









TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM
MÔN: **KỸ THUẬT LẬP TRÌNH**

HƯỚNG DẪN THỰC HÀNH

Tập tin văn bản, Tập tin nhị phân, Tập tin CSV, Tập tin HTML

-  ĐẶNG HOÀNG HẢI
-  TRẦN THANH HẢI
-  ĐỖ NGUYỄN KHA
-  ĐÀO ANH VŨ
-  NGUYỄN LÊ HOÀNG DŨNG
-  NGÔ CHÁNH ĐỨC

TP.HCM, ngày 24 tháng 03 năm 2022

MỤC LỤC

1	Quy định	3
2	Tập tin văn bản	3
3	Tập tin nhị phân.....	4
3.1	Một số hướng dẫn	4
3.2	Bài tập.....	5
4	Tập tin HTML (tham khảo).....	6
4.1	Mô tả HTML.....	6
4.2	Một số thẻ HTML cơ bản	7
4.3	Bài tập.....	8
4.4	HTML và CSS	12

1 Quy định

Hoàn thành các bài tập dưới đây, tuân thủ các qui tắc lập trình đã học trong kỳ trước:

1. Qui tắc đặt tên
2. Qui tắc viết code
3. Qui tắc ghi chú (ghi chú mỗi project và ghi chú mỗi hàm)
4. Qui tắc chia file (.h và .cpp)
5. Qui tắc bố trí project (1 solution, n projects: Bai01, Bai02, ...)

2 Tập tin văn bản

Bài tập 1: Bảng 1 chứa mã nguồn của tập tin file1.c. Hãy thử chạy đoạn chương trình đó và trả lời câu hỏi trong phần chú thích.

```
int main(void)
{
    FILE *fout;
    int c;

    if((fout=fopen("file1.txt", "w")) == NULL)
    {
        printf("Cannot open output file\n");
        exit(1);
    }

    printf("Input string\n");
    //Voi Windows ta cung Ctrl-z de thoat.
    //Voi Linux ta se dung lenh gi?
    printf("Ctrl-z if you want to exit\n");
    while ((c=getchar()) != EOF)
    {
        putc(c, fout);
    }

    fclose(fout);

    exit(0);
}
```

Bảng 1. Mã nguồn trong tập tin file1.c

{dhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn

Bài tập 2: Bảng 2 chứa mã nguồn trong tập tin file2.c và đoạn mã nguồn này dùng để đọc toàn bộ nội dung tập tin file1.txt (trong bài tập 2) và ghi ra màn hình. Hãy hoàn thiện đoạn mã nguồn trong bảng 2, chỗ có các dấu ****, và chạy thử.

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

int main(void)
{
    FILE *fin;
    char c[256];
    if((fin = fopen("file1.txt", "r")) == NULL){
        printf("Cannot open input file\n");
        exit(1);
    }

    printf("Contents of file1.txt:\n");
    while (/* **** */){
        printf("%s", c);
    }

    fclose(fin);

    exit(0);
}
```

Bảng 2. Mã nguồn trong tập tin file2.c

Bài tập 3: Viết chương trình đếm số lượng ký tự (bao gồm cả các ký tự như khoảng trắng, tab...) của một tập tin *.cpp cho trước.

3 Tập tin nhị phân

3.1 Một số hướng dẫn

Mở file dạng nhị phân để đọc	FILE* p = fopen("file1.bin", "rb");
Mở file dạng nhị phân để ghi mới	FILE* p = fopen("file1.bin", "wb");
Mở file dạng nhị phân để đọc và ghi	FILE* p = fopen("file1.bin", "rb+");
Mở file dạng nhị phân để ghi và đọc	FILE* p = fopen("file1.bin", "wb+");
Mở file dạng nhị phân để ghi thêm (append)	FILE* p = fopen("file1.bin", "ab");
#include <stdio.h>	

<pre>int main(int argc, char* argv[]) { FILE* p = fopen("file1.bin", "wb"); char* c = "abcdef"; if (p != NULL) // mở file thành công { // ghi c dưới dạng chuỗi byte (char) fwrite(c, sizeof(char), sizeof(c), p); fclose(p); } }</pre>	
Hàm ghi nội dung nhị phân. Trả về số mẫu tin đã ghi.	<pre>FILE* p = fopen("file1.bin", "wb"); int c[] = {1, 2, 3}; if (p != NULL) // mở file thành công { // ghi 3 số nguyên: 3*sizeof(int) ~ 12 byte int n = fwrite(c, sizeof(int), 3, p); printf("n = %d", n); // n = 3 fclose(p); }</pre>
Hàm đọc nội dung nhị phân. Trả về số mẫu tin đã đọc.	<pre>FILE* p = fopen("file1.bin", "rb"); int c[5]; if (p != NULL) // mở file thành công { // đọc 3 số nguyên: 5*sizeof(int) ~ 20 bytes int n = fread(c, sizeof(int), 5, p); printf("n = %d", n); // n = 5 fclose(p); }</pre>
Các hàm thao tác với con trỏ file	fseek ftell rewind feof

3.2 Bài tập

Bài tập 4: Hãy chạy đoạn mã nguồn sau. Sau đó mở file binary **file.bin** bằng [HexEdit](#) và xem nội dung file được tạo ra.

```
#include <stdio.h>
```

{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn

```
int main(int argc, char* argv[]) {
    FILE* p = fopen("file.bin", "wb");
    char* c = "abcdefghijklmnopqrstuvwxyz";
    if (p != NULL) { // mở file thành công
        // ghi c dưới dạng chuỗi byte (char)
        fwrite(c, sizeof(char), sizeof(c), p);
        fclose(p);
    }
}
```

Bảng 3. Mã nguồn cần thực thi

Bài tập 5: Cho cấu trúc như sau

```
typedef struct SV
{
    char name[10];
    char mssv[10];
    int diemSo[3];
} SinhVien;
```

Bảng 4. Cấu trúc SinhVien

Hãy viết chương trình nhập giá trị cho 1 mảng SinhVien. Xuất mảng SinhVien ra màn hình và ghi mảng này xuống tập tin "**sinhvien.bin**" ở dạng nhị phân. Mở tập tin bằng [HexEdit](#) và nhận xét.

Bài tập 6: Viết chương trình đọc lại tập tin "**sinhvien.bin**" ở bài tập trước vào 1 mảng SinhVien, in kết quả ra màn hình.

4 Tập tin HTML (tham khảo)

4.1 Mô tả HTML

HTML (HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ mô tả trang web. Trong HTML, mỗi thực thể được đặt trong một cặp thẻ (tag) đóng – mở.

Một trang HTML cơ bản gồm có:

- thành phần khai báo trang HTML: thẻ **<html>**
- thành phần khai báo thông tin trang cho trình duyệt hiểu: thẻ **<head>**
- thành phần khai báo thông tin hiển thị của trang: thẻ **<body>**

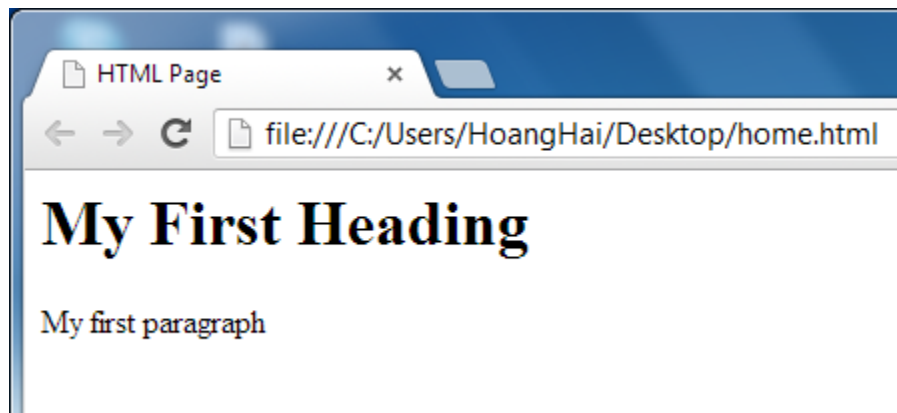
{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn

Một trang HTML mẫu được cung cấp dưới đây. Để ý các cặp đóng-mở thẻ: mở thẻ có dạng `<tên-thẻ>` và đóng có dạng `</tên-thẻ>`. Chú thích có dạng `<!--chú thích-->`.

```
<!DOCTYPE html>
<html>           <!--comment: html-->
  <head>          <!--head-->
    <title>HTML Page</title> <!--tiêu đề trang-->
  </head>         <!--head-->
  <body>          <!--body-->
    <h1>My First Heading</h1>
    <p>My first paragraph</p>
  </body>         <!--body-->
</html>          <!--kết thúc html-->
```

Bảng 5. Mã nguồn file `home.html`

Lưu trang này thành file có tên `home.html` (đuôi .html) và mở bằng trình duyệt web:



Hình 1. Mở `home.html` bằng Google Chrome

4.2 Một số thẻ HTML cơ bản

Các thẻ html cơ bản mà chúng ta quan tâm trong môn này gồm:

- Định dạng chữ `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `` `<i>` `<p>`
- Xuống dòng `
`
- Bảng `<table>` `<tr>` `<td>` `<th>`
- Ảnh ``
- Danh sách `` `` ``
- Kẻ ngang `<hr />`

{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn

Tham khảo các thẻ này theo bài hướng dẫn Lập trình Web: <http://goo.gl/2RiLj>.

4.3 Bài tập

Bài tập 7: Hãy tạo project chương trình C với các nội dung tập tin như sau. Chạy và xem kết quả.

html_helper.h

```
#pragma once
#include <stdio.h>
#include <stdlib.h>

FILE* openHTML(char* fileName, char* title = "");
FILE* closeHTML(FILE* pFile);
```

html_helper.cpp

```
#include "html_helper.h"

FILE* openHTML(char* fileName, char* title)
/**
 * Cách dùng: html = openHTML(fileName, "HTML Page");
 * html được gán cho giá trị con trỏ FILE* được tạo
 */
{
    FILE* pFile = fopen(fileName, "wt");

    if (pFile == NULL)
    {
        printf("Cannot open file %s.\n", fileName);
    }
    else
    {
        fprintf(pFile, "<html>\n");
        fprintf(pFile, "<head><title>%s</title></head>", title);
        fprintf(pFile, "\n<body>");
    }
    return pFile;
}

FILE* closeHTML(FILE* pFile)
/**
 * Cách dùng: html = closeHTML(html);
 * html là con trỏ FILE* sẽ được reset về NULL
 */
```


{dhhai, tthai, dnkha, davu, nlhdung, ncdudc}@fit.hcmus.edu.vn

```
{  
    fprintf(pFile, "\n</body>\n</html>");  
    return NULL;  
}
```

main.cpp

```
#include "html_helper.h"
```

```
int main()  
{  
    FILE* html = 0;  
    if ((html = openHTML("home.html", "HTML Page")))  
    {  
        html = closeHTML(html);  
    }  
    return 0;  
}
```

Bảng 6. Mã nguồn tạo file html rỗng

Bài tập 8: Viết tiếp chương trình ở **Bài tập 7**; bổ sung thêm các hàm sau đây

```
void br(FILE* pFile, int number = 1);  
void heading(FILE* pFile, char* content, int size);  
void para(FILE* pFile, char* content);  
void img(FILE* pFile, char* path, int width, int height);  
void hyperlink(FILE* pFile, char* path, char* content);
```

Bảng 7. Khai báo các hàm mới

Ví dụ sử dụng

Cách gọi	Xuất ra HTML
br(pFile);	
br(pFile, 3);	
heading(pFile, "My Home", 2);	<h2>My Home</h2>
para(pFile, "This is my home!");	<p>This is my home!</p>
img(pFile, "home/image.jpeg",	
hyperlink(pFile, "http://google.com.vn", "Link to Google");	Link to Google

Bảng 8. Ví dụ sử dụng các hàm mới

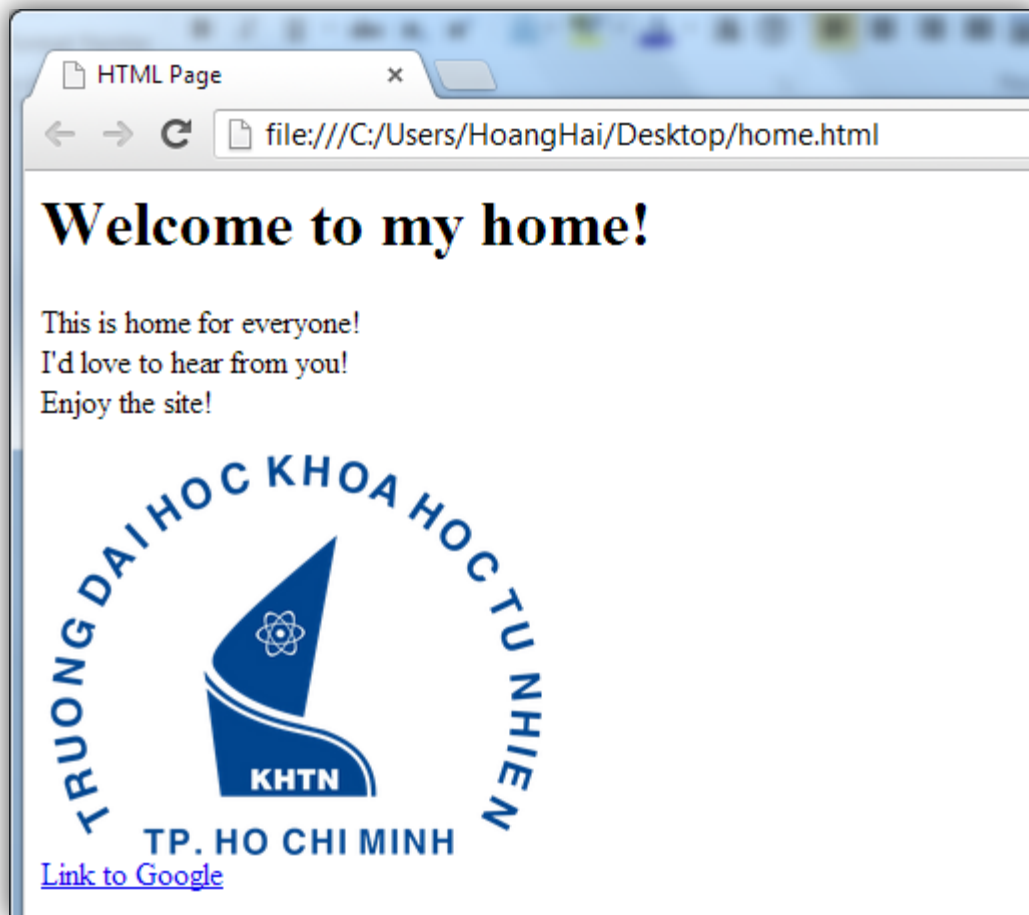
Sau khi cài đặt, sửa lại hàm main như sau

```
int main()
{
    char* fileName = "home.html";
    FILE* html = 0;
    if ((html = openHTML(fileName, "HTML Page")))
    {
        heading(html, "Welcome to my home!", 1);
        para(html, "This is home for everyone!\nI'd love to hear  
from you!\nEnjoy the site!");
        img(html, "http://www.hcmus.edu.vn/images/stories/logo-  
khtn2009_transparency.png", 250, 200);
        br(html);
        hyperlink(html, "http://google.com.vn", "Link to Google");
        html = closeHTML(html);
    }
    return 0;
}
```

Bảng 9. Chương trình main mới

Kết quả cần giống như sau:

{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn



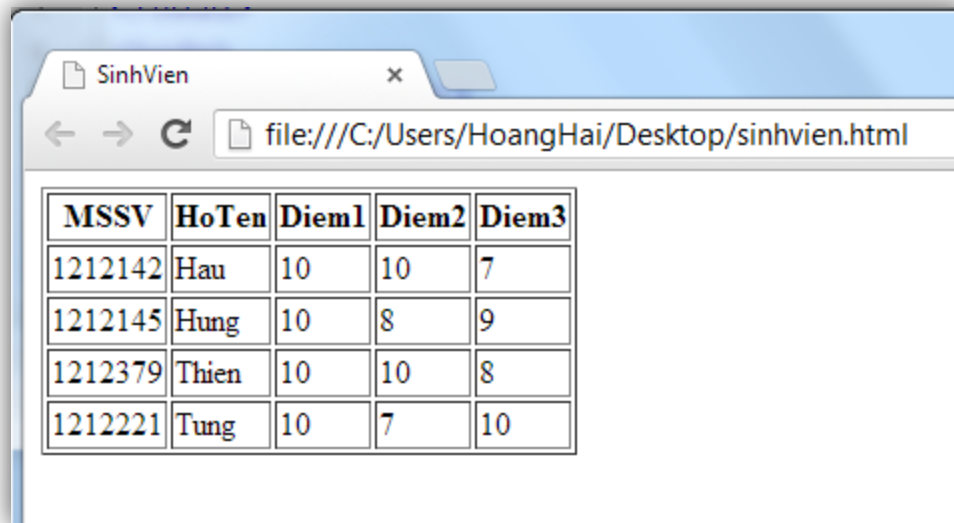
Hình 2. Mở [home.html](#) mới bằng Google Chrome

Bài tập 9: Tiếp tục bài tập trước, hãy tự thiết kế và cài đặt các hàm cho các thẻ HTML sau:

- Danh sách `` `` ``
- Kẻ ngang `<hr />`
- *** Bảng `<table>` `<tr>` `<td>` `<th>`

Bài tập 10: Viết chương trình C đọc tập tin nhị phân lưu theo cấu trúc SinhVien (**Bài tập 5:**), in kết quả ra thành bảng HTML như sau

{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn



Hình 3. Mở `sinhvien.html` bằng Google Chrome

4.4 HTML và CSS

Mặc dù HTML cung cấp một số thuộc tính định cho các đối tượng nhưng khá hạn chế. Do đó, người ta dùng thêm ngôn ngữ để hỗ trợ kiểu hiển thị là **CSS (Cascading Style Sheets)** đi kèm với trang web HTML thuần túy.

Trong môn này, các tập tin CSS sẽ được cung cấp sẵn (tập tin có đuôi `.css`), do đó sinh viên không cần quan tâm ngôn ngữ này được đặc tả như thế nào, chỉ cần biết rằng các đối tượng trong HTML có thể có kiểu hiển thị bằng cách gán thuộc tính **class** tương ứng.

Giả sử sinh viên được cung cấp sẵn file `style.css`. Để khai báo sử dụng (giống như `#include` trong C) thì thêm thẻ `<link>` vào phần `<head>`.

```
<head>
  <title>HTML Page</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

Bảng 10. Mã nguồn include `style.css`

Để sử dụng kiểu hiển thị, ví dụ như kiểu `bigHeader` (đã được định nghĩa trong `style.css`), viết tên kiểu trong thuộc tính **class** của thẻ:

{dhhai, tthai, dnkha, davu, nlhdung, ncduc}@fit.hcmus.edu.vn

```
<body>
    <h1 class="bigHeader">My First Heading</h1>
    <p>My first paragraph</p>
</body>
```

Bảng 11.Mã nguồn sử dụng kiểu hiển thị **bigHeader**

Bài tập 11: Hãy bổ sung tham số `cssClass` cho các hàm đã xây dựng trước đó, và thuộc tính đường dẫn file `.css` cho `html`. Ví dụ

```
FILE* openHTML(char* fileName, char* title = "", char* cssFile = "");
void heading(FILE* pFile, char* content, int size, char* cssClass="");
void para(FILE* pFile, char* content, char* cssClass = "");
```

Bảng 12.Khai báo hàm với tham số `cssClass`