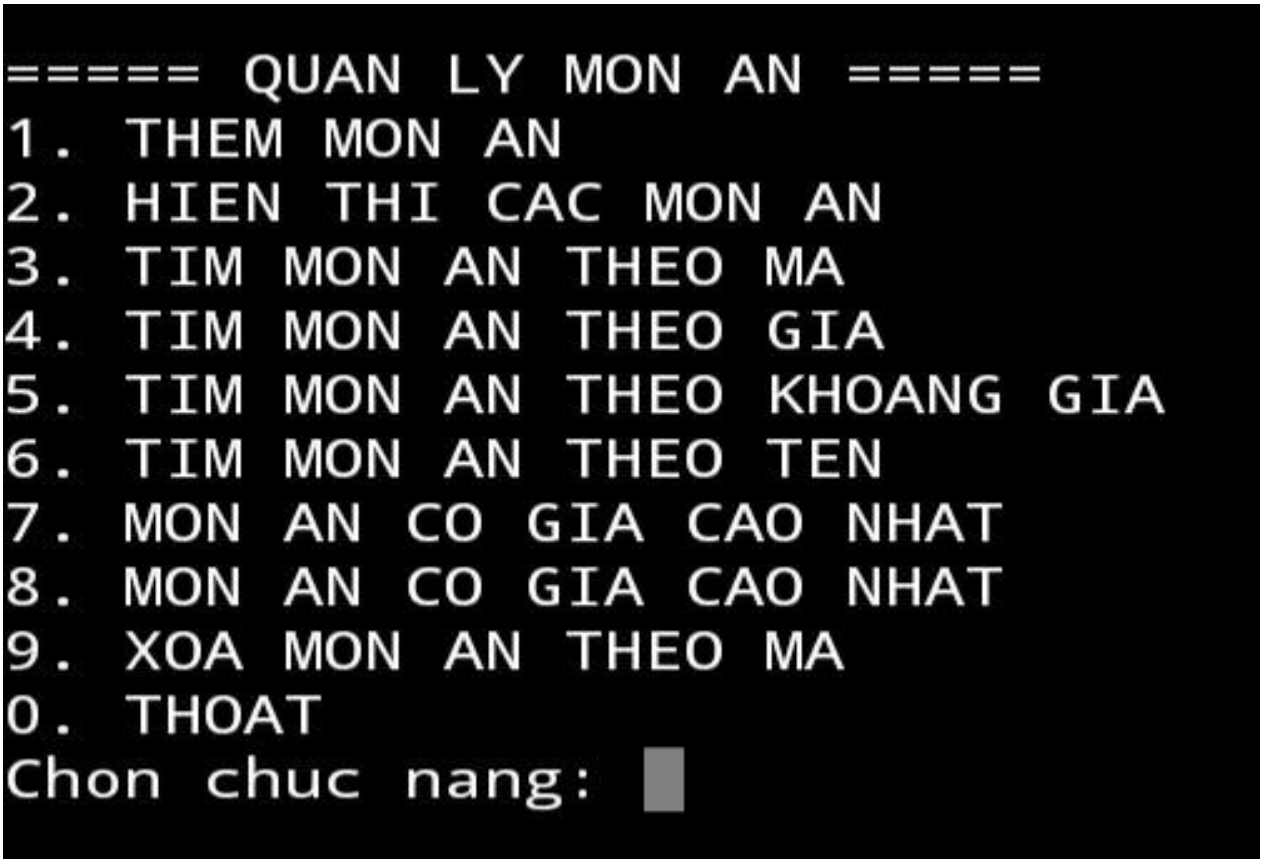
```
        }
        Printf("\n====  MON AN DAT NHAT  ====\n");
        Printf("TEN MON\t\t MA\tGIA\n");
        Printf("%s\t\t| %s\t|%.2f\n",datnhat->ten,datnhat->ma,datnhat->gia);
}
Monan*monanmin(Monan*MA){
        Float min= MA->gia;
        Monan*renhat=MA;
        Monan*ma= MA->tiep;
        While(ma != NULL){
                If(ma->gia<min){
                        Min=ma->gia;
                        Renhat=ma;}
                        Ma=ma->tiep;
        }
        Printf("\n====   MON AN RE NHAT   ====\n");
        Printf("TEN MON\t\t MA\tGIA\n");
        Printf("%s\t\t| %s\t|%.2f\n",renhat->ten,renhat->ma,renhat->gia);}
Void xoamonantheoma(Monan*MA) {
        Char macanxoa[50];
        Printf("Nhap ma mon an can xoa: ");
        Getchar();
        Fgets(macanxoa,sizeof(macanxoa),stdin);
        Macanxoa[strcspn(macanxoa, "\n")]=0;
    Monan* a = MA;
    Monan* b = NULL;
```

```
   If (strcmp(a->ma, macanxoa) == 0) {
      MA = a->tiep;
      Free(a);
      Printf("Da xoa mon an co ma: %s\n", macanxoa);
      Return;
   }
   While (a != NULL){
   If(strcmp(a->ma, macanxoa) == 0) {
      Break;}
      B = a;
      A = a->tiep;
   }
   If (a == NULL) {
      Printf("Khong tim thay mon an co ma: %s\n", macanxoa);
      Return;
   }
   b->tiep = a->tiep;
   free(a);
   printf("Da xoa mon an co ma: %s\n", macanxoa);
   return;
}
Int main(){
        Int luachon;
        Monan*MA = NULL;
        Monan*ma;
        While(1){
```

```
        Printf(“\n===== QUAN LY MON AN =====\n”);
        Printf(“1. THEM MON AN\n”);
        Printf(“2. HIEN THI CAC MON AN\n”);
        Printf(“3. TIM MON AN THEO MA\n”);
        Printf(“4. TIM MON AN THEO GIA\n”);
        Printf(“5. TIM MON AN THEO KHOANG GIA\n”);
        Printf(“6. TIM MON AN THEO TEN\n”);
        Printf(“7. MON AN CO GIA CAO NHAT\n”);
        Printf(“8. MON AN CO GIA CAO NHAT\n”);
        Printf("9. XOA MON AN THEO MA\n");
        printf("0. THOAT\n");
        printf("Chon chuc nang: ");
        scanf("%d",&luachon);
switch(luachon){
        case 1:
        ma= themmonan();
        if(MA==NULL){
                MA=ma;}
        else{
                Monan*p=MA;
        while(p->tiep !=NULL){
                p=p->tiep;}
        p->tiep=ma;}
        break;
        case 2:
        hienthidanhsach(MA);
```

```
break;
case 3:
if(MA==NULL){
printf("Danh sach trong, khong tim thay mon an");}
else{
timmonantheoma(MA);}
break;
case 4:
if(MA==NULL){
printf("Danh sach trong, khong tim thay mon an");}
else{
timmonantheogia(MA);}
break;
case 5:
if(MA==NULL){
printf("Danh sach trong, khong tim thay mon an");}
else{
timmonantheokgia(MA);}
break;
case 6:
if(MA==NULL){
printf("Danh sach trong, khong tim thay mon an");}
else{
timmonantheoten(MA);}
break;
case 7:
```

```
            if(MA==NULL){
        printf("Danh sach trong, khong tim thay mon an");}
        else{
        monanmax(MA);}
        break;
        case 8:
        if(MA==NULL){
        printf("Danh sach trong, khong tim thay mon an");}
        else{
        monanmin(MA);}
        break;
        case 9:
        if(MA==NULL){
        printf("Danh sach trong, khong tim thay mon an");}
        else{
        xoamonantheoma(MA);}
        break;
        case 0:
        return 0;
    }
}
}
```



Hình B.2: Hình ảnh khi chạy ứng dụng quản lý danh sách các món ăn