



HUTECH
Đại học Công nghệ Tp.HCM

KHOA CÔNG NGHỆ THÔNG TIN

Bài giảng:

KỸ THUẬT LẬP TRÌNH

Bài 6:

TẬP TIN - FILE



C Ngôn ngữ lập trình số 1 thế giới

Giảng viên: Th.S Dương Thành Phết

Email: phetcm@gmail.com

Website: <http://www.thayphet.net>

Mobile: 0918158670



MỤC TIÊU

- ✓ Trình bày được các khái niệm về tập tin;
- ✓ Mô tả được các bước thao tác với tập tin;
- ✓ Sử dụng được một số hàm truy xuất đến tập tin văn bản;
- ✓ Sử dụng được một số hàm truy xuất đến tập tin nhị phân.



NỘI DUNG

1. Khái niệm về tập tin.
2. Các thao tác trên tập tin
3. Truy cập tập tin văn bản
4. Truy cập tập tin nhị phân
5. Bài tập.



6.1. KHÁI NIỆM VỀ TẬP TIN

- ✓ Các kiểu dữ liệu như: kiểu số, kiểu mảng, kiểu cấu trúc thì dữ liệu được tổ chức trong bộ nhớ trong (RAM) khi kết thúc việc thực hiện chương trình thì dữ liệu bị mất;
- ✓ Điều đó vừa mất thời gian vừa không giải quyết được các bài toán với số liệu lớn cần lưu trữ.
- ✓ Để giải quyết vấn đề, người ta đưa ra kiểu tập tin (file) cho phép lưu trữ dữ liệu ở bộ nhớ ngoài (đĩa).
- ✓ Khi kết thúc chương trình thì dữ liệu vẫn còn để sử dụng nhiều lần.
- ✓ Kiểu tập tin là kích thước lớn với số lượng các phần tử không hạn chế
- ✓ Có 3 loại dữ liệu kiểu tập tin:



6.1. KHÁI NIỆM VỀ TẬP TIN

6.1.1. Tập tin văn bản (Text File):

- ✓ Là loại tập tin dùng để ghi các ký tự, các ký tự này được lưu trữ dưới dạng mã Ascii.
- ✓ 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'; ký tự này là sự kết hợp của 2 ký tự CR (Carriage Return - Về đầu dòng, mã Ascii là 13) và LF (Line Feed - Xuống dòng, mã Ascii là 10).
- ✓ 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).
- ✓ Tập tin văn bản chỉ có thể truy xuất theo kiểu tuần tự.



6.1. KHÁI NIỆM VỀ TẬP TIN

6.1.2. Tập tin định kiểu (Type File):

- ✓ Là loại tập tin bao gồm nhiều phần tử có cùng kiểu: char, int, long, cấu trúc...
- ✓ Được lưu trữ trên đĩa dưới dạng một chuỗi các byte liên tục.



6.1. KHÁI NIỆM VỀ TẬP TIN

6.1.3. Tập tin không định kiểu (Untype File):

- ✓ Là loại tập tin mà dữ liệu gồm các CTDL, không quan tâm đến nội dung hoặc kiểu, chỉ lưu ý đến các yếu tố vật lý của tập tin: độ lớn và các yếu tố tác động lên tập tin.
- ✓ Biến tập tin: Là một biến thuộc kiểu dữ liệu tập tin dùng để đại diện cho một tập tin.
- ✓ Dữ liệu chứa trong 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 tập tin được mở ra để làm việc, tại mỗi thời điểm, sẽ có 1 vị trí của tập tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra. Con trỏ đang chỉ đến vị trí đó.
- ✓ 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).



6.2. CÁC THAO TÁC TRÊN TẬP TIN

Thao tác trên tập tin, theo các bước:

- ✓ 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()`.

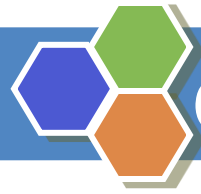
Thao tác với tập tin nhờ các hàm được định nghĩa trong thư viện **stdio.h**



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.1. Khai báo biến tập tin

- ✓ Cú pháp: **FILE** <Danh sách các biến con trỏ>
- ✓ Các biến trong danh sách phải là các con trỏ và được phân cách bởi dấu phẩy(,).
- ✓ Ví dụ: **FILE *f1,*f2;**



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.2. Mở tập tin

- ✓ Cú pháp: **FILE *fopen(char *Path, const char *Mode)**
- ✓ Trong đó:
 - Path: chuỗi chỉ đường dẫn đến tập tin trên đĩa.
 - Type: 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



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.2. Mở tập tin

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

```
FILE *f;
f = fopen(" c:\\TEST.txt", "w");
if (f!=NULL)
{
    /* Các câu lệnh để thao tác với tập tin*/
    /* Đóng tập tin*/
}
```

- ✓ Trong ví dụ trên, có sử dụng câu lệnh kiểm tra điều kiện để xác định mở tập tin có thành công hay không?
- ✓ Khi mở tập tin để ghi (*chế độ w*), nếu tập tin đã tồn tại rồi thì nội dung của tập tin sẽ bị xóa và một tập tin mới được tạo ra.
- ✓ Nếu muốn ghi nối dữ liệu, phải sử dụng chế độ "a".
- ✓ Khi mở với chế độ đọc, tập tin phải tồn tại, nếu không lỗi sẽ xuất hiện



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.3. Đóng tập tin

- ✓ Hàm **fclose()** dùng để đóng tập tin được mở bởi **fopen()**
 - Hàm 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 là đóng tập tin thành công.
 - Hàm trả về EOF nếu có xuất hiện lỗi.
- ✓ Hàm **fcloseall()** để đóng tất cả các tập tin lại.
 - Cú pháp: **int fcloseall()**
 - Kết quả trả về là tổng số các tập tin được đóng lại.
 - Nếu không thành công, kết quả trả về là EOF.



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.4. 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
- ✓ Trả về EOF nếu cuối tập tin được chạm tới, ngược lại trả về 0.



6.2. CÁC THAO TÁC TRÊN TẬP TIN

6.2.4. Di chuyển con trỏ tập tin về đầu tập tin

✓ Hàm `rewind()`

- Cú pháp: `void rewind (FILE *f)`
- Khi ta đang thao tác một tập tin đang mở, con trỏ tập tin luôn di chuyển về phía cuối tập tin.
- Muốn cho con trỏ quay về đầu tập tin như khi mở nó, ta sử dụng hàm `rewind()`.



6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.1. Ghi dữ liệu lên tập tin văn bản

✓ Hàm putc()

- Dùng để ghi một ký tự lên một tập tin văn bản đang được mở để làm việc.
- Cú pháp: `int putc (int c, FILE *f)`
- Trong đó: c là mã Ascii của 1 ký tự, được ghi lên tập tin liên kết với con trỏ f.
- Hàm này trả về EOF nếu gặp lỗi.



6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.1. Ghi dữ liệu lên tập tin văn bản

✓ Hàm fputs()

- Dùng để ghi một chuỗi ký tự chứa trong vùng đệm lên tập tin văn bản.
- Cú pháp: `int puts (const char *buffer, FILE *f)`
- Trong đó, buffer là con trỏ có kiểu char chỉ đến vị trí đầu tiên của chuỗi ký được ghi vào.
- Hàm này trả về giá trị 0 nếu buffer chứa chuỗi rỗng và trả về EOF nếu gặp lỗi.

6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.1. Ghi dữ liệu lên tập tin văn bản

✓ Hàm fprintf()

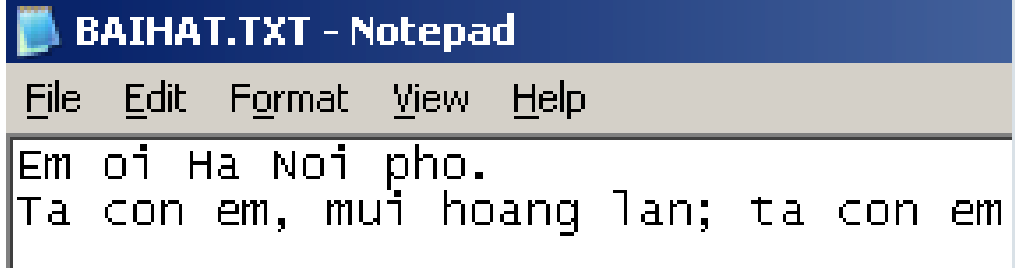
- Dùng để ghi dữ liệu có định dạng lên tập tin văn bản.
- Cú pháp: **fprintf (FILE *f, const char *format, varexpr)**
- Trong đó:
 - format: chuỗi định dạng (như hàm printf()).
 - varexpr: DS biểu thức, cách nhau dấu phẩy (,).

Định dạng	Ý nghĩa
%d	Ghi số nguyên
%[.số chữ số thập phân] f	Ghi số thực có <số chữ số thập phân> theo quy tắc làm tròn số.
%o	Ghi số nguyên hệ bát phân
%X	Ghi số nguyên hệ thập lục phân
%c	Ghi một ký tự
%s	Ghi chuỗi ký tự
%e hoặc %E hoặc %g hoặc %G	Ghi số thực dạng khoa học (nhân 10 mũ x)

6.3. TRUY CẬP TẬP TIN VĂN BẢN

Ví dụ: Chương trình ghi chuỗi ký tự lên tập tin D:\\Baihat.txt

```
#include<stdio.h>
#include<conio.h>
int main ()
{
    FILE *f;
    f = fopen ("D:\\Baihat.txt","r+");
    if (f!=NULL)
    {
        fputs("Em oi Ha Noi pho.\n",f);
        fputs("Ta con em, mui hoang lan; ta con em",f);
        fclose(f);
    }
    getch();
}
```



Nội dung tập tin Baihat.txt được mở bằng Notepad.

6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.2. Đọc dữ liệu từ tập tin văn bản

✓ Hàm `getc()`

- Dùng để đọc dữ liệu từ tập tin V.Bản đang được mở.
- Cú pháp: `int getc (FILE *f)`
- Hàm này trả về mã Ascii của một ký tự nào đó (kể cả EOF) trong tập tin liên kết với con trỏ f.



6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.2. Đọc dữ liệu từ tập tin văn bản

✓ Hàm fgetc()

- Cú pháp: `char *fgetc (char *buffer, int n, FILE *f)`
- Dùng để đọc một chuỗi ký tự từ tập tin văn bản đang mở và liên kết với con trỏ f cho đến khi đọc đủ n ký tự hoặc gặp ký tự xuống dòng '\n' hay ký tự kết thúc EOF
- Trong đó:
 - buffer (vùng đệm): Con trỏ có kiểu char chỉ đến vùng nhớ đủ lớn chứa các ký tự nhận được.
 - n: Độ dài lớn nhất của chuỗi ký tự nhận được.
 - f: Con trỏ liên kết với một tập tin nào đó.
- Hàm trả về địa chỉ đầu tiên của vùng đệm khi không gặp lỗi và chưa gặp ký tự kết thúc EOF.
- Ngược lại, hàm trả về giá trị NULL.

6.3. TRUY CẬP TẬP TIN VĂN BẢN

6.3.2. Đọc dữ liệu từ tập tin văn bản

✓ Hàm fscanf()

- Dùng để đọc dữ liệu từ tập tin văn bản vào danh sách các biến theo định dạng.
- Cú pháp: `fscanf (FILE *f, const char *format, varlist)`
- Trong đó:
 - format: chuỗi định dạng (giống hàm scanf());
 - varlist: DS các biến, cách nhau dấu phẩy (,).



6.3. TRUY CẬP TẬP TIN VĂN BẢN

Ví dụ: Viết chương trình chép tập tin D:\Baihat.txt ở trên sang tập tin D:\Baica.txt.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *f1,*f2 ;
    f1=fopen("D:\\Baihat.txt","rt");
    f2=fopen("D:\\Baica.txt","wt");
    if (f1!=NULL && f2!=NULL){
        int ch=fgetc (f1);
        while (! feof (f1)){
            fputc(ch,f2);
            ch=fgetc(f1);
        }
        fcloseall();
    }
    getch();
}
```



6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

6.4.1. Ghi dữ liệu lên tập tin nhị phân - Hàm fwrite()

Cú pháp: `size_t fwrite(const void *ptr, size_t size, size_t n, FILE*f)`

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ chứa thông tin cần ghi lên tập tin.
- n: số phần tử sẽ ghi lên tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.

Giá trị trả về của hàm là số phần tử được ghi lên tập tin. Giá trị này bằng n trừ khi xuất hiện lỗi.



6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

6.4.2. Đọc dữ liệu từ tập tin nhị phân - Hàm fread()

Cú pháp: `size_t fread (const void *ptr, size_t size, size_t n, FILE *f)`

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ sẽ nhận dữ liệu từ tập tin.
- n: số phần tử được đọc từ tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.
- ✓ Giá trị trả về của hàm này là số phần tử đã đọc được từ tập tin.
- ✓ Giá trị này $\leq n$ nếu chạm đến cuối tập tin hoặc có lỗi xuất hiện.



6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

6.4.3. Di chuyển con trỏ tập tin - Hàm fseek()

Cú pháp: `int fseek (FILE *f, long offset, int whence)`

Trong đó:

- Fseek: di chuyển con trỏ f đến vị trí offset theo mốc whence
- f : con trỏ tập tin đang thao tác.
- offset: số byte cần dịch chuyển con trỏ tập tin kể từ vị trí trước đó. Phần tử đầu tiên là vị trí 0.
- whence: vị trí bắt đầu để tính offset, ta có thể chọn điểm xuất phát là:

`#define SEEK_SET 0 // tính từ đầu tập tin`

`#define SEEK_CUR 1 // tính từ vị trí hiện hành của con trỏ`

`#define SEEK_END 2 // tính từ cuối tập tin`

fseek trả về: 0 nếu thành công, $\neq 0$ nếu di chuyển có lỗi



6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

Ví dụ 1: Viết chương trình ghi lên tập tin CacSo.Dat, 3 giá trị số (thực, nguyên, nguyên dài). Đọc các số từ tập tin vừa ghi hiển thị lên màn hình

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *f;
    clrscr();
    f=fopen ("D:\\CacSo.txt","wb");
    if (f!=NULL)
    {
        double d=3.14;
        int i=101;
        long l=54321;
        fwrite (&d,sizeof(double),1,f);
        fwrite(&i,sizeof(int),1,f);
        fwrite(&l,sizeof(long),1,f);
```

```
        /* Doc tu tap tin*/
        rewind(f);
        fread (&d,sizeof(double),1,f);
        fread(&i,sizeof(int),1,f);
        fread(&l,sizeof(long),1,f);
        printf("Cac ket qua la:
                %f %d   %ld",d,i,l);
        fclose(f);
    }
    getch();
}
```



6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

Ví dụ 2: Mỗi sinh viên có 2 thông tin: mã SV và họ tên. Viết chương trình cho phép lựa chọn các chức năng:

- ✓ Nhập DSSV từ bàn phím rồi ghi lên tập tin SinhVien.dat,
- ✓ Đọc dữ liệu từ tập tin SinhVien.dat rồi hiển thị lên màn hình,
- ✓ Tìm kiếm họ tên của 1 SV dựa vào mã sinh viên nhập từ bàn phím

Cần khai báo cấu trúc Sinhvien:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
typedef struct
{
    char Ma[10];
    char HoTen[40];
} SinhVien;
```

6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

```
void WriteFile (char *FileName){
    FILE *f;
    int n,i;
    SinhVien sv;
    f=fopen(FileName,"ab");
    printf("Nhap so luong SV:");
    scanf("%d",&n);
    fflush(stdin);
    for(i=1;i<=n;i++) {
        printf("S.Vien thu %i\n",i);
        printf("    - MSSV: ");
        gets(sv.Ma);
        printf("    - Ho ten: ");
        gets(sv.HoTen);
        fwrite(&sv,sizeof(sv),1,f);
        fflush(stdin);
    }
    fclose(f);
    printf("Phim bat ky de tiep tục");
    getch();
}
```

```
void ReadFile(char *FileName)
{
    FILE *f;
    SinhVien sv;
    f=fopen(FileName,"rb");
    printf("MSSV   |           Ho ten\n");
    fread (&sv,sizeof(sv),1,f);
    while (!feof(f))
    {
        printf("    %s |%s\n",
               sv.Ma,sv.HoTen);
        fread(&sv,sizeof(sv),1,f);
    }
    fclose(f);
    printf("Phim bat ky de tiep tục!!!");
    getch();
}
```

6.4. TRUY CẬP TẬP TIN NHỊ PHÂN

```
void Search(char *FileName){
    char MSSV[10] ; FILE *f ;
    int Found=0;    SinhVien sv;
    fflush(stdin);
    printf("Ma so SV can tim: ");
    gets(MSSV);
    f=fopen(FileName,"rb");
    while (!feof(f) && Found==0){
        fread(&sv,sizeof(sv),1,f);
        if (strcmp(sv.Ma,MSSV)==0)
            Found=1;
    }
    fclose(f);
    if (Found == 1)
        printf("Tim thay SV co ma
                %s. Ho ten la: %s",
                sv.Ma,sv.HoTen);
    else
        printf("Tim khong thay");
    printf("\nPhim bat ky de tiep tuc!!!");
    getch();
}
```

```
void main()
{
    int c;
    for (;;)
    {
        printf("1. Nhap DSSV\n");
        printf("2. In DSSV\n");
        printf("3. Tim kiem\n");
        printf("4. Thoat\n");
        printf("Ban chon 1, 2, 3, 4: ");
        scanf("%d",&c);
        if(c==1)
            WriteFile("d:\\SinhVien.Dat");
        else if (c==2)
            ReadFile("d:\\SinhVien.Dat");
        else if (c==3)
            Search("d:\\SinhVien.Dat");
        else
            break;
    }
}
```



6.5 BÀI TẬP THAO TÁC TRÊN TẬP TIN

1. Viết chương trình quản lý một tập tin văn bản yêu cầu:
 - ✓ Nhập từ bàn phím nội dung một văn bản sau đó ghi vào đĩa
 - ✓ Đọc từ đĩa nội dung văn bản vừa nhập và in lên màn hình.
 - ✓ Đọc từ đĩa nội dung văn bản vừa nhập, in nội dung đó lên màn hình và cho phép nối thêm thông tin vào cuối tập tin.
2. Viết chương trình cho phép thống kê số lần xuất hiện của các ký tự ('A'..'Z', 'a'..'z') trong một tập tin văn bản.
3. Viết chương trình đếm số từ và số dòng trong một tập tin văn bản.
4. Viết chương trình nhập từ bàn phím và ghi vào 1 tập tin tên là DMHH.TXT với mỗi phần tử của tập tin là 1 cấu trúc bao gồm các trường:
 - ✓ Ma (mã hàng: char[5]).
 - ✓ Ten (Tên hàng: char[20]).
 - ✓ Kết thúc việc nhập bằng cách gõ ENTER vào Ma.



6.5 BÀI TẬP THAO TÁC TRÊN TẬP TIN

5. Viết chương trình cho phép nhập từ bàn phím và ghi vào 1 tập tin tên DSHH.TXT với mỗi phần tử của tập tin là một cấu trúc bao gồm các trường:

- mh (mã hàng: char[5]).
- sl (số lượng: int).
- dg (đơn giá: float).
- st (Số tiền: float).

Theo yêu cầu:

- Mỗi lần nhập một cấu trúc
- Trước tiên nhập mã hàng (mh), so mh với Ma trong tập tin DMHH.TXT đã được tạo, nếu mh=ma thì in tên hàng ngay bên cạnh mã hàng.
- Nhập số lượng (sl).
- Nhập đơn giá (dg).
- Tính số tiền = số lượng * đơn giá.

Kết thúc nhập bằng ENTER vào mã hàng. Sau khi nhập xong, in toàn bộ danh sách hàng hóa có sự giải mã về tên hàng.



6.5 BÀI TẬP THAO TÁC TRÊN TẬP TIN

6. Viết chương trình

- Đọc một chuỗi tối đa 100 kí tự từ bàn phím.
- Lưu các ký tự là nguyên âm vào tập tin “NguyenAm.txt”.
- Đọc các kí tự từ tập tin này và hiển thị lên màn hình.

7. Cho file TXT có cấu trúc như sau:

- Dòng đầu lưu giá trị của 1 số nguyên dương n
- Dòng còn lại lưu giá trị của 1 dãy n các số nguyên

Viết chương trình đọc file trên, lưu vào mảng 1 chiều

- Xuất ra màn hình
- Tính tổng các số chẵn có trong mảng
- Tìm phần tử lớn nhất
- Đếm số phần tử lớn nhất
- Sắp xếp mảng tăng dần



6.5 BÀI TẬP THAO TÁC TRÊN TẬP TIN

8. Cho file TXT có cấu trúc như sau:

- ✓ Dòng đầu lưu giá trị của 2 số nguyên dương m, n
- ✓ Các dòng còn lại lưu giá trị của 1 ma trận m dòng và n cột là các số nguyên.

Viết chương trình đọc file ma trận trên

- a. Xuất ma trận đó ra màn hình
- b. Tính tổng các số chẵn có trong ma trận
- c. Tìm phần tử lớn nhất có trong ma trận
- d. Đếm số phần tử lớn nhất có trong ma trận
- e. Sắp xếp mảng tăng dần



HUTECH
Đại học Công nghệ Tp.HCM

KHOA CÔNG NGHỆ THÔNG TIN

Bài giảng:

KỸ THUẬT LẬP TRÌNH

HẾT BÀI 6 TẬP TIN - FILE



C Ngôn ngữ lập trình số 1 thế giới