**Nội dung**

**Hình thức học**

* Học online
* Giới thiệu qua về lý thuyết và 1 số bài code mẫu
* Thực hành:
  + Chạy lại code mẫu
  + Áp dụng có sửa đổi code mẫu trong 1 số bài tập tương tự
* Kiểm tra
  + Đầu hoặc cuối buổi
* Thời gian học từ 1h15 – gần 5h, nghỉ giải lao tầm 10’ và sẽ có khoảng 2 lần nghỉ

**Đánh giá**

* Giữa kỳ: 40%
* Cuối kỳ: 60%
* Giữa kỳ: gọi trình bày bài cũ/ bài tập để lấy điểm
* Cuối kỳ: Làm bài tập + bảo vệ lấy điểm

**Ngôn ngữ lập trình**

* Nên dùng C/C++
* Có thể dùng ngôn ngữ khác nếu bạn thấy thành thạo: Jave, C#, Python, java script, Php,…
* Khuyến khích nên dùng thư viện : C/C++ STL, Java, C# có nhiều thư viện phụ
* Dùng CTDL&TT để giải bài toán hiện tại (hiểu và áp dụng)

**Công cụ lập trình**

* Emac/Vim/gEdit + biên dịch dùng gcc++ (linux)
* Dev-C++, CFree, Visual code, Visual Studio, Eclipse
* Online compiler (nên tạo tài khoản)

**Liên hệ**

NGuyễn Duy Hiệp, bộ môn KHMT, viện CNTT&TT

[Hiepnd@soict.hust.edu.vn](mailto:Hiepnd@hust.edu.vn), [hiep.nguyenduy@hust.edu.vn](mailto:hiep.nguyenduy@hust.edu.vn)

Nên hỏi đáp trên Teams

Nội dung học:

1. Các kiểu dữ liệu cơ bản

**Ví dụ 1**. Xây dựng chương trình đếm tần số xuất hiện của các ký tự trong xâu ký tự độ dài tối đa (200) được nhập từ bàn phím

* Chỉ đếm ký tự và KHÔNG phân biệt chữ hoa/thường
* Chỉ đếm ký tự và có phân biệt HOA/thường
* Đếm tất cả các ký tự (kể cả ký tự đặc biệt) 128 ký tự trong nửa đầu bảng ASCII

===================

#include <stdio.h>

#include <string.h>

//const int ALPHABET\_LEN = 26; // khai bao hang so

#define ALPHABET\_LEN 26

int main()

{

int i, count[ALPHABET\_LEN] ={0}; // tu gan gia tri cac phan tu trong mang count la 0

char str[200];

printf("Nhap xau:");

gets(str);

// chi dem ky tu va khong phan biet HOA va thuong

for(i=0; i<strlen(str); i++)

{

if('a'<=str[i] && str[i]<='z')

count[str[i]-'a']++;

if('A'<=str[i] && str[i]<='Z')

count[str[i]-'A']++;

}

printf("Cac ky tu trong xau:\n");

for(i=0; i<ALPHABET\_LEN; i++)

if(count[i]>0) printf("%c - %d\n",i+'a',count[i]);

return 0;

}

===============================

#include <stdio.h>

#include <string.h>

//const int ALPHABET\_LEN = 52; // khai bao hang so

#define ALPHABET\_LEN 52

int main()

{

int i, count[ALPHABET\_LEN] ={0}; // tu gan gia tri cac phan tu trong mang count la 0

char str[200];

printf("Nhap xau:");

gets(str);

// chi dem ky tu va co phan biet HOA va thuong (dung 1 mang count)

for(i=0; i<strlen(str); i++)

{

if('a'<=str[i] && str[i]<='z') // nua dau mang

count[str[i]-'a']++;

if('A'<=str[i] && str[i]<='Z') // nua sau

count[str[i]-'A'+26]++;

}

printf("Cac ky tu trong xau:\n");

for(i=0; i<ALPHABET\_LEN; i++)

if(count[i]>0)

if(i<26) printf("%c - %d\n",i+'a',count[i]);

else printf("%c - %d\n",i+'A'-26 ,count[i]);

return 0;

}

==============================

#include <stdio.h>

#include <string.h>

//const int ALPHABET\_LEN = 52; // khai bao hang so

#define ALPHABET\_LEN 26

int main()

{

int i, count\_thuong[ALPHABET\_LEN] ={0}, count\_hoa[ALPHABET\_LEN] ={0}; // tu gan gia tri cac phan tu trong mang count la 0

char str[200];

printf("Nhap xau:");

gets(str);

// chi dem ky tu va co phan biet HOA va thuong (dung 1 mang count)

for(i=0; i<strlen(str); i++)

{

if('a'<=str[i] && str[i]<='z') // nua dau mang

count\_thuong[str[i]-'a']++;

if('A'<=str[i] && str[i]<='Z') // nua sau

count\_hoa[str[i]-'A']++;

}

printf("Cac ky tu trong xau:\n");

for(i=0; i<ALPHABET\_LEN; i++)

if(count\_thuong[i]>0) printf("%c - %d\n",i+'a',count\_thuong[i]);

for(i=0; i<ALPHABET\_LEN; i++)

if(count\_hoa[i]>0) printf("%c - %d\n",i+'A' ,count\_hoa[i]);

return 0;

}

================================

**Truyền tham số cho hàm**

Tên mảng = **Hằng con trỏ** trỏ đến địa chỉ ô nhớ đàu tiên cấp phát cho mảng

Vì thế mọi phép gán trực tiếp tên mảng đều KHÔNG thể thực hiện được

Int A[5] = {1,2,3,4,5};

Int B[5]

B = A ; // câu lệnh này không đúng

Truyền tham số cho hàm là con trỏ (địa chỉ) ==> hàm có thể thay đổi giá trị vùng nhớ mà con trỏ đang trỏ tới

int test(int \*a)  
{

\*a = 100;

}

============

#include <stdio.h>

#include <string.h>

int test(int \*a)

{

\*a = 100;

}

int main()

{

int i=7;

printf("i=%d\n",i);

test(&i);

printf("i=%d\n",i);

return 0;

}

====================

Nếu muốn hàm không được thay đổi giá trị của vùng nhớ mà con trỏ đang trỏ tới --> **con trỏ hằng**

int test(**const** int \*a)

{

\*a = 100; // câu lệnh thay đổi giá trị này sẽ bị lỗi

}

===================

// cho ham chi duoc phep doc gia tri cua mang, nhung KHONG duoc phep thay doi gia tri cac phan tu

int compare\_arrays(**const** int arr1[], **const** int arr2[], int size)

{

int i = 0;

for (i = 0; i < size; ++i){

if (arr1[i] != arr2[i])

return 0;

}

/\* if we got here, both arrays are identical \*/

return 1;

}

================================

**Con trỏ hằng** = con trỏ chỉ được trỏ đến 1 vùng nhớ (địa chỉ ô nhớ), nó sẽ KHÔNG thể trỏ sang vùng nhớ khác (VD. Tên mảng). Bạn vẫn thay đổi giá trị vùng nhớ mà con trỏ đang trỏ tới bình thường!

int const \*a = &x;

**Hằng con trỏ** = con trỏ trỏ tới 1 giá trị vùng nhớ mà không thể thay đổi được, con trỏ nó có thể chuyển sang vùng nhớ khác được (hay dùng trong trường hợp chỉ cho phép vùng nhớ được đọc giá trị)

const int \*a;

=====================

**Bài tập 2**: Xây dựng chương trình nhập vào 1 xâu ký tự từ bàn phím, in ra các từ của xâu này với mỗi từ trên 1 dòng. (chỉ in ra, KO cần tách và lưu lại từ)  
Nếu có nhiều dấu cách liên tiếp cũng chỉ được xuống dòng 1 lần!

VD.

|  |  |
| --- | --- |
| Đầu vào | Đầu ra |
| “Hello World” | Hello World |
| “Nguyen Van Tuan, Va Tran Van Dung” | Nguyen Van Tuan, Va Tran Van Dung |
| “23 343 34.4 ” | 23 343 34.4 |

===========================================

**Con trỏ**

* LÀ kiểu biến đặc biệt, lưu trữ địa chỉ ô nhớ
* Trên 1 hệ thống cố định, kích thước địa chỉ ô nhớ (số bit để đánh địa chỉ ô nhớ) là như nhau --> các con trỏ cùng kích thước (trên cùng 1 hệ thống)
* Kiểu dữ liệu của con trỏ dùng để xác định phạm vi truy cập của con trỏ (1 con trỏ chỉ có thể truy cập tới vùng nhớ cùng kiểu với kiểu được khai báo) - Trừ con trỏ void
* Chỉ có thể gán giá trị cho con trỏ
  + GÁn qua địa chỉ của 1 biến cùng kiểu
  + Gán qua giá trị 1 con trỏ khác cùng loại
* Toán tử &: toán tử tham chiếu --> để lấy địa chỉ của 1 biến
* Toán tử \*: toán tử giải tham chiếu --> để truy cập vào giá trị vùng nhớ mà con trỏ đó đang trỏ tới

**Con trỏ và hàm (C/C++)**

* Truyền tham số cho hàm là truyền bằng tham trị (bản copy của tham số gốc), trong hàm sẽ KHÔNG thể thay đổi giá trị của tham số ban đầu
* Muốn thay đổi giá trị của tham số trong hàm --> truyền vào là địa chỉ của tham số

int test(int \*a)

{

\*a = 100;

}

* Hàm chỉ có thể trả về 1 giá trị thông qua tên hàm (lệnh return), nếu muốn trả về nhiều hơn 1 giá trị --> phải trả về giá trị qua tham số --> kiểu của tham số bắt buộc là con trỏ

VD. Viết hàm trả về phần nguyên và phần thập phân của 1 số thực

Cần trả về 2 giá trị --> trả về qua tham số

void split(double input, int \*phannguyen, double \*thapphan)

=====================

**Tham số cho hàm main trong dòng lệnh**

Int main(int argc, char \*argv[])

======================

int main(int argc, char \*argv[])

{

printf("So tham so %d\n",argc);

printf("Gia tri cac tham so:\n");

for(int i=0; i<argc; i++)

printf("%s\n",argv[i]);

return 0;

}

======================

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[])

{

if(argc!=3) {

printf("Khong du tham so!\n");

return -1;

}

int a = atoi(argv[1]);

int b = atoi(argv[2]);

printf("Tong 2 so la: %d",a+b);

return 0;

}

==================

**Bài tập về nhà**

**Bài 1.**

Viết chương trình có khả năng sinh ra các câu tự động sử dụng kỹ thuật lựa chọn dựa trên số ngẫu nhiên. Chương trình dùng bốn mảng xâu ký tự để lưu trữ các mạo từ (article), danh từ (noun), động từ (verb), giới từ (preposition). Câu được tạo ra bằng cách lựa chọn ngẫu nhiên các phần tử trong các mảng trên và ghép lại theo thứ tự: mạo từ, danh từ, động từ, mạo từ và danh từ. Câu sinh ra cần bắt đầu với kỹ tự in hoa và kết thúc với dấu chấm. Chương trình cần sinh ra tối thiểu 10 câu.

Ví dụ về các phần tử mảng

mạo từ: "the", "a", "one", "some" and "any";

danh từ: "boy", "girl", "dog", "town" and "car";

động từ: verbs "drove", "jumped", "ran", "walked" and "skipped";

giới từ: "to", "from", "over", "under" and "on".

**Bài 2.**

Viết chương trình cho phép người dùng nhập một câu dưới dạng đối số dòng lệnh (mỗi từ trong câu là một đối số). Chương trình hiển thị nội dung câu đảo ngược của câu đã nhập.

VD: ./inverse I love HUST

Cho kết quả: HUST love I