

*Trong chương này, chúng ta sẽ tìm hiểu cấu trúc tập tin, cài đặt các thao tác, một số hàm thư viện và ứng dụng trong việc tổ chức dữ liệu trên tập tin.*

## I. TÓM TẮT LÝ THUYẾT

### I.1. Khái niệm

Trong các chương trình trước thì các dữ liệu đưa vào chương trình chỉ được tồn tại trong RAM, khi thoát chương trình thì tất cả dữ liệu đều bị mất. Để khắc phục tình trạng này Borland C cung cấp cho ta các hàm để lưu trữ và truy xuất tập tin, đó là kiểu **FILE**. Và ở đây ta chỉ đề cập đến 2 loại tập tin :

- Tập tin văn bản: là tập tin dùng để ghi các ký tự lên đĩa theo các dòng.
- Tập tin nhị phân: là tập tin dùng để ghi các cấu trúc dạng nhị phân (được mã hoá).

### I.2. Thao tác với tập tin

Quá trình thao tác trên tập tin thông qua 4 bước:

Bước 1: Khai báo con trỏ trỏ đến tập tin.

Bước 2: Mở tập tin.

Bước 3: Các xử lý trên tập tin.

Bước 4: Đóng tập tin.

#### a. Khai báo

**FILE \*< tên biến >;**

Ví dụ :      `FILE *f;      //      Khai báo biến con trỏ file f`

#### b. Mở tập tin

**fopen (< đường dẫn tên tập tin> , < kiểu truy nhập >);**

Ví dụ : `FILE *f;      //      Khai báo biến con trỏ f`  
`f = fopen ( "C:\\\\VD1.txt" , "rt" );`

**Các kiểu truy nhập tập tin thông dụng:**

- t** là kiểu truy nhập tập tin đối với dạng tập tin văn bản (text).
- b** là kiểu truy nhập tập tin đối với dạng tập tin nhị phân (binary).
- r** mở ra để đọc ( ready only).
- w** mở ra để ghi (create / write).
- a** mở ra để thêm vào (append).
- r+** mở ra để đọc và ghi (modify).

**c. Các hàm đọc ghi nội dung tập tin**

- Tập tin văn bản

STT	TÊN HÀM	Ý NGHĨA SỬ DỤNG	VÍ DỤ
<b>ĐỌC TẬP TIN</b>			
1	fscanf(<FILE *>, <định dạng>, <các tham biến>);	Dữ liệu từ một tập tin theo định dạng.	fscanf(f, "%d", &x);
2	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 s[80]; fgets(s, 80, f);
3	getc(<FILE *>);	Đọc một ký tự từ tập tin đang mở.	char c=getc(f);
<b>GHI TẬP TIN</b>			
1	fprintf(<FILE *>, <định dạng>[, <các tham biến>]);	Ghi dữ liệu theo một định dạng nào đó vào tập tin.	fprintf(f, "%d", x);
2	fputs(<chuỗi ký tự>, <FILE *>);	Ghi một chuỗi ký tự vào tập tin đang mở.	fputs("Giao trình BT", f);

- Tập tin nhị phân

STT	TÊN HÀM	Ý NGHĨA SỬ DỤNG	VÍ DỤ
<b>ĐỌC TẬP TIN</b>			
1	fread(<&ptr>, <size>, <len>, <FILE *>);	<ul style="list-style-type: none"> <li>ptr: vùng nhớ để lưu dữ liệu đọc.</li> <li>size: kích thước mỗi ô nhớ (tính bằng byte).</li> <li>len: độ dài dữ liệu cần đọc.</li> </ul> FILE: đọc từ tập tin nhị phân nào.	int a[30], b, n; fread(a, sizeof(int), n, f); Fread(&b, sizeof(int), 1, f);
<b>GHI TẬP TIN</b>			
1	fwrite(<&prt>, <size>, <len>, <FILE *>);	Tham số tương tự như hàm fread.	fwrite(a, sizeof(int), n, f);

**d. Đóng tập tin**

Sau khi không còn làm việc với tập tin, để đảm bảo an toàn cho dữ liệu thì nhất thiết ta phải đóng tập tin lại.

**fclose ( < biến con trỏ tập tin > );**  
hoặc **fcloseall ();**

*Ví dụ : fclose (f) ;*

**e. Các thao tác khác trên tập tin****\* Xoá tập tin :**

**remove ( < đường dẫn tập tin > );**

**\* Đổi tên tập tin :**

**rename ( < tên tập tin cũ >, < tên tập tin mới > );**

**\* Di chuyển con trỏ tập tin :**

**fseek ( < FILE \* >, < độ dời >, < mốc > );**

Các mốc :

**SEEK\_SET** dời đến đầu tập tin (giá trị 0).

**SEEK\_END** dời đến cuối tập tin (giá trị 2).

**SEEK\_CUR** dời vị trí hiện hành (giá trị 1).

*Ví dụ :*

*fseek (f, +5, SEEK\_CUR); // dời vị trí hiện hành về cuối 5 bytes*

*fseek (f, -4, SEEK\_CUR); // dời vị trí hiện hành về trước 4 bytes*

**\* Cho biết vị trí con trỏ file:**

**ftell ( < FILE \* > );**

*Ví dụ :*

*int size = ftell (f);*

*/\*size: khoảng cách từ đầu tập tin đến vị trí hiện hành (tính bằng byte)\*/*

**f. Ví dụ minh hoạ**

```
void KiemTra (FILE *f, char duongdan [])  
{  
    f = open (duongdan, "rt");  
    if (f == NULL)  
    {  
        printf ("Khong mo duoc tap tin %s", duongdan);  
        perror ("\nLy do");  
    }  
}
```

```

        getch ();
        return ;
    }
    printf ("Tập tin %s đã được mở", duongdan);
    fclose (f);
}

```

### I.3. Các ví dụ minh họa

#### a. Tập tin văn bản

Ví dụ 1: Viết chương trình tạo tập tin văn bản *SO.OUT* gồm  $n$  số nguyên, các số của dãy được tạo ngẫu nhiên có giá trị tuyệt đối không vượt quá  $M$  ( $n, M$  đọc từ tập tin *SO.INP*). Kết quả chương trình là 1 tập tin văn bản có dòng thứ nhất ghi số  $n$ ;  $n$  dòng tiếp theo ghi các số tạo được, mỗi số trên một dòng.

SO.INP
3
10

SO.OUT
3
5
7
2

```

#include < conio.h >
#include < stdio.h >

#define in "SO.INP"
#define out "SO.OUT"
int n, M ;

void Nhap ()
{
    FILE *fi;
    fi = fopen ( in , "rt" );
    fscanf ( fi, " %d %d ", &n, &M );
    fclose ( fi );
}

void Xuat ()
{
    FILE *fo;
    fo = fopen ( out , "wt " );
    fprintf ( fo , " %d\n", n );
    randomize ();
    for ( ; n > 0 ; n -- )
        fprintf ( fo , " %d\n", random ( ( 2 * M + 1 ) - M ) );
    fclose ( fo );
}

```

```

}
void main ()
{
    clrscr ();
    Nhap ();
    Xuat ();
}

```

Ví dụ 2: Viết chương trình phát sinh ngẫu nhiên ma trận  $a$  kích thước  $5 \times 6$ , lưu ma trận này vào file **test.inp**. Đọc lại file **test.inp** đưa dữ liệu vào ma trận  $b$  và xuất ra màn hình xem kết quả lưu đúng không? Cấu trúc của file **test.inp** như sau:

- Dòng đầu lưu 2 số nguyên:  $m, n$  thể hiện số dòng và số cột của ma trận.
- $m$  dòng tiếp theo, mỗi dòng gồm  $n$  phần tử là giá trị các phần tử trên một dòng của ma trận.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

#define MAX 100
#define dl "test.inp"

void LuuFile(int a[MAX][MAX], int m, int n)
{
    FILE *f;
    f=fopen(dl, "wt");
    if(f==NULL)
    {
        printf("\nKhong tao duoc file.");
        getch();
        exit(0);
    }
    fprintf(f, "%d %d\n", m, n);
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<n; j++)
            fprintf(f, "%d\t", a[i][j]);
        fprintf(f, "\n");
    }
    fclose(f);
}

void DocFile(int a[MAX][MAX], int &m, int &n)
{

```

```

    FILE *f;
    f=fopen(dl, "rt");
    if(f==NULL)
    {
        printf("\nKhong doc duoc file.");
        getch();
        exit(0);
    }
    fscanf(f, "%d%d", &m, &n);
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<n; j++)
            fscanf(f, "%d", &a[i][j]);
    }
    fclose(f);
}

void main()
{
    int a[MAX][MAX], m=5, n=6, i, j;
    int b[MAX][MAX], x, y;

    randomize();
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            a[i][j]=random(1000);
    LuuFile(a, m, n);
    DocFile(b, x, y);
    for(i=0; i<x; i++)
    {
        for(j=0; j<y; j++)
            printf("%d\t", b[i][j]);
        printf("\n");
    }
}

```

Kết quả ví dụ sau khi chạy chương trình, file test.inp có dạng sau:

5	6				
480	661	395	736	998	987
31	414	211	801	774	416
166	191	454	830	508	72
121	382	35	365	567	726
159	309	1	275	870	378

**b. Tập tin nhị phân**

Viết hàm đọc/ ghi một danh sách sinh viên của một lớp vào tập tin SV.DAT

```
SINHVIEN ds[100];
```

```

int siso;

void nhap ( )
{
    FILE *fi;
    fi = fopen ( "SV.DAT", "rb");
    fseek ( fi , 0 , SEEK_END );
    siso = ( ftell ( fi ) + 1 ) / sizeof ( SINHVIEN );
    fseek ( fi , 0 , SEEK_SET );
    fread ( ds , sizeof ( SINHVIEN ) , siso , fi );
    fclose ( fi );
}

void xuat ( )
{
    FILE *fo;
    fo = fopen ( "SV.DAT", "wb");
    fwrite ( ds , sizeof ( SINHVIEN ) , siso , fo );
    fclose ( fo );
}

```

## II. BÀI TẬP

### II.1. Bài tập cơ bản

- Viết chương trình tạo tập tin văn bản chứa 1 dãy số nguyên bất kỳ.
- Viết chương trình tạo tập tin nhị phân chứa 10000 số nguyên bất kỳ ghi vào file SONGUYEN.INP. Mỗi dòng 10 số, sau đó viết chương trình đọc file SONGUYEN.INP, sắp xếp theo thứ tự tăng dần và lưu kết quả vào file SONGUYEN.OUT.
- Viết chương trình tạo một file chứa 10000 số nguyên ngẫu nhiên đôi một khác nhau trong phạm vi từ 1 đến 32767 và đặt tên là "SONGUYEN.INP".
- Viết chương trình tạo một file chứa các số nguyên có tên SONGUYEN.INP. Sau đó đọc file SONGUYEN.INP và ghi các số chẵn vào file SOCHAN.OUT và những số lẻ vào file SOLE.OUT.
- Viết chương trình ghi vào tập tin SOCHAN.DAT các số nguyên chẵn từ 0 đến 100.
- Viết chương trình đọc tập tin SOCHAN.DAT và xuất ra màn hình, mỗi dòng 30 số.
- Viết chương trình giả lập lệnh **COPY CON** để tạo tập tin văn bản. Khi kết thúc tập tin nhấn phím **F6** để lưu.

8. Viết chương trình giả lập lệnh **TYPE** để in nội dung của tập tin văn bản ra màn hình.
9. Viết chương trình kiểm tra một tập tin nào đó có trong một thư mục được chỉ định hay không?
10. Viết chương trình giả lập lệnh **DEL** để xóa tập tin. Yêu cầu nhập đường dẫn và tên tập tin, kiểm tra sự tồn tại của tập tin, nếu có thì xóa tập tin được chỉ định.
11. Viết chương trình giả lập lệnh **RENAME** để đổi tên một tập tin.
12. Viết chương trình tạo file văn bản có tên là “MATRIX.INP” có cấu trúc như sau:
  - Dòng đầu ghi hai số  $m, n$ .
  - Trong  $m$  dòng tiếp theo mỗi dòng ghi  $n$  số và các số cách nhau một khoảng cách.Hãy kiểm tra xem trong file đó có bao nhiêu số nguyên tố.  
Kết quả cần ghi vào file “MATRIX.OUT” có nội dung là một số nguyên đó là số lượng các số nguyên tố trong file “MATRIX.INP”.
13. Cho số nguyên  $n$ , hãy in tam giác PASCAL gồm  $n$  dòng  
Dữ liệu vào: tập tin văn bản PAS.INP gồm 1 dòng chứa giá trị  $n$ .  
Kết quả: đưa ra tập tin văn bản PAS.OUT thể hiện một tam giác PASCAL  $n$  dòng.
14. Cho mảng các số nguyên, hãy sắp xếp mảng theo thứ tự tăng dần.  
Dữ liệu vào : tập tin văn bản ARRAY.INP gồm 2 dòng
  - Dòng 1 chứa số nguyên  $n$  ( $n \leq 100$ ).
  - Dòng 2 chứa  $n$  số nguyên.Kết quả : Đưa ra tập tin văn bản ARRAY.OUT gồm hai dòng
  - Dòng 1 chứa  $n$  phần tử của mảng các số nguyên.
  - Dòng 2 chứa  $n$  số nguyên được xếp tăng dần.
15. Cho mảng các số nguyên, tìm phần tử lớn nhất của mảng.  
Dữ liệu vào: tập tin văn bản ARRAY.INP gồm hai dòng:
  - Dòng 1 chứa số nguyên  $n$  ( $n \leq 100$ ).
  - Dòng 2 chứa  $n$  số nguyên.Kết quả: Đưa ra tập tin văn bản ARRAY.OUT gồm 1 dòng ghi 2 giá trị  $x, y$  trong đó  $x$  là giá trị lớn nhất,  $y$  là vị trí của  $x$  trong mảng.



## II.2. Bài tập luyện tập và nâng cao

16. Cho mảng các số nguyên, tính tổng các phần tử của mảng.

Dữ liệu vào : tập tin văn bản ARRAY.INP gồm hai dòng

- Dòng 1 chứa số nguyên  $n$  ( $n \leq 10$ )
- Dòng 2 chứa  $n$  số nguyên

Kết quả : Đưa ra tập tin văn bản ARRAY.OUT gồm một dòng ghi tổng các phần tử trong mảng.

17. Cho mảng các số nguyên, hãy liệt kê các phần tử là số nguyên tố

Dữ liệu vào : tập tin văn bản NT.INP gồm hai dòng

- Dòng 1 chứa số nguyên  $n$  ( $n \leq 100$ )
- Dòng 2 chứa  $n$  số nguyên

Kết quả : đưa ra tập tin văn bản NT.OUT gồm hai dòng:

- Dòng 1 chứa số lượng các phần tử nguyên tố trong mảng.
- Dòng 2 liệt kê các số nguyên tố đó.

18. (\*) Tạo file văn bản có tên là "INPUT.TXT" có cấu trúc như sau:

- Dòng đầu tiên ghi  $N$  ( $N$  là số nguyên dương nhập từ bàn phím).
- Trong các dòng tiếp theo ghi  $N$  số nguyên ngẫu nhiên trong phạm vi từ 0 đến 100, mỗi dòng 10 số (các số cách nhau ít nhất một khoảng trắng).

Hãy đọc dữ liệu của file "INPUT.TXT" và lưu vào mảng một chiều A.

Thực hiện các công việc sau :

- Tìm giá trị lớn nhất của mảng A.
- Đếm số lượng số chẵn, số lượng số lẻ của mảng A.
- Hãy sắp xếp các phần tử theo thứ tự tăng dần.

Hãy ghi các kết quả vào file văn bản có tên OUTPUT.TXT theo mẫu sau:

INPUT.TXT	OUTPUT.TXT
18	Cau a: 99
87 39 78 19 89 4 40 98 29	Cau b: 9 9
65	Cau c:
20 43 1 99 38 34 58 4	1 4 4 19 20 29 34 38 39 40
	43 58 65 78 87 89 98 99

19. (\*) Viết chương trình nhập và lưu hồ sơ của sinh viên vào một file có tên là "DSSV.TXT". Sau đó đọc file "DSSV.TXT" và cất vào mảng, hãy sắp xếp các hồ sơ sinh viên theo thứ tự giảm dần theo điểm trung bình môn

học rồi in ra màn hình hồ sơ các sinh viên theo thứ tự đó ra màn hình có thông tin như sau :

- Mã số sinh viên.
- Họ và tên sinh viên.
- Điểm trung bình kiểm tra.
- Điểm thi hết môn.
- Điểm trung bình môn học (tính bằng (điểm TBKT+điểm thi)/2).

20. (\*) Tạo một file text có tên là “INPUT.TXT” có cấu trúc như sau :

- Dòng đầu tiên ghi hai số  $M$  và  $N$  ( $M, N$  là hai số nguyên dương nhập từ bàn phím).
- Trong  $M$  dòng tiếp theo mỗi dòng ghi  $N$  số nguyên ngẫu nhiên trong phạm vi từ 0 đến 100 (các số này cách nhau ít nhất một khoảng trắng).

Hãy đọc dữ liệu từ file trên và lưu vào mảng hai chiều. Rồi thực hiện các công việc sau:

- Tìm giá trị lớn nhất của ma trận.
- Đếm số lượng số chẵn, lẻ, nguyên tố có trong ma trận.
- Hãy tính tổng các phần tử trên mỗi dòng của ma trận.

Hãy ghi kết quả này vào filetext có tên là “OUTPUT.TXT”

INPUT.TXT						OUTPUT.TXT					
6	6					Cau a: 49					
41	17	33	23	12	1	Cau b: 17 19					
44	24	23	49	5	24	Cau c: 127 169 147 214 132					
33	20	17	25	33	19	146					
0	48	45	48	41	32						
10	24	36	19	19	24						
30	4	23	26	27	36						

21. (\*\*) Xét dãy số  $a_1, a_2, \dots, a_N$ . Một đoạn con của dãy là dãy các phần tử liên tiếp nhau được xác định bởi chỉ số của số bắt đầu ( $L$ ) và chỉ số của số cuối cùng ( $R$ ). Tổng các số trên đoạn được gọi là tổng đoạn.

Yêu cầu : Cho dãy ( $a_N$ ), hãy tìm đoạn con có tổng đoạn lớn nhất ( $T$ )

Dữ liệu được cho trong tập tin văn bản SUMMAX.INP

- Dòng thứ nhất chứa số nguyên  $N$  ( $0 < N \leq 30000$ )

- $N$  dòng tiếp theo, mỗi dòng chứa một số là các số của dãy đã cho theo đúng thứ tự. Giá trị tuyệt đối của mỗi số không vượt quá 30000

Kết quả tìm được ghi vào tập tin văn bản SUMMAX.OUT gồm 1 dòng ghi 3 số T, L, R.

**Ví dụ :**

SUMMAX.INP	SUMMAX.OUT
5	8 2 5
-1	
5	
-3	
2	
4	

22. (\*) Cho dãy (  $a_N$  ), hãy tìm đoạn con tăng dần có tổng lớn nhất

Dữ liệu : được cho trong tập tin AMAX.INP

- Dòng 1 chứa số nguyên  $N$  (  $0 < N \leq 30000$  ).
- $N$  dòng tiếp theo, mỗi dòng chứa một số là các số của dãy đã cho theo đúng thứ tự. Giá trị tuyệt đối của mỗi số không vượt quá 30000.

Kết quả tìm được ghi vào tập tin văn bản AMAX.OUT gồm hai dòng:

- Dòng 1 ghi tổng của dãy con.
- Dòng 2 ghi mảng con tăng dần có tổng lớn nhất.

23. Viết chương trình nhập lý lịch một nhân viên vào danh sách các nhân viên.

Khi không nhập nữa bấm phím **Esc** và ghi vào tập tin NHANVIEN.DAT sau đó :

- Đọc từ tập tin NHANVIEN.DAT vừa tạo và in danh sách các nhân viên lên màn hình.
- Tìm và in lý lịch một nhân viên bằng các nhập và họ tên hoặc mã số nhân viên.

24. (\*\*) Để lắp ráp một máy vi tính hoàn chỉnh cần phải có tối thiểu 10 linh kiện loại A và có thể lắp bổ sung thêm vào khoảng tối đa 8 linh kiện loại B. Tại một cửa hàng vi tính cần quản lý bán hàng các loại linh kiện tại cửa hàng. Thông tin về một loại linh kiện gồm có: Tên linh kiện, quy cách , loại, đơn giá loại 1 ( chất lượng tốt – số nguyên), đơn giá loại 2 ( chất lượng thường – số nguyên ). Viết chương trình thực hiện những công việc sau :

- Nhập vào thông tin của các loại linh kiện có ở cửa hàng. Xuất danh sách các linh kiện đã nhập theo thứ tự tăng dần của loại linh kiện và tên linh kiện. Cho biết đã có đủ 10 linh kiện loại A cần thiết để lắp ráp máy tính hay chưa?
- Với giả định là cửa hàng đã có đủ 10 linh kiện loại A để lắp ráp máy. Nhập vào một số tiền để lắp ráp một máy tính. Có thể lắp được một máy tính hoàn chỉnh với các linh kiện toàn bộ theo đơn giá loại 1 hay đơn giá loại 2 hay không? Nếu số tiền trong khoảng giữa thì hãy tìm một phương án gồm những linh kiện theo đơn giá 1 và linh kiện theo đơn giá 2 để lắp?
- Tất cả dữ liệu phải lưu ở tập tin.

### III. KẾT LUẬN

- ❖ Mục đích của kiểu dữ liệu tập tin cho phép chúng ta lưu lại những thông tin cần thiết tương đối lớn: những dữ liệu đầu vào, những kết quả của chương trình hoặc những dữ liệu dùng để test chương trình, ...
- ❖ Khi thao tác trên tập tin phải thông qua 4 bước: ***Khai báo con trỏ trỏ đến tập tin, Mở tập tin, Xử lý trên tập tin và cuối cùng là Đóng tập tin.***
- ❖ Lưu ý khi mở tập tin để ghi thì phải cẩn thận với thao tác tạo mới hay chỉnh sửa nội dung tập tin, di chuyển con trỏ hợp lý để tránh mất thông tin.
- ❖ Sử dụng hàm thao tác trên tập tin phải dùng **đúng loại hàm** cho tập tin kiểu nhị phân hay kiểu văn bản.