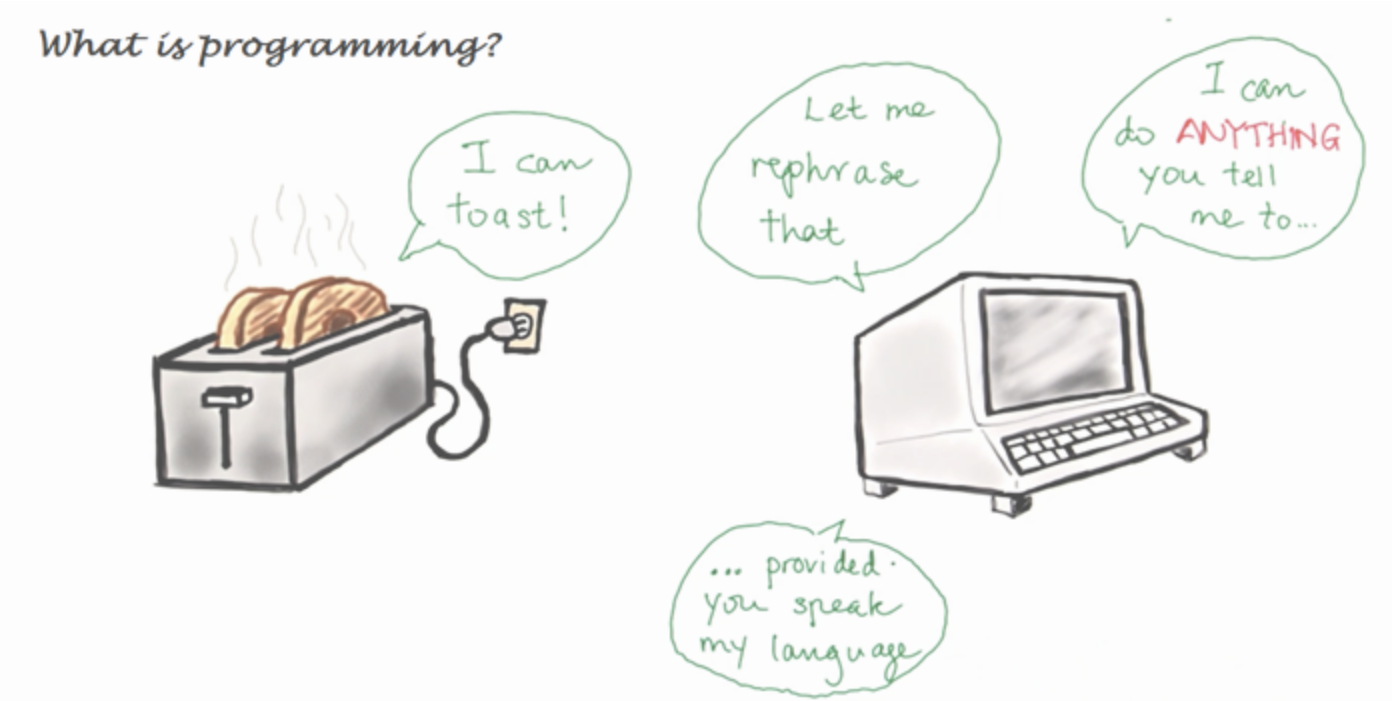


Tuần 12:

NHẬP XUẤT TẬP TIN

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT

Mục đích

- Hiểu được thế nào là tập tin.
- Biết cách khai báo và sử dụng các dạng của tập tin.

Yêu cầu

- Hiểu lý thuyết và vận dụng để giải bài tập.
- Hoàn thành hết bài tập.

Nội dung

- Khái niệm dòng (stream)
- Một số khái niệm về tập tin
- Các thao tác trên tập tin
- Truy cập tập tin văn bản
- Truy cập tập tin nhị phân

Tại sao cần phải có tập tin?

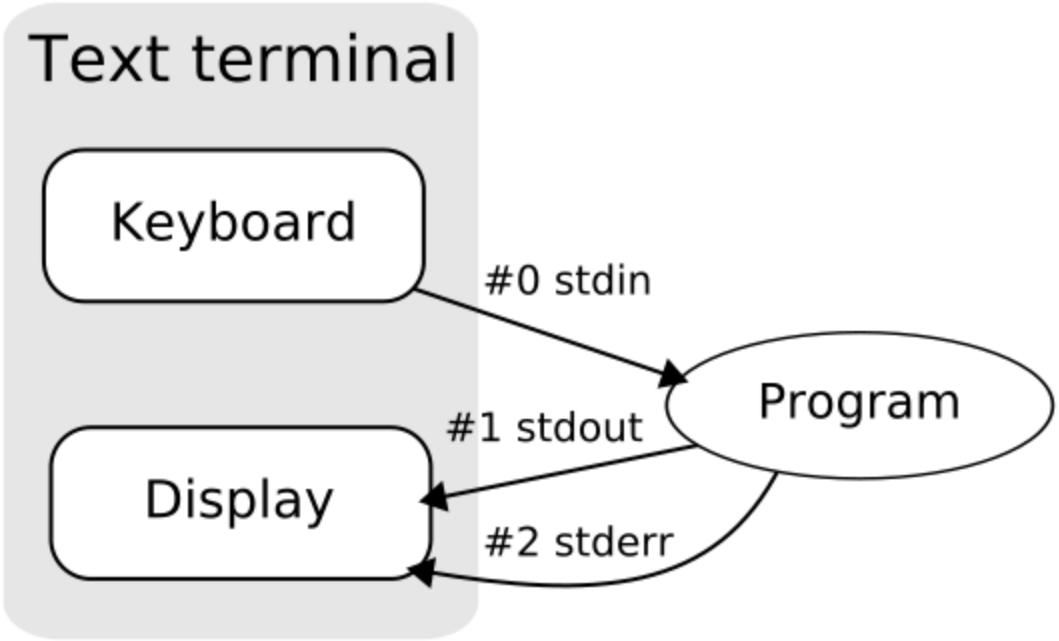
- Khi chương trình kết thúc: Dữ liệu vào, dữ liệu ra đều mất hết.

➔ Cần Tập tin để lưu lại.

- Tập tin giúp:
  - Tiết kiệm thời gian nhập dữ liệu.
  - Tiết kiệm thời gian chạy chương trình.
  - Sử dụng được nhiều lần.

Nhập xuất

- Khái niệm



- Trong C, tất cả hoạt động nhập xuất được thực hiện với các dòng (streams).
- Thiết bị nhập chuẩn là bàn phím.
- Thiết bị xuất chuẩn là màn hình.
- Thiết bị vừa nhập vừa xuất: tập tin

Dòng (stream)

- Khái niệm
  - Stream là chuỗi byte dữ liệu
    - “Chảy” vào chương trình gọi là stream nhập (input stream).
    - “Chảy” ra chương trình gọi là stream xuất (output stream)
  -

Dòng (stream)

- Phân loại
  - Stream văn bản (text)
    - Chỉ chứa các ký tự.
    - Tổ chức thành từng dòng, mỗi dòng tối đa 255 ký tự, kết thúc bởi ký tự cuối dòng '\0' hoặc ký tự sang dòng mới '\n'.
  - Stream nhị phân (binary)
    - Chứa các byte.
    - Được đọc và ghi chính xác từng byte.
    - Xử lý dữ liệu bất kỳ, kể cả dữ liệu văn bản.
    - Được sử dụng chủ yếu với các tập tin trên đĩa.

Dòng (stream)

- Các stream chuẩn định nghĩa sẵn:

Tên	Stream	Thiết bị
stdin	Nhập chuẩn	Bàn phím
stdout	Xuất chuẩn	Màn hình
stderr	Lỗi chuẩn	Màn hình
stdprn	In chuẩn	Máy in (LPT1:)
stdaux	Phụ chuẩn	Cổng nối tiếp COM 1:

- Ví dụ: với hàm fprintf
  - Xuất ra màn hình: fprintf(stdout, “Xin chao”);
  - Xuất ra máy in: fprintf(stdprn, “Xin chao”);
  - Xuất ra thiết bị báo lỗi: fprintf(stderr, “Xin chao”);
  - Xuất ra tập tin (stream fp): fprintf(fp, “Xin chao”);

Tập tin

Đặc điểm:

- Kích thước dữ liệu giới hạn và được lưu trữ tạm thời
  - Nhập: gõ từ bàn phím.
  - Xuất: hiển thị trên màn hình.
  - Lưu trữ dữ liệu: trong bộ nhớ RAM.

→ Mất thời gian, không giải quyết được bài toán với kích thước dữ liệu lớn.

→ Cần một thiết bị lưu trữ sao cho dữ liệu vẫn còn khi kết thúc chương trình, có thể sử dụng nhiều lần và kích thước không hạn chế.

Tập tin

- Khái niệm
  - Tập hợp thông tin (dữ liệu) được tổ chức theo một dạng nào đó với một tên xác định.
  - Một dãy byte liên tục (ở góc độ lưu trữ).
  - Được lưu trữ trong các thiết bị lưu trữ ngoài như đĩa mềm, đĩa cứng, USB…
    - Vẫn tồn tại khi chương trình kết thúc.
    - Kích thước không hạn chế (tùy vào thiết bị lưu trữ)
  - Cho phép đọc dữ liệu (thiết bị nhập) và ghi dữ liệu (thiết bị xuất).

Tập tin

- Phân loại
  - Theo người sử dụng: quan tâm đến nội dung tập tin nên sẽ phân loại theo phần mở rộng
    →.EXE, .COM, .CPP, .DOC, .PPT, …
  - Theo người lập trình: tự tạo các stream tường minh để kết nối với tập tin xác định nên sẽ phân loại theo cách sử dụng stream trong C

→tập tin kiểu văn bản (ứng với stream văn bản) và tập tin kiểu nhị phân (ứng với stream nhị phân).

Phân loại tập tin

- Tập tin kiểu văn bản (stream văn bản)
  - Dãy các dòng kế tiếp nhau.
  - Mỗi dòng dài tối đa 255 ký tự và kết thúc bằng ký hiệu cuối dòng (end_of_line).
  - Dòng không phải là một chuỗi vì không được kết thúc bởi ký tự '\0'.
  - Khi ghi '\n' được chuyển thành cặp ký tự CR (về đầu dòng, mã ASCII 13) và LF (qua dòng, mã ASCII 10).
  - Khi đọc thì cặp CR-LF được chuyển thành '\n'.

Phân loại tập tin

- Tập tin kiểu nhị phân (stream nhị phân)
  - Dữ liệu được đọc và ghi một cách chính xác, không có sự chuyển đổi nào cả.
  - Ký tự kết thúc chuỗi '\0' và end_of_line không có ý nghĩa là cuối chuỗi và cuối dòng mà được xử lý như mọi ký tự khác.

Biến & con trỏ tập tin

- Biến tập tin
  - Được dùng để đại diện cho một tập tin
  - Các thao tác lên tập tin sẽ được thực hiện thông qua biến này
- Con trỏ tập tin
  - Tại mỗi thời điểm, sẽ có một vị trí của tập tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra
  - Ta hình dung có 1 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

Các thao tác trên tập tin

- Khai báo biến tập tin
- Mở tập tin
- Đóng tập tin
- Kiểm tra đến cuối tập tin hay chưa?
- Di chuyển con trỏ tập tin về đầu tập tin - Hàm rewind()

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;

Mở tập tin (1/3)

- **Cú pháp**:

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

- **Ý nghĩa:**
  - Trả về con trỏ tập tin của tập tin được mở
  - Trả về NULL nếu có lỗi

Mở tập tin (2/3)

- 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

Mở tập tin (3/3)

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

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

=> *mở tập tin để ghi*

=> *nếu tập tin đã tồn tại rồi thì tập tin sẽ bị xóa và một tập tin mới được tạo ra*

Đóng tập tin

- **Cú pháp**: int fclose(FILE *f)
  - Ghi dữ liệu còn lại trong vùng đệm vào tập tin và đóng lại tập tin
  - f là con trỏ tập tin được mở bởi hàm fopen()
  - Giá trị trả về là 0 báo rằng việc đóng tập tin thành công
  - Giá trị trả về là EOF nếu có xuất hiện lỗi

- **Cú pháp**: int fcloseall()
  - Đóng tất cả các tập tin lại
  - Trả về 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.
  - Trả về EOF nếu cuối tập tin được chạm tới, ngược lại trả về 0.

Di chuyển con trỏ về đầu tập tin - rewind()

- Cú pháp:

```
void rewind(FILE *f)
```

- Ý nghĩa:
  - Làm cho con trỏ quay về đầu tập tin như khi mở nó

Truy cập tập tin văn bản

- Ghi dữ liệu lên tập tin văn bản
- Đọc dữ liệu từ tập tin văn bản

Ghi dữ liệu lên tập tin văn bản (1/4)

Hàm putc()

```
int putc(int c, FILE *f)
```

- Được dùng để ghi một ký tự lên một tập tin văn bản đang được mở (liên kết với con trỏ f) để làm việc
- c chứa mã Ascii của ký tự
- Hàm này trả về EOF nếu gặp lỗi

Ghi dữ liệu lên tập tin văn bản (2/4)

Hàm fputs()

```
int fputs(const char *buffer, FILE *f)
```

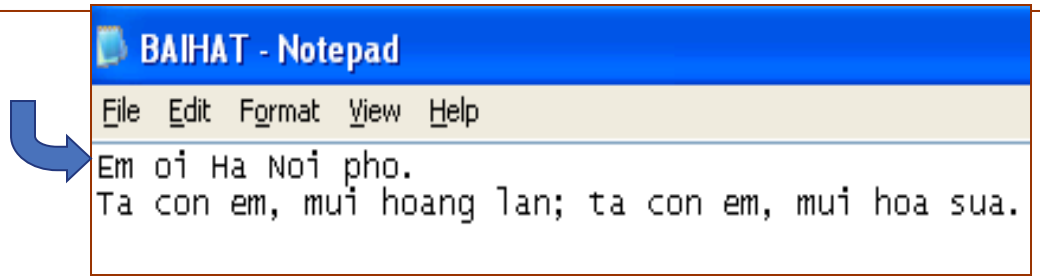
- Được dùng để ghi một chuỗi ký tự chứa trong vùng đệm lên tập tin văn bản
- Hàm này trả về giá trị 0 nếu buffer chứa chuỗi rỗng và trả về EOF nếu gặp lỗi

Ghi dữ liệu lên tập tin văn bản (3/4)

- **Ví dụ:** Viết chương trình ghi chuỗi ký tự lên tập tin văn bản **D:\\Baihat.txt**

```
#include<stdio.h>
#include<conio.h>
int main(){
   FILE *f;
   clrscr();
   f=fopen("D:\\Baihat.txt","w+");

   if (f!=NULL){
      fputs("Em oi Ha Noi pho.\n",f);
      fputs("Ta con em, mui hoang lan; ta con em, mui hoa sua.",f);
      fclose(f);
   }
   getch();
   return 0;
}
```



Ghi dữ liệu lên tập tin văn bản (4/4)

- **Hàm fprintf()**

```
fprintf(FILE *f, const char *format, varexpr)
```

  - Được dùng để ghi dữ liệu có định dạng lên tập tin văn bản.
  - format: chuỗi định dạng (giống với các định dạng của hàm printf())
  - varexpr: danh sách các biểu thức, mỗi biểu thức cách nhau dấu phẩy (,)

Đọc dữ liệu từ tập tin văn bản (1/4)

- **Hàm getc()**

```
int getc(FILE *f)
```

  - Được dùng để đọc dữ liệu từ tập tin văn bản đang được mở để làm việc (liên kết với f)
  - Hàm này trả về mã Ascii của một ký tự được đọc (kể cả EOF)

Đọc dữ liệu từ tập tin văn bản (2/4)

- **Hàm fgets()**

```
char *fgets(char *buffer, int n, FILE *f)
```

  - Đọc 1 chuỗi ký tự từ tập tin văn bản đang được mở.
  - Đọc cho đến khi đủ n ký tự hoặc gặp ký tự xuống dòng '\n' (ký tự này được đưa vào chuỗi kết quả) hay gặp ký tự kết thúc EOF (ký tự này không được đưa vào chuỗi kết quả)
  - buffer: chỉ đến vùng nhớ đủ lớn chứa các ký tự nhận được
  - Ký tự NULL ('\0') tự động được thêm vào cuối chuỗi kết quả lưu trong vùng đệm
  - 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

Đọc dữ liệu từ tập tin văn bản (3/4)

- **Hàm fscanf()**

  fscanf(FILE *f, const char *format, varlist)

  - Được dùng để đọc dữ liệu từ tập tin văn bản vào danh sách các biến theo định dạng.
  - format: chuỗi định dạng (giống hàm scanf())
  - varlist: danh sách các biến mỗi biến cách nhau dấu phẩy (,).

Đọc dữ liệu từ tập tin văn bản (4/4)

- **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>
int main(){
 FILE *f1,*f2;
 clrscr();
 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();
 return 0;
}
```

Truy cập tập tin nhị phân

- Ghi dữ liệu lên tập tin nhị phân
- Đọc dữ liệu từ tập tin nhị phân
- Di chuyển con trỏ tập tin
- Ví dụ

Ghi dữ liệu lên tập tin nhị phân

Hàm fwrite()

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

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

Đọc dữ liệu từ tập tin nhị phân

- **Hàm fread()**

  size_t fread(const void *ptr, size_t size, size_t n, FILE *f)

  - 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 bằng n hay nhỏ hơn n nếu đã chạm đến cuối tập tin hoặc có lỗi xuất hiện

Di chuyển con trỏ tập tin

- Hàm fseek()

  int fseek(FILE *f, long offset, int whence)

  - Được dùng để di chuyển con trỏ tập tin đến vị trí chỉ định
  - 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

0	SEEK_SET	Vị trí đầu tập tin
1	SEEK_CUR	Vị trí hiện tại của con trỏ tập tin
2	SEEK_END	Vị trí cuối tập tin

  - Kết quả trả về của hàm là 0 nếu việc di chuyển thành công. Nếu không thành công, 1 giá trị khác 0 (đó là 1 mã lỗi) được trả về.

Ví dụ

- Viết chương trình ghi lên tập tin CacSo.Dat 3 giá trị số (thực, nguyên, nguyên dài). Sau đó đọc các số từ tập tin vừa ghi và hiển thị lên màn hình

```
#include<stdio.h>
#include<conio.h>
int main(){
   clrscr();
   FILE *f;
   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();
   return 0;
}
```

Tổng kết

- Tập tin kiểu văn bản:
  - Mỗi dòng dài tối đa 255 ký tự và kết thúc bằng ký hiệu cuối dòng (end_of_line).
  - Dòng không phải là một chuỗi.
  - Khi ghi '\n' được chuyển thành cặp ký tự CR và LF.
  - Khi đọc thì cặp CR-LF được chuyển thành '\n'.
- Tập tin kiểu nhị phân:
  - Dữ liệu được đọc và ghi một cách chính xác.
  - Ký tự kết thúc chuỗi '\0' và end_of_line không có ý nghĩa.

Kiểm tra kiến thức (1/2)

1) Đối tượng nào đại diện cho 1 tập tin trong C?
   A. FILE*
   B. fopen
   C. printf
   D. fprintf
2) Trước khi đọc hoặc ghi vào tập tin, bạn cần làm gì?
   A. Gọi fopen
   B. Tạo tập tin
   C. Gọi fclose
   D. Dùng fprintf

Kiểm tra kiến thức (2/2)

3) Làm sao để ghi 1 chuỗi vào tập tin văn bản?
   A. Mở tập tin và dùng fprintf.
   B. Mở tập tin và dùng fprintf, kết quả sẽ ghi vào tập tin thay vì hiển thị lên màn hình.
   C. Mở tập tin and dùng fputc nhiều lần.
   D. Dùng fread để đọc dữ liệu đưa vào tập tin.
4) Chế độ nào dùng để mở và ghi tiếp vào tập tin thay vì tạo tập tin mới?
   A. a
   B. r
   C. w
   D. W+

Bài tập tổng kết

- Viết ứng dụng quản lý sinh viên:
  - Khai báo cấu trúc sinh viên gồm thành phần: MSSV, HỌ TÊN, NGÀY SINH, QUÊ QUÁN, ĐIỂM TRUNG BÌNH TÍCH LUỸ, ĐIỂM RÈN LUYỆN.
  - Khai báo mảng cấu trúc để quản lý 1 lớp gồm 50 sinh viên.
  - Nhập thông tin của n sinh viên, lưu thông tin vào tập tin Sinhvien.txt.
  - Đọc tập tin Sinhvien.txt và in thông tin n sinh viên.
  - Sắp xếp danh sách sinh viên theo MSSV và lưu vào tập tin sv.txt.
  - Nhập danh sách sinh viên từ 1 tập tin văn bản khác.

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT