



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# LẬP TRÌNH C CƠ BẢN

Kiểu dữ liệu cơ bản, vào ra file

# File văn bản và File nhị phân

- ❖ File văn bản:
  - Có nội dung là văn bản chứa các ký tự nhìn thấy được.
- ❖ File nhị phân
  - Chứa biểu diễn của dữ liệu

Ví dụ: `int x = 64; // 4 bytes`

Dạng văn bản: `input.txt - 64`

Dạng nhị phân: `input.dat - 00000000 00000000 00000000 01000000`  
(big edian vs little edian)

Tập văn bản:

**Khái niệm dòng:** dấu hiệu kết thúc dòng:

Windows cặp 2 ký tự CR và LF/ “\r\n”

Unix/Linux chỉ sử dụng 1 ký tự LF “\n”

‘\n’ - LF

`fgets` - nội dung văn bản nhìn thấy giống nhau, nhưng độ dài (`strlen` - có thể khác nhau).

`s[strlen(s) - 1] = '\0'; // Xóa LF`

`s[strlen(s) - 2] = '\0'; // Xóa CR`

# Xử lý tệp

---

- ❖ Mở tệp - fopen
- ❖ ...
- ❖ Đóng tệp - fclose

# Ví dụ 1.3. Tính tổng dãy số

```
vd1-3.c x
1  #include <stdio.h>
2
3  int main(int argc, char *argv[]) {
4      if (argc != 2) {
5          printf("Usage: ./prog input.txt\n");
6          return 1;
7      }
8      FILE *inp = fopen(argv[1], "r");
9      int n;
10     fscanf(inp, "%d", &n);
11     int sum = 0, x;
12     for (int i = 0; i < n; ++i) {
13         fscanf(inp, "%d", &x);
14         sum += x;
15     }
16     printf("Tổng các số là: %d\n", sum);
17     fclose(inp);
18     return 0;
19 }
20
```

```
bangoc: ~/git/c-basic
File Edit View Search Terminal Help
bangoc:$gcc -o prog vd1-3.c
bangoc:$./prog
Usage: ./prog input.txt
bangoc:$cat vd1-3-inp.txt
5
1 2 3 4 ./prog vd1-3-inp.txt
Tổng các số là: 15
bangoc:$
```

# Bài tập 1.12. Sao chép nội dung tệp văn bản

---

- Tạo một file văn bản với tên lab1.txt với nội dung bất kỳ, lưu trong thư mục cùng với chương trình.
- Viết chương trình đọc từ file trên mỗi lần một ký tự, sau đó ghi vào một file mới với tên lab1w.txt

*Đọc ghi tệp văn bản bằng fgetc và fputc*

```
if( !feof(inp) ) {...}
```

## Bài tập 1.13. Thao tác với tập tin theo từng ký tự

- Viết chương trình đọc nội dung từ một tập tin văn bản, mỗi lần đọc một ký tự.
- Chương trình sẽ chuyển ký tự chữ cái hoa thành ký tự chữ cái thường và ngược lại, sau đó ghi vào một tập tin khác.
- Chú ý với các ký tự khác – chương trình vẫn thực hiện sao chép một cách thông thường sang tập tin mới.

ctype.h

toupper

tolower

isalpha

# Bài tập 1.14

---

- Điều chỉnh bài tập 1.12 sử dụng các tham số dòng lệnh:
  - mycp <tập\_tin\_1> <tập\_tin\_2>
- Đường dẫn, tên các tập tin được cung cấp dưới dạng đối số dòng lệnh.
- *Chú ý: Chương trình phải kiểm tra cú pháp sử dụng (vd số đối số – thông báo lỗi và hiển thị hướng dẫn khi cần..)*

# Bài tập 1.15. Chương trình nối tập tin

- ❖ Viết chương trình nhận tên hai tệp văn bản ở đối số dòng lệnh, sau đó tiến hành ghép nội dung của tập tin thứ hai vào cuối tập tin thứ nhất. Giả sử cả hai tập tin đều tồn tại.
- ❖ Cú pháp sử dụng:
  - `apd <file1> <file2>`
- ❖ *Chú ý: Chương trình phải kiểm tra cú pháp sử dụng (vd số đối số – thông báo lỗi và hiển thị hướng dẫn khi cần..)*



# Bài tập 1.16. Đổi sang chữ hoa

- ❖ Viết chương trình có tên uconvert có chức năng chuyển đổi tất cả các chữ cái trong nội dung một tập tin (được cung cấp trong đối số dòng lệnh) thành chữ hoa và ghi lại nội dung mới vào một tệp khác.
- ❖ Cú pháp: uconvert tata.txt out.txt
- ❖ Ví dụ :
  - File nguồn tata.txt: helloworld
  - Nội dung out.txt: HELLOWORD.

# Bài tập 1.17.

- ❖ Viết chương trình `double_line` nhận một tập tin đầu vào và biến đổi nội dung của nó như sau: chương trình sẽ chèn thêm 1 dòng trống giữa các dòng trong văn bản. Kết quả được ghi vào một tập tin ở đầu ra. Chương trình chạy dưới dạng đối số dòng lệnh như sau: `./double_line <File 1> <File 2>`
- ❖ Ví dụ minh họa nội dung hai file khi chạy chương trình

Welcome to  
C Programming!



Welcome to  
  
C Programming!

# Bài tập 1.18. Mã hóa Caesar

- ❖ Viết một chương trình có thể sử dụng cùng một lúc hai chức năng mã hóa và giải mã một tập tin văn bản sử dụng mật mã Caesar (mã hóa cộng) như sau. Chương trình nhận ba đối số:
  - <tập tin nguồn> <độ dịch chuyển> < tập tin đích>
- ❖ Khi cần mã hóa, chạy chương trình với độ dịch chuyển (offset) n là một số nguyên dương. Chương trình sẽ thay thế mỗi ký tự trong tập tin bởi một ký tự đứng sau nó n vị trí trong bảng mã ASCII, dịch chuyển theo quy tắc xoay vòng. Ví dụ với offset =3 thì A → D, B → E
- ❖ Khi giải mã, chạy chương trình với đầu vào là tập tin mã hóa và giá trị độ dịch chuyển là số âm tương ứng (VD offset = -3)

# Đọc tệp văn bản theo dòng

fgets/fputs - Đọc ghi theo dòng

fgetc/fputc - Đọc ghi theo ký tự

fgetc - fscanf( ..., "%c")

fgets != fscanf( ..., "%s") ;

// tương tự scanf("%[^\\n]\*c) -> fscanf(

scanf - đọc từ stdin (con trỏ FILE \*)/ fscanf(stdin, ....)

printf - viết ra stdout (con trỏ FILE \*)/ fprintf(stdout, ....)

sscanf - đọc từ chuỗi (mảng ký tự/bộ nhớ đệm)

sprintf - viết ra chuỗi (mảng ký tự/bộ nhớ đệm)

# Bài tập 1.19

---

- Thực hiện lại bài tập 1.12, tuy nhiên thay vì sử dụng cặp hàm fgetc và fputc – ta sử dụng cặp hàm fgets và fputs.

# Bài tập 1.20. Đếm dòng & hiển thị nội dung

---

- Điều chỉnh bài tập 1.19 để chương trình chỉ hiển thị nội dung tập tin ra màn hình, sau đó hiển thị số các dòng văn bản.
- Minh họa về giao diện của chương trình:
  - Reading file Haiku.txt.... done!
  - Haiku haiku
  - Tokyo
  - Hanoi
  - This file has 3 lines.

# Bài tập 1.21. Hiển thị số thứ tự dòng trong tập tin văn bản

---

- ❖ Viết chương trình đọc một tập tin văn bản và hiển thị ra màn hình số hiệu từng dòng, theo sau là nội dung của dòng đó. Tên tập tin được cung cấp dưới dạng đối số dòng lệnh.
- ❖ Ví dụ với tập tin có nội dung  
This is sample file.  
Hello!
- ❖ Kết quả trên màn hình là:  
1 This is sample file.  
2 Hello!

# Bài tập 1.22. Trộn hai tập tin theo dòng

- ❖ Viết chương trình có đối số dòng lệnh như sau
  - `merge <file1> <file2> <file3>`
- ❖ Chương trình ghi vào file 3 bằng cách đọc và trộn lần lượt từng dòng từ file 1 và file 2. Chương trình đọc một dòng từ file 1, ghi vào file 3 sau đó đọc một dòng từ file 2 – ghi vào file 3. Chú ý: File 1 và File 2 có thể có số dòng khác nhau, khi đọc hết nội dung một file, chương trình sao chép các dòng tiếp theo của file còn lại vào file 3.



# Bài tập 1.23.

---

- Viết chương trình mycat đọc và hiển thị nội dung một tập tin văn bản trên màn hình. Chương trình hỗ trợ hai cú pháp sử dụng như sau:
- mycat <filename> : Hiển thị một lần toàn bộ nội dung
- mycat <filename> -p : Hiển thị theo từng trang, mỗi trang 10 dòng. Người dùng nhấn Enter để xem trang tiếp theo.

# Bài tập 1.24

---

- Viết chương trình nhận đối số dòng lệnh là đường dẫn đến một file văn bản (nội dung dưới 80 dòng). Chương trình thêm một dòng mới vào cuối file nói trên với nội dung chứa các ký tự đầu tiên của các dòng trong file ban đầu.

# Bài tập 1.25

- ❖ Viết chương trình đọc từng dòng văn bản từ một tập tin, sau đó tính độ dài xâu ký tự trên mỗi dòng và ghi ra một tập tin mới theo định dạng sau: <độ dài dòng> <Nội dung dòng>
- ❖ Ví dụ, với một dòng trong tập tin đầu vào:  
*The quick brown fox jumps over the lazy dog.*  
trong tập tin đầu ra ở dòng tương ứng sẽ là:  
*44 The quick brown fox jumps over the lazy dog.*

# fscanf & fprintf

---

# Bài tập 1.26

- Viết chương trình để đọc một dãy số nhập từ bàn phím và ghi chúng ra tệp “out.txt” theo thứ tự ngược lại. Ngoài ra, tổng các số được ghi vào cuối file.
- Cú pháp nhập liệu từ bàn phím như sau: Số đầu tiên là số lượng các số trong dãy sẽ nhập, sau đó là dãy các số nguyên. Ví dụ: khi người dùng nhập: 4 12 -45 56 3
- “4” là số các số sẽ được nhập, bao gồm “12 -45 56 3”. Nội dung của tệp tin “out.txt” sẽ là, với 26 là tổng của 4 số
- 3 56 -45 12 26
- Vì số lượng các số nhập thay đổi theo mỗi lần chạy, chương trình cần cấp phát động bộ nhớ cho các số này sử dụng hàm malloc( ).

# Bài tập 1.27

- ❖ Tạo một tập tin văn bản có tên product.txt, mỗi dòng trong đó chứa thông tin về một sản phẩm: ID (kiểu int), Product Name (xâu ký tự không chứa ký tự trắng), Price (kiểu double). Các trường dữ liệu trên phân tách với nhau bởi một ký tự space hoặc tab. Ví dụ
  - 1      Samsung\_Television\_4K      20000000
  - 2      Apple\_MacBook\_2020    18560000
- ❖ Viết chương trình đọc tập tin trên vào một mảng các phần tử cấu trúc và sau đó hiện nội dung mảng trên ra màn hình dưới dạng:
  - No    Product Name                      Price
  - 1      Samsung\_Television\_4K              20000000
  - ...

# Bài tập 1.28

- Tạo một file văn bản nội dung là danh sách lớp gồm ít nhất 6 sinh viên. Mỗi dòng gồm 4 trường sau:
- STT(số thứ tự) Mã số sinh viên Họ và tên (không chứa ký tự trắng) Số điện thoại. Ví dụ
  - 1 20181110 Bui\_Van 0903112234
  - 2 20182111 Joshua\_Kim 0912123232
- Viết chương trình đọc tập tin trên vào một mảng các cấu trúc phù hợp. Chương trình yêu cầu nhập bổ sung thêm trường điểm cho mỗi sinh viên sau đó ghi lại kết quả vào tập tin bangdiem.txt (transcript.txt) gồm tất cả các trường nói trên (cùng trường điểm).

# Tập nhị phân



# Ví dụ 1.4.

---

# Bài tập 1.29

---

- Viết chương trình sao chép tập tin (từ lab1.txt sang lab1a.txt) tương tự các bài tập trước nhưng sử dụng các thao tác đọc ghi tập tin theo khối dữ liệu.
- Sử dụng các hàm: fread, fwrite, feof

# Bài tập 1.30

---

- ❖ Điều chỉnh bài tập 1.29 sử dụng các tham số dòng lệnh.

# Bài tập 1.31

---

- ❖ Điều chỉnh bài tập 1.30 sao cho chương trình in ra kích thước tệp tính bằng Bytes sau khi hoàn thành sao chép.

# Bài tập 1.32 \*

- Viết chương trình sao chép tập tin theo nhiều chế độ, hoạt động với giao diện menu dòng lệnh với các chức năng chính như sau:
  1. Copy by character
  2. Copy by line
  3. Copy by block - optional size
  4. Quit
- Ở mỗi chức năng sao chép, sau khi hoàn thành việc sao chép, hiển thị thời gian thực hiện theo đơn vị mili giây để so sánh.  
Chú ý: Tập tin nguồn phải là tập tin văn bản có kích thước tối thiểu là 640KB.

# Bài tập 1.33

---

- Giả sử bạn cần quản lý một danh bạ điện thoại của mình bằng chương trình. Định nghĩa một cấu trúc biểu diễn danh bạ gồm các trường "name," "telephone number," "e-mail address," và khai báo một mảng chứa tối đa 100 phần tử thuộc kiểu cấu trúc trên.
- Nhập liệu cho khoảng 10 phần tử mảng.
- Chương trình sau đó ghi nội dung mảng với các phần tử nói trên vào tập tin có tên phonebook.dat sử dụng hàm fwrite.
- Đọc lại dữ liệu từ tập tin vào mảng sử dụng hàm fread và in nội dung mảng ra màn hình để kiểm tra.

# Bài tập 1.34

- Viết chương trình đọc tập tin bảng điểm (bangdiem.txt kết quả của bài tập về danh sách sinh viên và bảng điểm đã cho) và lưu trữ dữ liệu sử dụng bộ nhớ động, sau đó ghi chúng ra tập tin nhị phân grade.dat (chứa mảng các phần tử kiểu cấu trúc về sinh viên).
- Chương trình có thể: đọc file grade.dat và hiển thị bảng điểm trên màn hình, tìm kiếm một sinh viên dựa trên mã số sinh viên và cập nhật điểm mới nhập từ người dùng và lưu vào tập tin.
- Chương trình nên được viết với tương tác menu dòng lệnh:
  1. TextToDat
  2. Display Dat file
  3. Search and Update.
  - 4 Quit.

# Bài tập 1.35

---

- Viết chương trình đọc một phần cụ thể trong dữ liệu danh bạ lưu trữ trong tập tin phonebook.dat. Ví dụ từ dữ liệu danh bạ (bản ghi) thứ 2 đến thứ 3, hay từ thứ 3 đến thứ 6. Sau đó thay đổi giá trị trường email và ghi lại vào tập tin ở đúng vị trí đã trích xuất.
- Chương trình cần cấp phát bộ nhớ để lưu trữ đúng lượng dữ liệu đọc ra từ tập tin khi chạy chương trình. Ví dụ cần mảng động với 4 phần tử cấu trúc để lưu dữ liệu từ phần tử thứ 3 đến thứ 6.



# Bài tập 1.36

- Viết chương trình chuyển đổi dữ liệu từ điển Việt Anh từ định dạng văn bản sang nhị phân.
- Dữ liệu có tại
- <http://www.denisowski.org/Vietnamese/vnedict.txt>
  - Miên Đất Hứa : the Promised Land
  - Miến : Burma (short for Miến Điện)
  - Miến Điện : Burma
  - Miền Trung : Central Vietnam
- Sau đó chương trình đọc dữ liệu từ tập tin nhị phân, hỏi người dùng vị trí bắt đầu và kết thúc của mục từ và hiển thị các từ nằm ở các vị trí này trong từ điển.

# Bài tập 1.37

- Từ các trang web của các showroom điện thoại di động, xây dựng một tập tin văn bản có tên PhoneDB.txt chứa thông tin về ít nhất 20 mẫu điện thoại gần đây như iPhone, Samsung, Oppo, Huawei, mỗi điện thoại một dòng.. theo định dạng sau
  - Model Memory Space (GB) Screen Size (inches) Price
- Viết chương trình có giao diện menu như sau:
  1. Import DB from text: Đọc file PhoneDB.txt và chuyển thành định dạng nhị phân PhoneDB.dat sử dụng kỹ thuật cấp phát động bộ nhớ.
  2. Import from DB: Đọc dữ liệu từ file PhoneDB.dat và nạp vào bộ nhớ chương trình. Cho phép người dùng lựa chọn hai chế độ đọc: Đọc toàn bộ và đọc một phần (chỉ định vị trí bản ghi bắt đầu và kết thúc).
  3. Print All Database: Hiển thị dữ liệu về các mẫu điện thoại trên màn hình, mỗi mẫu một dòng và căn thẳng các cột.
  4. Search by phone by Phone Model: Tìm kiếm điện thoại dựa trên model do người dùng nhập.
  5. Exit

# Bài tập 1.38. Chia tách và ghép File

- Sử dụng tập tin phonebook.dat (kết quả của bài tập trên Lab) chứa ít nhất 20 số liên lạc. Viết các chương trình sau
- Chương trình **filesplit** nhận hai đối số: tên file nguồn (.dat) và một số nguyên N và tên hai file kết quả. Nó sẽ tách file nguồn thành 2 file, trong đó file đầu tiên chứa N số liên lạc đầu tiên và file thứ hai chứa các số liên lạc còn lại. Ví dụ
  - **filesplit** phone.dat 10 phone1.dat phone2.dat
- Chương trình **filemerge** ghép hai file đã tách thành một file:
  - **filemerge** phone1.dat phone2.dat phone.dat
- Chương trình **fileread** đọc và hiển thị danh sách các số liên lạc chứa trong một file .dat bất kỳ ra màn hình. Nó được sử dụng để kiểm tra kết quả thực hiện các chương trình filesplit and filemerge.



25 YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

