



Chuyên đề STRUCT và FILE lập trình C

Anh Quân đz (LA Quân) + Huy

Tài liệu được minh tống hợp (cx ko đầy đủ lăm
nhưng cũng gần vừa vặn giải đề) .kk . Code 8 bài vd
là của anh Quân đz được minh tham khảo và edit
thêm một chút .hhi. Có lỗi false các bạn góp ý nha

<3 <3 <3 .

P T I T

Contents

Phần 1:Hướng dẫn sử dụng tập tin (File) trong C	2
Một số khái niệm	2
Cách thao tác với file.....	2
Quy trình xử lý File	3
Khai báo biến tập tin	3
Mở tập tin	4
Kết thúc file	5
Đóng tập tin	5
Kiểm tra đến cuối tập tin hay chưa?.....	6
Các hàm đọc ghi nội dung tập tin dạng text stream	6
-lấy thông tin File :	6
-Ghi file :	6
Các hàm đọc ghi nội dung tập tin dạng binary stream.....	7
-Ghi file:fwrite:.....	7
-đọc File : fread:	7
Phần 2: Một số kiến thức cần biết :	8
1. Tác dụng của câu lệnh while(getchar()!='\n'); trong lập trình C?	8
2. EOF là gì?.....	8
3. Hàm fgetc() trong C	9
4. Toán tử sizeof với struct (C)	9
5. Hàm fseek() trong C	10
Phần 3: Giải 8 bài tập ví dụ	11
quản lý thông tin sinh viên (tổng điểm tăng dần).....	11
quản lý thông tin sinh viên (điểm thành phần tăng dần)	14
quản lý thông tin sản phẩm (lãi giảm dần)	18
quản lý thông tin sản phẩm (lãi gấp đôi).....	21
quản lý thông tin sách (sắp xếp tăng dần theo giá sách)	25
quản lý thông tin thiết bị (sắp xếp giảm dần theo giá nhập)	28
quản lý tiền cá nhân (thống kê kết quả)	32

Phần 1:Hướng dẫn sử dụng tập tin (File) trong C

Một số khái niệm

Dữ liệu trong chương trình được lưu trữ ở RAM máy tính, vì thế khi kết thúc chương trình, tắt máy dữ liệu sẽ bị giải phóng (mất dữ liệu). Để tránh vấn đề đó dữ liệu cần phải lưu trữ trên bộ nhớ ngoài (đĩa cứng, USB, ...) dưới dạng file. File có các đặc trưng sau:

- Là một đơn vị lưu trữ logic.
- Được biểu thị bằng một tên.
- Bao gồm một tập hợp dữ liệu do người tạo xác định.
- Được lưu trữ trên thiết bị lưu trữ ngoài bằng cách ánh xạ lên đơn vị vật lý của thiết bị.

Hầu hết các chương trình đều cần phải lưu trữ dữ liệu sau khi xử lý. Vì vậy C cung cấp cho chúng ta các kỹ thuật xử lý lưu trữ trên file.

Cách thao tác với file

Thao tác chuẩn: người lập trình không cần biết quá trình thực hiện việc thao tác với file như thế nào. Đó là việc của hệ thống.

Thao tác mức hệ thống (thao tác thủ công): thao tác file thông qua bộ đệm (buffer – một vùng nhớ). Người lập trình phải tự quản lý các bộ đệm đọc ghi file. Thao tác file này gần giống với cách thao tác file của hệ điều hành MS – DOS. Thông thường chỉ có những người lập trình hệ thống mới sử dụng thao tác file mức hệ thống.

Thao tác với file là phải thao tác với phần cứng. Do đó, việc thao tác với file có thể thành công hoặc thất bại.

Ngôn ngữ lập trình C xem file như là một dòng (stream) các byte, với các thiết bị xuất nhập theo từng byte cũng được xem là file, C định nghĩa sẵn các tên cho các thiết bị này và các file này đã được mặc định mở sẵn cho ta truy xuất ngay khi mở máy tính.

Handle	Tên	Thiết bị
0	stdin	Standard input – Thiết bị nhập chuẩn – Bên phím
	stdout	Standard output – Thiết bị xuất chuẩn – Màn hình
1	stderr	Standard error – Thiết bị xuất lỗi chuẩn – Màn hình
2	stdaux	Standard auxililary – Thiết bị ngoại vi chuẩn – Cổng nối tiếp
3	stdprn	Standard printer – Thiết bị in chuẩn – Máy in song song

Tên và ý nghĩa trong ứng của các lệnh trong thư viện chuẩn

Khi thao tác với file, ở mỗi thời điểm chỉ truy xuất được một phần tử lưu trữ trong file. Vị trí hiện hành đang thao tác (file position) gọi là chỉ số trong file hay con trỏ file – chính là số thứ tự của phần tử truy xuất hiện hành. Chỉ số thứ tự này bắt đầu từ 0.

File văn bản (Text File): là loại tập tin dùng để ghi các ký tự lên đĩa.

- Điểm đặc biệt là dữ liệu của tập tin được lưu trữ thành các dòng, mỗi dòng được kết thúc bằng ký tự xuống dòng (new line), ký hiệu ‘\n’;
- Mỗi tập tin được kết thúc bởi ký tự EOF (End Of File) có mã Ascii là 26 (xác định bởi tổ hợp phím Ctrl + Z).
- Truy xuất tập tin theo kiểu văn bản chỉ có thể truy xuất theo kiểu tuần tự.

File nhị phân: Dữ liệu được xem như là một dãy byte liên tục, việc xử lý với tập tin dựa trên việc đọc ghi dãy byte.

- Biến tập tin: là một biến thuộc kiểu dữ liệu tập tin dùng để đại diện cho một tập tin. Dữ liệu chứa trong một tập tin được truy xuất qua các thao tác với thông số là biến tập tin đại diện cho tập tin đó.
- Con trỏ tập tin: Khi một tập tin được mở ra để làm việc, tại mỗi thời điểm, sẽ có một vị trí của tập tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra. Người ta hình dung có một con trỏ đang chỉ đến vị trí đó và đặt tên nó là con trỏ tập tin.
- Sau khi đọc/ghi xong dữ liệu, con trỏ sẽ chuyển dịch thêm một phần tử về phía cuối tập tin. Sau phần tử dữ liệu cuối cùng của tập tin là dấu kết thúc tập tin EOF (End Of File).

Quy trình thao tác với tập tin (File)

- Khai báo biến tập tin.
- Mở tập tin bằng hàm fopen().
- Thực hiện các thao tác xử lý dữ liệu của tập tin bằng các hàm đọc/ghi dữ liệu.
- Đóng tập tin bằng hàm fclose().

Ở đây, ta thao tác với tập tin nhờ các hàm được định nghĩa trong thư viện stdio.h.

Quy trình xử lý File

Bước 1: Mở file.

- Xác định chế độ mở chính xác(text/binary).
- Kiểm tra lỗi.

Bước 2: Truy xuất xử lý.

- Áp dụng hợp lý các hàm truy xuất tùy theo chế độ mở.
- Quản lý con trỏ chỉ vị trí.
- Kiểm tra lỗi.

Bước 3: Đóng file nhằm đảm bảo tính toàn vẹn dữ liệu.

Khai báo biến tập tin

Cú pháp

FILE <Danh sách các biến con trỏ, đại diện cho tập tin>

Thao tác file chuẩn

FILE * f;

Thao tác mức hệ thống

int f;

Dữ liệu trên file là một dãy các byte(8 bit) có giá trị từ 0 đến 255. Số byte của dãy là kích thước thật của file(size on disk).

Các biến trong danh sách phải là các con trỏ và được phân cách bởi dấu phẩy(,). Ví dụ: FILE

*f1,*f2;

Mở tập tin

Cú pháp

FILE *fopen(char *Path, const char *Mode)

Trong đó:

- Path: chuỗi chỉ đường dẫn đến tập tin trên đĩa.
- Mode: chuỗi xác định cách thức mà tập tin sẽ mở. Các giá trị có thể của Mode:

Chế độ	Ý nghĩa
r	Mở tập tin văn bản để đọc
w	Tạo ra tập tin văn bản mới để ghi
a	Nối vào tập tin văn bản
rb	Mở tập tin nhị phân để đọc
wb	Tạo ra tập tin nhị phân để ghi
ab	Nối vào tập tin nhị phân
r+	Mở một tập tin văn bản để đọc/ghi
w+	Tạo ra tập tin văn bản để đọc/ghi
a+	Nối vào hay tạo mới tập tin văn bản để đọc/ghi
r+b	Mở ra tập tin nhị phân để đọc/ghi
w+b	Tạo ra tập tin nhị phân để đọc/ghi
a+b	Nối vào hay tạo mới tập tin nhị phân

Mặc định là mở dạng text nếu không có xác định là b, nếu rõ ràng hơn thì thêm chỉ định t để xác định là kiểu text.

Hàm fopen trả về một con trỏ tập tin. Chương trình của ta không thể thay đổi giá trị của con trỏ này. Nếu có một lỗi xuất hiện trong khi mở tập tin thì hàm này trả về con trỏ NULL.

Ví dụ: Mở một tập tin tên TEST.txt để ghi.

1 FILE *f;

2 f = fopen("TEST.txt", "w");

3 if (f!=NULL)

4 {

```

5  /* Các câu lệnh để thao tác với tập tin*/
6  /* Đóng tập tin*/
7 }

```

Kiểm tra con trỏ f với giá trị NULL cho phép xác định được lệnh thực hiện thành công hay không? Nếu mở tập tin để ghi, trường hợp tập tin đã tồn tại rồi thì tập tin sẽ bị xóa và một tập tin mới được tạo ra. Nếu ta muốn ghi nối dữ liệu, ta phải sử dụng chế độ “a”. Khi mở với chế độ đọc, tập tin phải tồn tại rồi, nếu không một lỗi sẽ xuất hiện.

Kết thúc file

Sau khi tạo xong một file văn bản, đóng file này, byte mang giá trị 1Ah (26 của hệ 10 – tương đương với khi gõ tổ hợp phím Ctrl + Z) sẽ tự động chèn vào cuối file để ấn định hết file.

Nói chung, file được quản lý bằng kích thước của file (số bytes). Khi đã đọc hết số byte có trong file, thì dấu kí hiệu EOF (end of file) được DOS thông báo cho chương trình. Dấu hiệu EOF là tên hằng mà C khai báo sẵn trong thư viện STDIO.H và nó mang giá trị -1.

Như vậy, nếu một file dữ liệu (có cả số) được mở dạng văn bản, nếu trong giữa file mà có giá trị 1Ah thì quá trình đọc sẽ bị ngưng nửa chừng (hàm đọc file sẽ trả về giá trị -1 cho chương trình báo đã kết thúc file).

Chỉ định file ở chế độ	Dữ liệu trong chương trình C	Được lưu trữ trên file
Văn bản	, „\n“ EOF	CR (13), LF (10) 1Ah
Nhị phân	, „\n“, „\r“, „\n“ 1Ah (số)	LF CR LF 1Ah

Sự tương ứng giữa dữ liệu trong chương trình C và dữ liệu trên file

Đóng tập tin

Hàm **fclose()** được dùng để đóng tập tin được mở bởi hàm **fopen()**. Hàm này sẽ ghi dữ liệu còn lại trong vùng đệm vào tập tin và đóng lại tập tin.

Cú pháp: **int fclose(FILE *f)**

Trong đó **f** là con trỏ tập tin được mở bởi hàm **fopen()**. Giá trị trả về của hàm là 0 báo rằng việc đóng tập tin thành công. Hàm trả về **EOF** nếu có xuất hiện lỗi.

Ngoài ra, ta còn có thể sử dụng hàm **fcloseall()** để đóng tất cả các tập tin lại.

Cú pháp: **int fcloseall()**

Kết quả trả về của hàm là tổng số các tập tin được đóng lại. Nếu không thành công, kết quả trả về là EOF.

Kiểm tra đến cuối tập tin hay chưa?

Cú pháp: `int feof(FILE *f)`

Ý nghĩa: Kiểm tra xem đã chạm tới cuối tập tin hay chưa và trả về EOF nếu cuối tập tin được chạm tới, ngược lại trả về 0.

Trả về -1 : Đã hết file

0 : Chưa hết file

Số nguyên khác : Mô tả lỗi

Các hàm đọc ghi nội dung tập tin dạng text stream

-lấy thông tin File :

+ `fscanf(<FILE *>, <định dạng>, <các tham biến>);` Dữ liệu từ một tập tin theo định dạng

```
fscanf(fp, "%s %s %s %d", str1, str2, str3, &year);
```

+ `fgets(<vùng nhớ>, <kích thước tối đa>, <FILE *>);` Đọc một chuỗi ký tự từ một tập tin với kích thước tối đa cho phép, hoặc gặp ký tự xuống dòng.

```
char str[60];
if( fgets (str, 60, fp)!=NULL ) {
    /* writing content to stdout */
    puts(str);
}
```

+ `fgetc(<FILE * >);` Đọc một ký tự từ tập tin đang mở.

```
c = fgetc(fp);
```

+Xóa bộ nhớ đệm:

```
fflush(stdin);
```

-Ghi file :

+ `fprintf(<FILE *>, <định dạng>[, <các tham biến>]);` Ghi dữ liệu theo một định dạng nào đó vào tập tin.

Ví dụ : `fprintf(f,"%d",x);`

+ `fputs(<chuỗi ký tự>, <FILE *>);` Ghi một chuỗi ký tự vào tập tin đang mở.

```
fputs("This is c programming.", fp);
```

vd://ghi chuoi n xuong file text

```
fputs("Nguyen Thi Le",fp);
```

```
fputc("\n',fp);// phải thêm xuống dòng
```

Các hàm đọc ghi nội dung tập tin dạng binary stream

-Ghi file:fwrite:

```
size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream)
```

Tham số

- **ptr** – Đây là con trỏ tới mảng các phần tử được ghi.
- **size** – Đây là kích cỡ (giá trị byte) của mỗi phần tử được ghi.
- **nmemb** – Đây là số phần tử, với mỗi phần tử có kích cỡ là size byte.
- **stream** – Đây là con trỏ tới một đối tượng FILE mà xác định một Output Stream.

Trả về giá trị

Hàm này trả về tổng số phần tử được trả về thành công dưới dạng một đối tượng size_t, mà là một kiểu dữ liệu nguyên. Nếu số này khác tham số nmemb, nó sẽ chỉ một lỗi.

```
#include<stdio.h>
int main ()
{
    FILE *fp;
    char str[] = "Hoc C co ban va nang cao tai VietJack !!!";
    fp = fopen( "baitapc.txt" , "w" );
    fwrite(str , 1 , sizeof(str) , fp );
    fclose(fp);
    return(0);
}//Biên dịch và chạy chương trình output : Hoc C co ban va nang cao tai VietJack !!!
```

-đọc File : fread:

Dưới đây là phần khai báo cho **hàm fread()** trong C:

```
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)
```

Tham số

- **ptr** – Đây là con trỏ tới một khối bộ nhớ với kích cỡ tối thiểu là **size nmemb byte**.
- **size** – Đây là kích cỡ (giá trị byte) của mỗi phần tử được đọc.
- **nmemb** – Đây là số phần tử, với mỗi phần tử có kích cỡ là size byte.
- **stream** – Đây là con trỏ tới một đối tượng FILE mà xác định một Input Stream.

Trả về giá trị

Tổng số phần tử được đọc thành công được trả về dưới dạng một đối tượng **size_t**, mà là một kiểu dữ liệu nguyên. Nếu số này khác với tham số nmemb, thì hoặc một lỗi đã xảy ra hoặc bắt gặp End-Of-File.

```
Vd: #include <stdio.h>
#include <string.h>
int main()
```

```

{
FILE *fp;
char c[] = "Hoc C co ban va nang cao tai Hoclaptoptrinh !!!";
char buffer[100]; /* mo file de doc va ghi */
fp = fopen("baitapc.txt", "w+"); /* Ghi du lieu vao file */
fwrite(c, strlen(c) + 1, 1, fp); /* thiet lap vi tri con tro tim kiem ve dau file */
fseek(fp, SEEK_SET, 0); /* Doc va hien thi du lieu */
fread(buffer, strlen(c)+1, 1, fp);
printf("%s\n", buffer);
fclose(fp);

return(0);
}

```

VD : fread(&ss[i],sizeof(sv),1,outfile);

Phần 2: Một số kiến thức cần biết :

1. Tác dụng của câu lệnh while(getchar()!='\n'); trong lập trình C?

hàm getchar() trả về mã phím do người dùng ấn

'\n' là mã phím của nút enter.

khi người dùng ấn enter thì getchar() trả về '\n'

while(getchar()!='\n');

** Chức năng của vòng lặp while này là xóa đầu vào bất hợp pháp của người dùng. Khi đầu vào là '\n', vòng lặp kết thúc. Khi bạn cung cấp đầu vào cho chương trình, sau đó bạn kết thúc nó bằng Enter phím. Khóa này được gửi đến chương trình của bạn dưới dạng một dòng mới.

Như đã đề cập bởi những người khác, vòng lặp này sẽ loại bỏ các ký tự không mong muốn stdin để hàm đầu vào tiếp theo có một luồng sạch và đặc biệt là nó loại bỏ '\n' ký tự theo sau ký tự cuối cùng được nhập bởi người dùng. Nhưng, getchar() trả về EOF trong trường hợp có lỗi đọc, do đó, vòng lặp cũng nên kiểm tra EOF

2. EOF là gì?

EOF - End Of File : chỉ sự kết thúc việc đọc của 1 file

EOF chỉ là một macro có giá trị (thường là -1). Bạn phải kiểm tra điều gì đó chống lại EOF, chẳng hạn như kết quả của cuộc gọi getchar().

Một cách để kiểm tra sự kết thúc của luồng là với hàm feof.

Bạn nên thay đổi ngoặc của bạn để

while((c = getchar()) != EOF)

Bởi vì "=" nhả điều hành có độ ưu tiên thấp hơn "!= " Điều hành. Sau đó, bạn sẽ nhận được kết quả mong đợi. Biểu thức của bạn bằng

while (c = (getchar()) != EOF)

Bạn nhận được kết quả của hai là đầu ra, vì bạn đang so sánh "c! = EOF". Điều này sẽ luôn luôn trở thành một cho ký tự bạn đã nhập và sau đó "\ n" theo sau bằng cách nhấn trả về. Ngoại trừ lần so sánh cuối cùng trong đó c thực sự là EOF, nó sẽ cung cấp cho bạn 0.

3. Hàm fgetc() trong C

```
int fgetc(FILE *stream);
```

Parameter:

stream: con trỏ file. Đây là con trỏ tới một đối tượng FILE mà nhận diện Stream trên đó hoạt động được thực hiện.

Remark:

Hàm fgetc() đọc 1 ký tự từ file.

Hàm fgetc() trả về mã ASCII của ký tự đọc được

Hàm trả về EOF nếu gặp lỗi hoặc cuối file.

4. Toán tử sizeof với struct (C)

Toán tử **sizeof** hay được dùng để tính kích thước bộ nhớ cần khi cấp phát động. Để diễn tả một cách dễ hiểu thì ví dụ là hay nhất!!!

```
struct HS {  
    char name;  
    int id;  
};
```

Bạn đoán thử sizeof(HS) = ???

Có lẽ bạn sẽ đoán nó = 5 (sizeof(char) = 1, sizeof(int) = 4). Nhưng trình biên dịch lại cho kết quả là 8??? Vì sao vậy?

Cơ chế cấp phát là như sau:

- Đầu tiên nó sẽ lấy kích thước của biến lớn nhất trong struct HS là sizeof(**id**) = 4 cấp phát 4 byte để chứa biến **name** (kích thước = 1 byte) còn dư 3 byte.
- Tiếp theo nó sẽ chứa tiếp biến **id** (kích thước 4 byte) mà chỉ còn có 3 byte (không đủ) nên bộ nhớ sẽ cấp phát thêm 4 byte để chứa biến **id**.
- => Sau khi cấp phát: 8 byte (còn dư 3 byte) !!!

Đến đây chắc bạn có câu trả lời rồi nhỉ? ^^. Để hiểu thêm ta có 1 ví dụ típ theo như sau:

```
struct SV {  
    int DiemAV;  
    double diemToan;  
    double diemLT;  
    int diemRL;  
};
```

Bạn đoán xem **sizeof(SV) = ???**

Đáp án là 32 byte!!! (dư 8 byte). Vậy phải làm sao để tiết kiệm được 8 byte này???

Đơn giản nhất là sắp các biến trong struct SV lại thôi!!!

```

struct SV {
    int DiemAV;
    int diemRL;
    double diemToan;
    double diemLT;
};

Và bây giờ sizeof(SV) = 24 byte !!! ^^
struct SV { };
Now, sizeof(SV) = ????
=> 1 byte!!!
Mặc định sẽ dành 1 byte để chứa struct!!!

```

5. Hàm fseek() trong C

Dưới đây là phần khai báo cho hàm fseek() trong C:

`int fseek(FILE *stream, long int offset, int whence)`

Tham số

- stream – Đây là con trỏ tới một đối tượng FILE mà nhận diện Stream.
- offset – offset: Số byte sẽ di chuyển con trỏ file từ vị trí whence
- whence – vị trí ban đầu của con trỏ file. Dưới đây là một số hằng số đã được định nghĩa sẵn trong thư viện stdio.h

chú ý : Hàm fseek thay đổi vị trí trỏ đến vị trí trong tập tin. Nó thường sử dụng để ghi dữ liệu vào vị trí mong muốn

- Hàm fseek() trả về 0 nếu dịch con trỏ file thành công
- Hàm trả về giá trị khác 0 nếu xảy ra lỗi.

Ví dụ minh họa cách sử dụng của hàm fseek() trong C:

```

#include <stdio.h>
int main () {
    FILE *fp;
    fp = fopen("file.txt", "w+");
    fputs("This is tutorialspoint.com", fp);

    fseek( fp, 7, SEEK_SET );
    fputs(" C Programming Language", fp);
    fclose(fp);
    return(0);
}

```

Biên dịch và chạy chương trình trên sẽ tạo file.txt với nội dung sau. Lúc đầu, chương trình tạo file và ghi *This is tutorialspoint.com* nhưng sau đó chúng ta đã phục hồi con trỏ ghi tại vị trí thứ 7 bắt đầu từ phần đầu file và sử dụng lệnh puts() mà ghi đè file với nội dung sau:

This is C Programming Language

Phần 3: Giải 8 bài tập ví dụ

quản lý thông tin sinh viên (tổng điểm tăng dần)

Thông tin về sinh viên gồm có: 1. mã sinh viên là số nguyên tự động tăng; 2. tên sinh viên; 3. điểm sinh viên là số thực gồm điểm môn A, môn B, môn C.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
2. cập nhật điểm theo mã sinh viên
3. hiển thị danh sách sinh viên theo tổng điểm tăng dần

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện
- Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sinh viên sẽ thêm mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sinh viên sẽ sửa đổi
- Dòng tiếp theo là thông tin sinh viên

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì in ra số sinh viên được thêm
- Nếu lựa chọn là 2 thì in ra mã sinh viên được sửa
- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

nguyen van hai

8.5 5.5 7.5

tran van tuan

8.5 .50 9.0

OUTPUT (1)

2

INPUT (2)

2

2

tran van nam

5.5 5.0 6.0

OUTPUT (2)

2

INPUT (3)

3

OUTPUT

2 tran van nam 5.5 5.0 6.0
1 nguyen van hai 8.5 5.5 7.5

Giải

```
#include<stdio.h>
#include<string.h>
#define flush while(getchar()!='\n')

struct sv{
    int nums;
    char name[50];
    float p[3];
} ;
sv ss[100];

int count(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int c=0;
    int temp= fgetc(outfile);
    while(temp!=fgetc(outfile)!=EOF){
        c++;
        fseek(outfile,sizeof(sv)*c,SEEK_SET)      ;
    }
    fclose(outfile);
    return c;
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c=count();
    for(int i=1;i<=n;i++){
        sv s;
        s.nums=i+c;
        flush;
        gets(s.name);
        for(int i=0;i<3;i++){
            scanf("%f",&s.p[i]);
        }
        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

void edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
```

```

fseek(outfile,sizeof(sv)*(n-1),SEEK_SET) ;
sv s;
s.nums=n;
flush;
gets(s.name);
for(int i=0;i<3;i++){
    scanf("%f",&s.p[i]);
}
fwrite(&s,sizeof(s),1,outfile);
}
int sumpoint(int n){
    float sum=0;
    for(int i=0;i<3;i++){
        sum+=ss[n].p[i];
    }
    return sum;
}

void sort(){
    int n=count();
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if(sumpoint(i)>sumpoint(j)){
                sv s=ss[i];
                ss[i]=ss[j];
                ss[j]=s;
            }
        }
    }
}

void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int n=count();
    int c=0;
    for (int i=0;i<n;i++){
        fseek(outfile,sizeof(sv)*i,SEEK_SET);
        fread(&ss[i],sizeof(sv),1,outfile);
    }
    sort();
    for (int i=0;i<n;i++){
        printf("%d %s",ss[i].nums,ss[i].name);
        for(int j=0;j<3;j++){
            printf(" %.1f",ss[i].p[j]);
        }
    }
}

```

```

        }
        printf("\n");
    }
    fclose(outfile);
}

int main(){
    int n;
    scanf("%d",&n);
    switch(n) {
        case 1:
            scanf("%d",&n);
            addnew(n);
            printf("%d",n);
            break;
        case 2:
            scanf("%d",&n);
            edit(n);
            printf("%d",n);
            break;
        case 3:
            display();
            break;
        default:
            printf("");
    }
    return 0;
}

```

quản lý thông tin sinh viên (điểm thành phần tăng dần)

Thông tin về sinh viên gồm có: 1. mã sinh viên là số nguyên tự động tăng; 2. tên sinh viên; 3. điểm sinh viên là số thực gồm điểm môn A, môn B, môn C.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
 2. cập nhật điểm theo mã sinh viên
 3. hiển thị danh sách sinh viên có điểm môn A, môn B, môn C tăng dần
- Mô tả dữ liệu vào:
- Dòng thứ nhất là lựa chọn chức năng thực hiện
 - Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sinh viên sẽ thêm mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sinh viên sẽ sửa đổi
 - Dòng tiếp theo là thông tin sinh viên

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì in ra số sinh viên đã được thêm
- Nếu lựa chọn là 2 thì in ra mã sinh viên đã được sửa
- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

nguyen van hai

8.5 5.5 7.5

tran van tuan

8.5 6.5 9.0

OUTPUT (1)

2

INPUT (2)

2

2

tran van nam

5.5 6.0 7.0

OUTPUT (2)

2

INPUT (3)

3

OUTPUT

2 tran van nam 5.5 6.0 7.0

#include<stdio.h>

#include<string.h>

#define flush while(getchar()!='\n')

```
struct sv{
    char name[50];
    int nums;
    float p[3];
}; sv ss[100];
```

int count(){

FILE *outfile=fopen("Huy314.bin","rb");

int c =0;

int temp=fgetc(outfile);

while(temp=fgetc(outfile)!=EOF){

c++;

```

        fseek(outfile,sizeof(sv)*c,SEEK_SET);
    }
    fclose(outfile);
    return c;
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c =count();
    for(int i=1;i<=n;i++){
        sv s;
        s.nums=i+c;
        flush;
        gets(s.name);
        for(int j=0;j<3;j++){
            scanf("%f",&s.p[j]);
        }
        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
    fseek(outfile,sizeof(sv)*(n-1),SEEK_SET);
    sv s;
    s.nums=n;
    flush;
    gets(s.name);
    for(int j=0;j<3;j++){
        scanf("%f",&s.p[j]);
    }
    fwrite(&s,sizeof(s),1,outfile);
}

/*void sort(int n){
    for (int i=0;i<2;i++){
        for(int j=1;j<3;j++){
            if(ss[n].p[i]>ss[n].p[j]) {
                float tg=ss[n].p[i];
                ss[n].p[i]=ss[n].p[j];
                ss[n].p[j]=tg;
            }
        }
}
```

```

        }
    } */
void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int n=count();
    int c=0;
    for(int i=1;i<n;i++){
        fseek(outfile,sizeof(sv)*i,SEEK_SET);
        fread(&ss[i],sizeof(sv),1,outfile);
    }
    for(int i=1;i<n;i++){
        if(ss[i].p[0]<ss[i].p[1]&&ss[i].p[1]<ss[i].p[2]){
            printf("%d %s",ss[i].nums,ss[i].name);
            for(int j=0;j<3;j++){
                printf(" %.1f",ss[i].p[j]);
            }
            printf("\n");
        }
    }
    fclose(outfile);
}

int main(){
    int n;
    scanf("%d",&n);
    switch (n){
        case 1:
            scanf("%d",&n);
            addnew(n);
            printf("%d",n);
            break;
        case 2:
            scanf("%d",&n);
            edit(n);
            printf("%d",n);
            break;
        case 3:
            display();
            break;
        default:
            printf("");
    }
    return 0;
}

```

}

quản lý thông tin sản phẩm (lãi giảm dần)

Thông tin về sản phẩm gồm có:

1. mã sản phẩm là số nguyên tự động tăng;
2. tên sản phẩm; 3. giá là số thực gồm giá nhập, giá xuất.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
2. cập nhật giá theo mã sản phẩm
3. hiển thị danh sách sản phẩm có chênh lệch giá xuất với giá nhập giảm dần

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện

- Dòng thứ hai

+ Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sản phẩm sẽ thêm mới

+ Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sản phẩm sẽ sửa đổi

- Dòng tiếp theo là thông tin sản phẩm

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì in ra số sản phẩm đã được thêm

- Nếu lựa chọn là 2 thì in ra mã sản phẩm đã được sửa

- Nếu lựa chọn là 3 thì hiển thị danh sách theo mô tả dưới đây: giá lấy 2 chữ số thập phân, các thông tin cách nhau một khoảng trắng

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

banh ngọt

5.50 8.50

nuoc ngọt

4.50 6.50

OUTPUT (1)

2

INPUT (2)

2

2

nuoc giai khat

4.50 10.50

OUTPUT (2)

INPUT (3)

3

OUTPUT

2 nuoc gai khat 4.50 10.50

1 banh ngọt 5.50 8.50

```
#include<stdio.h>
```

```
#include <string.h>
```

```
#define flush while(getchar()!='\n')
```

```
struct product{
    int nums;
    char name[50];
    float a,b;//chu y float
};product ss[100];
```

```
int count(){
    FILE *outfile= fopen("Huy314.bin","rb");
    int c=0;
    int temp= fgetc(outfile);
    while(temp!= fgetc(outfile)!= EOF){
        c++;
        fseek(outfile,sizeof(product)*c,SEEK_SET);
    }
    fclose(outfile);
    return c;
}
```

```
void addnew(int n){
    FILE *outfile= fopen("Huy314.bin","ab");
    int c=count();
    for(int i=1;i<=n;i++){
        product s;
        s.nums=i+c;
        flush;
        gets(s.name);
        scanf("%f%f",&s.a,&s.b);
        fseek(outfile,0,SEEK_END);
```

```

        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);

}

void edit(int n){
    FILE *outfile= fopen("Huy314.bin","rb+");
    fseek(outfile,sizeof(product)*(n-1),SEEK_SET);
    product s;
    s.nums=n;
    flush;
    gets(s.name);
    scanf("%f%f",&s.a,&s.b);
    fwrite(&s,sizeof(s),1,outfile);// edit ko co fclose(outfile);
}

void sort(){
    int n=count();
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if((ss[i].b-ss[i].a)<(ss[j].b-ss[j].a)){
                product s=ss[i];
                ss[i]=ss[j];
                ss[j]=s;
            }
        }
    }
}

void display(){
    FILE *outfile= fopen("Huy314.bin","rb");
    int n=count();
    int c=0;
    for(int i=0;i<n;i++){
        fseek(outfile,sizeof(product)*i,SEEK_SET);
        fread(&ss[i],sizeof(product),1,outfile);//fread
    }
    sort();
}

```

```

        for(int i=0;i<n;i++){
            printf("%d %s",ss[i].nums,ss[i].name);
            printf(" %.2f %.2f",ss[i].a, ss[i].b);
            printf("\n");
        }
        fclose(outfile);
    }

int main(){
    int n;
    scanf("%d",&n);
    switch (n){
        case 1:
            scanf("%d",&n);
            addnew(n);
            printf("%d",n);
            break;
        case 2:
            scanf("%d",&n);
            edit(n);
            printf("%d",n);
            break;
        case 3:
            display();
            break;
        default:
            printf("");
    }
    return 0;
}

```

quản lý thông tin sản phẩm (lãi gấp đôi)

Thông tin về sản phẩm gồm có: 1. mã sản phẩm là số nguyên tự động tăng; 2. tên sản phẩm; 3. giá là số thực gồm giá nhập, giá xuất.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
2. cập nhật giá theo mã sản phẩm

3. hiển thị danh sách sản phẩm có tiền lãi(giá xuất - giá nhập) lớn hơn giá nhập

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện

- Dòng thứ hai

+ Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sản phẩm sẽ thêm mới

+ Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sản phẩm sẽ sửa đổi

- Dòng tiếp theo là thông tin sản phẩm

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì in ra số sản phẩm được thêm

- Nếu lựa chọn là 2 thì in ra mã sản phẩm được sửa

- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây: giá lấy 2 chữ số thập phân, các thông tin cách nhau một khoảng trắng

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

banh ngọt

5.50 8.50

nuoc ngọt

4.50 6.50

OUTPUT (1)

2

INPUT (2)

2

2

nuoc gai khat

2.50 5.50

OUTPUT (2)

2

INPUT (3)

3

OUTPUT

2 nuoc gai khat 2.50 5.50

```
#include<stdio.h>
```

```
#include<string.h>
```

```

#define flush while(getchar()!="\n")//chu y !='\n

struct product{
    int nums;
    char name[50];
    float a,b;
}; product ss[100];

int count(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int c=0;
    int temp=fgetc(outfile);
    while(temp!=fgetc(outfile)!=EOF){
        c++;
        fseek(outfile,sizeof(product)*c,SEEK_SET);
    }
    fclose(outfile);
    return c;
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c=count();
    for(int i=1;i<=n;i++){
        product s;
        s.nums=c+i;
        flush;
        gets(s.name);
        scanf("%f%f",&s.a,&s.b);
        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

void edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
    fseek(outfile,sizeof(product)*(n-1),SEEK_SET);
    product s;

```

```

s.nums=n;
flush;
gets(s.name);
scanf("%f%f",&s.a,&s.b);//chu y &s.a khong phai a
fwrite(&s,sizeof(s),1,outfile);
fclose(outfile);
}

void display(){
FILE *outfile=fopen("Huy314.bin","rb");
int n=count();
for(int i=0;i<n;i++){
    fseek(outfile,sizeof(product)*i,SEEK_SET);
    fread(&ss[i],sizeof(product),1,outfile);
}
for(int i=0;i<n;i++){
    if(ss[i].b>2*ss[i].a){
        printf("%d %s",ss[i].nums,ss[i].name);
        printf(" %.2f %.2f",ss[i].a,ss[i].b);
    }
    printf("\n");
}
fclose(outfile);
}

int main(){
int n;
scanf("%d",&n);
switch (n){
    case 1:
        scanf("%d",&n);
        addnew(n);
        printf("%d",n);
        break;
    case 2:
        scanf("%d",&n);
        edit(n);
        printf("%d",n);
}

```

```

        break;
    case 3:
        display();
        break;
    default:
        printf("");
    }
    return 0;
}

```

quản lý thông tin sách (sắp xếp tăng dần theo giá sách)

Thông tin về sách gồm có: 1. mã sách là số nguyên tự động tăng; 2. tên sách; 3. tên tác giả; 4. giá tiền;

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới sách
2. cập nhật thông tin sách
3. hiển thị danh sách sách theo thứ tự tăng dần giá tiền

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện
- Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sách sẽ thêm mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sách sẽ sửa đổi
- Dòng tiếp theo là thông tin sản phẩm

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì sau khi thực hiện xong chức năng in ra số sách được thêm
- Nếu lựa chọn là 2 thì sau khi thực hiện xong chức năng in ra tên sách được sửa
- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây: giá lấy 2 chữ số thập phân, các thông tin cách nhau một khoảng trắng

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

hoa vang

nhat anh

12.5

hoa trang

nhat em

11.5

OUTPUT (1)

2

INPUT (2)

```

2
2
hoa do
nhat co
10.5
OUTPUT (2)
hoa do
INPUT (3)
3
OUTPUT
2 hoa do 10.50 nhat co
1 hoa vang 12.50 nhat anh
Giải
#include<stdio.h>
#include<string.h>
#define flush while(getchar()!='\n')

struct book{
    int nums;
    char name[50],author[50];
    float price;
}; book ss[100];

int count(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int c=0, temp =fgetc(outfile);
    while(temp =fgetc(outfile)!=EOF){
        c++;
        fseek(outfile,sizeof(book)*c,SEEK_SET);
    }
    fclose(outfile);
    return c;
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c=count();
    for(int i=1;i<=n;i++){
        book s;
        s.nums=i+c;
        flush;
        gets(s.name);
        gets(s.author);
        scanf("%f",&s.price);//scan o cuoi
}

```

```

        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

void edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
    book s;
    fseek(outfile,sizeof(s)*(n-1),SEEK_SET);
    s.nums=n;
    flush;
    gets(s.name);
    gets(s.author);
    scanf("%f",&s.price);
    fwrite(&s,sizeof(s),1,outfile);
    puts(s.name);
    fclose(outfile);
}

void sort(int n){
    n=count();
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if(ss[i].price>ss[j].price){
                book s=ss[i];
                ss[i]=ss[j];
                ss[j]=s;
            }
        }
    }
}

void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int n=count();
    for(int i=0;i<n;i++){
        fseek(outfile,sizeof(book)*i,SEEK_SET);
        fread(&ss[i],sizeof(book),1,outfile);
    }
    sort(n);
    for(int i=0;i<n;i++){
        book s;
        printf("%d ",ss[i].nums);
}

```

```

        printf("%s ",ss[i].name);
        printf("%.2f ",ss[i].price);
        printf("%s ",ss[i].author);
        printf("\n");//chu y "\n";
    }
    fclose(outfile);
}

int main(){
    int n;
    scanf("%d",&n);
    switch(n) {
        case 1:
            scanf("%d",&n);
            addnew(n);
            printf("%d",n);
            break;
        case 2:
            scanf("%d",&n);
            edit(n);
            break;
        case 3:
            display();
            break;
        default:
            printf("");
    }
    return 0;
}

```

quản lý thông tin thiết bị (sắp xếp giảm dần theo giá nhập)

Thông tin về thiết bị gồm có: 1. mã thiết bị là số nguyên tự động tăng; 2. tên thiết bị; 3. thông tin thiết bị gồm: giá nhập,số năm bảo hành.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
2. cập nhật thông tin theo mã thiết b
3. hiển thị danh sách thiết bị có giá nhập giảm dần

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện
- Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số thiết bị sẽ thêm mới

- + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã thiết bị cần sửa đổi
- Dòng tiếp theo là thông tin thiết bị

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì xuất ra số thiết bị đã được thêm
- Nếu lựa chọn là 2 thì in ra mã thiết bị đã được sửa
- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

tu lanh

16.5 2

may giat

8.5 4

OUTPUT (1)

2

INPUT (2)

2

2

dieu hoa

20.5 1

OUTPUT (2)

2

INPUT (3)

3

OUTPUT

2 dieu hoa 20.5 1

1 tu lanh 16.5 2

```
#include<stdio.h>
#include<string.h>
#define flush while(getchar()!='\n')
struct product{
    int nums,bh;
    char name[50];
```

```

float price;
}; product ss[100];

int count(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int c=0,temp=fgetc(outfile);
    while(temp!=fgetc(outfile)!=EOF){
        c++;
        fseek(outfile,sizeof(product)*c,SEEK_SET);
    }
    fclose(outfile);
    return c;//chu y phai tra ve gia tri c
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    product s;
    int c=count();
    for(int i=1;i<=n;i++){
        product s;
        s.nums=i+c;
        flush;//chu y co flush
        gets(s.name);
        scanf("%f %d",&s.price,&s.bh);
        fseek(outfile,sizeof(s)*(i-1),SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

void edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
    product s;
    fseek(outfile,sizeof(s)*(n-1),SEEK_SET);
    s.nums=n;
    flush;//chu y co flush
    gets(s.name);
    scanf("%f %d",&s.price,&s.bh);
}

```

```

fwrite(&s,sizeof(s),1,outfile);
fclose(outfile);
}

void sort(int n){
    n=count();
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if(ss[i].price<ss[j].price){
                product s=ss[i];
                ss[i]=ss[j];
                ss[j]=s;
            }
        }
    }
}

void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int n=count();
    for(int i=0;i<n;i++){
        fseek(outfile,sizeof(product)*i,SEEK_SET);
        fread(&ss[i],sizeof(product),1,outfile);
    }
    sort(n);
    for(int i=0;i<n;i++){
        printf("%d %s ",ss[i].nums,ss[i].name);
        printf("%.1f %d",ss[i].price,ss[i].bh);
        printf("\n");
    }
    fclose(outfile);
}

int main(){
    int n;
    scanf("%d",&n);
    switch(n) {
        case 1:

```

```

        scanf("%d",&n);
        addnew(n);
        printf("%d",n);
        break;
    case 2:
        scanf("%d",&n);
        edit(n);
        printf("%d",n);
        break;
    case 3:
        display();
        break;
    default:
        printf("");
    }
    return 0;
}

```

quản lý tiền cá nhân (thống kê kết quả)

Thông tin về tiền chi tiêu gồm có: 1. ID khoản mục chi tiêu là số nguyên tự động tăng; 2. loại khoản mục: 1 là thu và 2 là chi; 3. tên khoản mục; 4. số tiền

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới khoản mục
2. cập nhật thông tin khoản mục
3. hiển thị tổng thu, tổng chi và số tiền tiết kiệm được

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện
- Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập khoản mục chi tiêu mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập id khoản mục chi tiêu
- Dòng tiếp theo là thông tin khoản mục

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì sau khi thực hiện xong chức năng in ra số khoản mục thu và chi được thêm
- Nếu lựa chọn là 2 thì sau khi thực hiện xong chức năng in ra tên khoản mục được sửa
- Nếu lựa chọn là 3 thì in ra kết quả theo mô tả dưới đây: các thông tin cách nhau một khoảng trắng

Một trường hợp sử dụng chương trình

INPUT (1)

1
3
1
luong
10000

2
tien nha

5000

1

thuong
1000

OUTPUT (1)

2 1

INPUT (2)

2
3
tien an

2000

OUTPUT (2)

tien an

INPUT (3)

3

OUTPUT

10000 7000 3000

Thông tin về tiền chi tiêu gồm có: 1. ID khoản mục chi tiêu là số nguyên tự động tăng; 2. loại khoản mục: 1 là thu và 2 là chi; 3. tên khoản mục; 4. số tiền

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới khoản mục

2. cập nhật thông tin khoản mục

3. hiển thị tổng thu, tổng chi và số tiền tiết kiệm được

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện

- Dòng thứ hai

- + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập khoản mục chi tiêu mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập id khoản mục chi tiêu
- Dòng tiếp theo là thông tin khoản mục

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì sau khi thực hiện xong chức năng in ra số khoản mục thu và chi được thêm
- Nếu lựa chọn là 2 thì sau khi thực hiện xong chức năng in ra tên khoản mục được sửa
- Nếu lựa chọn là 3 thì in ra kết quả theo mô tả dưới đây: các thông tin cách nhau một khoảng trắng

Một trường hợp sử dụng chương trình

INPUT (1)

1
3
1
luong
10000
2
tien nha
5000
1
thuong
1000
OUTPUT (1)
2 1

INPUT (2)

2
3
tien an
2000
OUTPUT (2)
tien an

INPUT (3)

3
OUTPUT
10000 7000 3000

Giải

```
#include<stdio.h>
#include<string.h>
#define flush while(getchar()!='\n')
//onlai
struct money{
    int nums,option;
    char names[50];
    int cost;
}; money ss[100];

int count(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int c=0,temp=fgetc(outfile);
    while(temp!=fgetc(outfile)!=EOF){
        c++;
        fseek(outfile,sizeof(money)*c,SEEK_SET);/*c
    }
    fclose(outfile);
    return c;
}

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c=count(),count1=0,count2=0;
    for(int i=1;i<=n;i++){
        money s;
        s.nums=i+c;
        flush;
        scanf("%d",&s.option);
        if(s.option==1) count1++;
        else if(s.option==2) count2++;
        flush;
        gets(s.names);
        scanf("%d",&s.cost);
        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
}
```

```

printf("%d %d",count1,count2);
fclose(outfile);
}

void edit(int n){
    FILE *outfile=fopen("Huy314.bin","rb+");
    fseek(outfile,sizeof(money)*(n-1),SEEK_SET);//seek_set
    money s;
    s.nums=n;
    flush;
    s.option=2;
    gets(s.names);
    scanf("%d",&s.cost);
    fwrite(&s,sizeof(s),1,outfile);
    puts(s.names);
    fclose(outfile);

}

void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    int n=count(),thu=0,chi=0;
    for(int i=0;i<n;i++){
        fseek(outfile,sizeof(money)*i,SEEK_SET);
        fread(&ss[i],sizeof(money),1,outfile);
        if(ss[i].option==1) thu+=ss[i].cost;
        else if(ss[i].option==2) chi+=ss[i].cost;
    }
    printf("%d %d %d",thu ,chi, thu-chi );
    fclose(outfile);
}

int main (){
    int n;
    scanf("%d",&n);
    switch(n){
        case 1:
            scanf("%d",&n);

```

```

        addnew(n);
        break;
    case 2:
        scanf("%d",&n);
        edit(n);
        break;
    case 3:
        display();
        break;
    default :
        printf("");
    }
}

```

quản lý thông tin sinh viên (tìm kiếm theo mã sinh viên)

Thông tin về sinh viên gồm có: 1. Id sinh viên là số nguyên tự động tăng; 2. mã sinh viên; 3. tên sinh viên.

Viết chương trình cho phép ghi ra file nhị phân(yêu cầu file được đặt tên là mã sinh viên và tên file mở rộng .bin) thực hiện 3 chức năng:

1. thêm mới
2. cập nhật tên sinh viên

3. tìm kiếm sinh viên theo mã hoặc có mã bắt đầu bằng chuỗi nhập vào

Mô tả dữ liệu vào:

- Dòng thứ nhất là lựa chọn chức năng thực hiện
- Dòng thứ hai
 - + Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sinh viên sẽ thêm mới
 - + Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sinh viên sẽ sửa đổi
- Dòng tiếp theo là thông tin sinh viên

Mô tả dữ liệu ra:

- Nếu lựa chọn là 1 thì in ra số sinh viên được thêm
- Nếu lựa chọn là 2 thì in ra mã sinh viên được sửa
- Nếu lựa chọn là 3 thì in ra kết quả tìm kiếm theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

Một trường hợp sử dụng chương trình

INPUT (1)

1

2

b17dccn111

nguyen van a

b17dcvt112

nguyen van b

OUTPUT (1)

2

INPUT (2)

2

b17dccn111

b17dcvt113

tran van nam

OUTPUT (2)

b17dcvt113

INPUT (3)

3

b17dcvt

OUTPUT

1 tran van nam b17dcvt113

2 nguyen van b b17dcvt112

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#define flush while((getchar())!='\n')
```

```
struct sv{
```

```
    int nums;
```

```
    char name[50];
```

```
    char code[50];
```

```
};
```

```
int count(){
```

```
    FILE *outfile=fopen("Huy314.bin","rb");
```

```
    int c=0,temp=fgetc(outfile);
```

```
    while(temp==fgetc(outfile)!=EOF){
```

```
        c++;
```

```
        fseek(outfile, sizeof(sv)*c,SEEK_SET);/*c
```

```

        }
        fclose(outfile);
        return c;
    }

void addnew(int n){
    FILE *outfile=fopen("Huy314.bin","ab");
    int c=count();
    flush;//o day co vi ben trong khong co scanf so
    for(int i=1;i<=n;i++){//<=n
        sv s ;
        s.nums=i+c;
        // flush;
        gets(s.code);
        gets(s.name);
        //flush;
        fseek(outfile,0,SEEK_END);
        fwrite(&s,sizeof(s),1,outfile);
    }
    fclose(outfile);
}

void edit(){
    int n=count();
    FILE *outfile=fopen("Huy314.bin","rb+");
    char temp[50];
    flush;
    gets(temp);
    for(int i=0;i<n;i++){
        sv s;
        fseek(outfile,sizeof(sv)*i,SEEK_SET);
        fread(&s,sizeof(s),1,outfile);//fread
        if(strcmp(temp,s.code)==0){
            sv ss;
            ss.nums=i+1;//i+1
            // flush;
            gets(ss.code);
            // flush;
            gets(ss.name);
        }
    }
}

```

```

        fseek(outfile,sizeof(sv)*i,SEEK_SET);//fseek
        fwrite(&ss,sizeof(ss),1,outfile);//&ss
        puts(ss.code);
        break;
    }
}
fclose(outfile);
}

void display(){
    FILE *outfile=fopen("Huy314.bin","rb");
    char temp[50];
    flush;
    gets(temp);
    int l=strlen(temp);
    int n=count();
    for(int i=0;i<n;i++){
        sv s;
        fseek(outfile,sizeof(s)*i,SEEK_SET);
        fread(&s,sizeof(s),1,outfile);//doc truoc khi so sanh
        if(strncmp(temp,s.code,l)!=0 )
            continue;//node dong nay
        printf("%d %s %s",s.nums,s.name,s.code);
        printf("\n");
    }
    fclose(outfile);
}

int main(){
    int n;
    scanf("%d",&n);
    switch(n){
        case 1:
            scanf("%d",&n);
            addnew(n);
            printf("%d",n);
            break;
        case 2:
            edit();//khong co n
    }
}

```

```
        break;
    case 3:
        display();
        break;
    default:
        printf("");
    }
return 0;
}
```