

# Chapter 8. File

# NỘI DUNG

- Giới thiệu
- Một số hàm trên file text
  - ✓ File văn bản với dữ liệu số có cấu trúc giản đơn (VD1)
  - ✓ File văn bản với dữ liệu số có cấu trúc mảng 1 chiều (VD2)
  - ✓ File văn bản với dữ liệu số có cấu trúc mảng 2 chiều (VD3)
- File văn bản với dữ liệu ký tự và chuỗi (VD4)
- File nhị phân (VD5)
- Ví dụ tổng hợp (VD6, VD7, VD8)

# GIỚI THIỆU

Chương này trình bày các thao tác cần thiết nhất đối với file văn bản để giải quyết vấn đề lưu trữ dữ liệu bền vững trên đĩa từ.

Chương này hữu ích cho các vấn đề bài toán cần xử lý dữ liệu vào/ra từ file; hữu ích cho các học phần tiếp theo như: cấu trúc dữ liệu, hệ điều hành, cơ sở trí tuệ nhân tạo,...

# Một số hàm trên file text

## MỘT SỐ HÀM TRÊN FILE TEXT

- Khai báo biến con trỏ kiểu FILE:

FILE \* <pointer\_varname>;

trong đó pointer\_varname là tên biến con trỏ kiểu FILE

## Mở tập tin

```
FILE *fopen (const char * filename, const char *mode); //wt,rt
```

## Đóng tập tin

```
void fclose (FILE * pointer_varname);
```

## Ghi dữ liệu vào tập tin

```
void fprintf (FILE * pointer_varname,const char formattext, varname);
```

## Đọc dữ liệu từ tập tin

```
void fscanf (FILE * pointer_varname, const char formattext, &varname);
```

Hàm feof(FILE \* pointer\_varname)

## Lưu ý

Nếu không biết trước kích thước của file thì sử dụng hàm feof để duyệt qua các phần tử của file.

```
while (!feof(f))
```

```
{
```

```
...
```

```
}
```

# File văn bản với dữ liệu kiểu số

## Ví dụ 1

Cho file văn bản num.inp có một dòng duy nhất chứa 3 số nguyên; các số cách nhau ít nhất một khoảng trắng.

Hãy tính tổng và trung bình cộng của 3 số trên;

Kết quả ghi ra file num.out

-Dòng đầu ghi giá trị tổng

-Dòng thứ hai ghi giá trị trung bình cộng của 3 số đó.

num.inp	num.out
3 4 6	13
	4.33

## Ví dụ 2

Cho file văn bản sequence.inp có cấu trúc như sau:

- Dòng đầu ghi số nguyên dương n;
- Trong các dòng tiếp theo ghi n số; các số cách nhau một khoảng trắng.

Hãy thực hiện các công việc sau:

- a.Tính tổng các phần tử
- b.Tìm giá trị lớn nhất
- c.Đếm số lượng số nguyên tố.

## Ví dụ 2 (...)

d.Đếm số lượng số vừa nguyên tố vừa đối xứng.

(số đối xứng: 4,1001,121,222,...)

e.Tính tổng các chữ số của tất cả các số.

Kết quả ghi ra file sequence.out; mỗi dòng ghi kết quả mỗi câu trên.

note: các hàm srand, rand trong môi trường devc: thư viện **cstdlib** hoặc **bits/stdc++.h**

# Test

sequence.inp

10// so phan tu cua file

3 7 27

4 2 5 131

100 101

121

sequence.out

Cau a: 501

Cau b: 131

Cau c: 6

Cau d: 6

Cau e: 42

## Ví dụ 3

Cho file văn bản table.inp có cấu trúc như sau:

- Dòng đầu ghi hai số nguyên dương m và n;
- Trong m dòng tiếp; mỗi dòng ghi n số nguyên không âm; các số cách nhau ít nhất một khoảng trắng.

Hãy thực hiện các công việc sau:

- a.Tính tổng các phần tử
- b.Đếm số lượng số nguyên tố
- c.Tính tổng các phần tử trên mỗi dòng

- d.Tính tổng các phần tử trên mỗi cột
  - e.Đếm số lượng số chẵn, số lượng số lẻ của bảng.
- Kết quả ghi ra file table.out; mỗi dòng ghi kết quả mỗi câu trên.

# Test

table.inp

4 6

3 7 5 1 9 3

0 0 7 9 8 7

6 2 100 11 200 4

8 3 7 13 17 29

table.out

Cau a: 459

Cau b: 13

Cau c: 28 31 323 77

Cau d: 17 12 119 34 234 43

Cau e: 9 15

# File văn bản với dữ liệu ký tự và chuỗi ký tự

# Giới thiệu một số lệnh thường sử dụng

- Hàm fputc(int ch, FILE \*filepointer) ; fprintf
- Hàm fgetc(FILE \*filepointer); fscanf
- int fputs(const char\*s, FILE \* filepointer); fprintf
- char \*fgets(char \*s, int n, FILE \*filepointe);
- ❖ Các mã thường dùng:
  - 32: Mã khoảng trắng
  - 10: Mã xuống dòng/kết thúc dòng
  - 13: Mã phím Enter
  - 127: Mã kết thúc file

## HÀM XỬ LÝ KÝ TỰ

Hàm fputc(int ch, FILE \*filepointer) ;

- ch là một giá trị nguyên không dấu, filepointer là con trỏ file.
- Hàm này ghi lên file filepointer một ký tự có mã bằng  $m=ch\%255$ .

Hàm fgetc(FILE \*filepointer)

- Hàm đọc một ký tự từ file filepointer. Nếu thành công hàm cho mã đọc được có giá trị từ 0 đến 255.

## Ví dụ 4a

Nhập các ký tự ghi vào file đên khi gặp ký tự 0 thì dừng lại.

## Ví dụ 4b

Đọc từng ký tự trong file có tên là filechar.inp và xuất ra màn hình.

(Giả thiết file không có khoảng trắng đầu,cuối mỗi chuỗi; giữa các từ chỉ có 1 ký tự khoảng trắng)

Q1:Tìm mã ascii của mỗi ký tự ?

Q2:Hỏi file chứa bao nhiêu ký tự (chữ cái hoặc chữ số )?

## HÀM XỬ LÝ CHUỖI KÝ TỰ

**int fputs(const char\*s, FILE \* filepointer);**

- s là con trỏ trỏ tới địa chỉ đầu của một chuỗi ký tự.
- Lệnh này có tác dụng ghi chuỗi s lên file filepointer.

## **char \*fgets(char \*s, int n, FILE \*filepointe)**

- s là con trỏ kiểu char trỏ tới một vùng nhớ đủ lớn để chứa chuỗi ký tự đọc từ file.
- n là số nguyên xác định độ dài cực đại của dãy ký tự cần đọc.
- Lệnh này có tác dụng đọc một dãy ký tự từ file filepointe chứa vào vùng nhớ s.

# Ghi chú khi sử dụng lệnh fgets

- Việc đọc sẽ kết thúc khi:
  - Hoặc đã đọc n-1 ký tự;
  - Hoặc gặp dấu xuống dòng (mã 10); khi đó mã 10 được đưa vào chuỗi kết quả hoặc gặp dấu kết thúc file.
- File có nhiều chuỗi:
  - Mỗi chuỗi có 1 ký tự cuối dòng;
  - Chuỗi cuối file chỉ có ký tự kết thúc file mà không có ký tự kết thúc dòng.
  - Có thể loại bỏ mã xuống dòng theo cách sau:

```
if(s[strlen(s)-1]==10)
    s[strlen(s)-1]=NULL;
```

## Ví dụ 5a

VTC nhập n chuỗi; ghi mỗi chuỗi vào trên một dòng của file “str.txt”

## Ví dụ 5b

Đọc một file văn bản có tên là chuoi.inp trong đó chứa nhiều chuỗi; mỗi chuỗi trên mỗi dòng; kích thước mỗi chuỗi tối đa là 256 ký tự.

Hãy tìm chiều dài của mỗi chuỗi.