HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

LẬP TRÌNH C CƠ BẢN

Kiểu dữ liệu cơ bản, vào ra file

ONE LOVE. ONE FUTURE.

Nội dung

- Giới thiệu về môn học
- Ôn tập về C
- Mảng
- Xâu ký tự
- Con trỏ
- Tham số dòng lệnh

HUST

hust.edu.vn fb.com/dhbkhn

Nội dung môn học

Giới thiệu về môn học

- Thực hành lập trình theo các chủ đề về cấu trúc dữ liệu và giải thuật
  - Mức độ: cơ sở
  - Cài đặt các cấu trúc dữ liệu – thuật toán
  - Ứng dụng vào giải quyết một số bài toán thực tế.
- Ngôn ngữ lập trình: C

HUST

hust.edu.vn fb.com/dhbkhn

Ôn tập về C

Cú pháp dịch chương trình bằng gcc

- Các tham số:
  - -Wall : bật tất cả các cảnh báo
  - -c: tạo tập tin object
  - -o: tạo tập tin chương trình
  - -g: thêm thông tin gỡ rối
  - -l: sử dụng kèm thư viện

```
> gcc –Wall hello.c –o runhello
> ./runhello
```

Chủ đề

- Ôn tập về các kiểu dữ liệu:
  - mảng,
  - xâu ký tự,
  - con trỏ.
- Xây dựng chương trình với đối số dòng lệnh
- Thao tác với tập tin văn bản
  - Đọc ghi tập tin theo từng ký tự
  - Đọc ghi tập tin theo từng dòng
  - Đọc ghi tập tin theo đặc tả định dạng
- Các bài tập lập trình

HUST

hust.edu.vn fb.com/dhbkhn

Mảng - Array

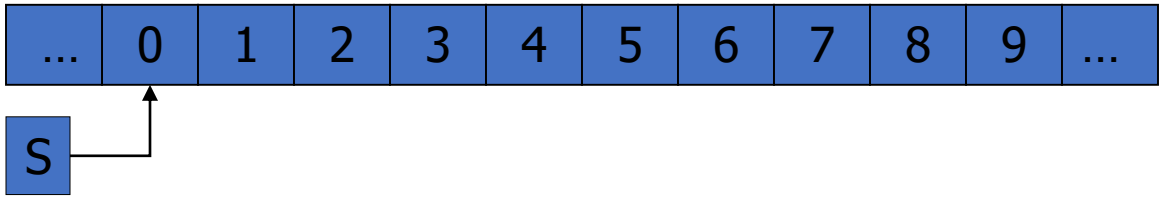
I. Mảng

- Khối liên tục các biến cùng kiểu dữ liệu, cùng tên
- Mảng có thể khai báo với bất cứ kiểu dữ liệu nào
  - VD: `int A[10];` là khai báo mảng 10 số nguyên.
- Các ví dụ về sử dụng mảng
  - Danh sách điểm của các sinh viên
  - Dãy số nhập từ người dùng
  - Véc tơ
  - Ma trận

Mảng trong bộ nhớ trong

- Dãy liên tục các biến cùng kiểu, nằm kế tiếp nhau
- Tên mảng chứa địa chỉ bộ nhớ của ô nhớ đầu tiên
- Ví dụ:

  double S[10];



- Phần tử thứ k của mảng A truy cập qua chỉ số: A[k-1] **(bắt đầu từ 0)**

Ví dụ - hiển thị mảng theo chiều ngược

Ví dụ 1.
Viết chương trình nhập vào 1 mảng 10 số thực từ bàn phím và in ra các giá trị vừa nhập theo thứ tự đảo ngược.
Các giá trị in ra lấy tới 4 chữ số thập phân

```c
#include <stdio.h>

int main(void)
{
    double i, A[10];

    printf("please enter 10 numbers:\n");
    for (i = 0; i < 10; i++)
        scanf("%f", &A[i]);

    printf("numbers in reversed order:\n");
    for (i = 9; i >= 0; i--)
        printf("%0.4f\n", A[i]);

    return 0;
}
```

Bài tập ứng dụng

- Bài tập 1. Với danh sách các phần tử đã nhập trong Ví dụ 1, hãy in ra (các) phần tử có giá trị gần nhất với giá trị trung bình các phần tử của dãy.

Gợi ý

1. Tính giá trị trung bình dãy – average
2. Duyệt dãy, so sánh giá trị lệch của các phần tử với giá trị trung bình average
   - Ghi nhận giá trị chênh lệch nhỏ nhất là mindiff
3. Duyệt lại dãy lần nữa, in ra các phần tử có giá trị chênh lệch so với average bằng mindiff

Chú ý:

Mảng số thực có sự sai số do làm tròn nên nếu cần chuyển về số nguyên và chấp nhận sai số

0.2 + 0.1 Không bằng 0.3

Bài tập ứng dụng

- Bài tập 2. Viết chương trình nhập vào 1 xâu ký tự chỉ gồm các ký tự chữ thường.
  Hãy in ra tần số xuất hiện của các ký tự chữ cái trong xâu, bỏ qua các ký tự không phải chữ cái

Kết quả hiển thị với xâu đầu vào: “hello, world!”

```
The letter 'd' appears 1 time(s).
The letter 'e' appears 1 time(s).
The letter 'h' appears 1 time(s).
The letter 'l' appears 3 time(s).
The letter 'o' appears 2 time(s).
The letter 'r' appears 1 time(s).
The letter 'w' appears 1 time(s).
```

Gợi ý:

- Dùng mảng để đếm tần số xuất hiện các ký tự chữ cái
- Có 26 ký tự chữ cái thường, mảng int count[26]
- Ký tự ‘a’ tương ứng với phần tử đầu mảng chỉ số 0
- Đọc vào xâu bằng cách đọc vào từng ký tự như mảng hoặc dùng hàm có sẵn trong thư viện string

Bài tập ứng dụng

```c
// maximum size 26
#define ALPHABET_LEN 26

int main(void)
{
    int i = 0, count[ALPHABET_LEN] = { 0 }; // initialize all its elements to zero
    char c = '\0';

    printf("Please enter a line of text: \n");

    /* Read in letter by letter and update the count array */
    c = getchar();
    while (c != '\n' && c >= 0) {
        if (c <= 'z' && c >= 'a')
            ++count[c - 'a']; if (c <= 'Z' && c >= 'A')
            ++count[c - 'A'];
        c = getchar();
    }
    for (i = 0; i < ALPHABET_LEN; ++i) {
        if (count[i] > 0)
            printf("The letter '%c' appears %d time(s).\n", 'a' + i, count[i]);
    }
    return 0;
}
```

Bài tập ứng dụng

```c
#include <stdio.h>
#include <string.h>
#define ALPHABET_LEN 26

int main() {
    int i = 0, count[ALPHABET_LEN] = { 0 }; // initialize all its elements to zero
    char s[20], c = '\0';
    printf("Please enter a line of text: \n");
    gets(s);
    for (i = 0; i < strlen(s); i++) {
        c = s[i];
        if (c <= 'z' && c >= 'a') {
            ++count[c - 'a'];
        }
    }
    for (i = 0; i < ALPHABET_LEN; ++i) {
        if (count[i] > 0) {
            printf("The letter '%c' appears %d time(s).\n", 'a' + i, count[i]);
        }

    }
    return 0;
}
```

Bài tập ứng dụng

- Bài tập 3. Viết chương trình dạng hàm
- Hàm so sánh 2 mảng (có cùng số lượng phần tử) xem có trùng nhau (các phần tử ở cùng vị trí giống nhau). Nếu 2 mảng trùng nhau thì trả về giá trị 1, và 0 nếu ngược lại.
- Hàm kiểm tra 2 mảng (có cùng số lượng phần tử) có trùng nhau (mở rộng). Hai mảng sẽ chứa các phần tử như nhau, nhưng các phần tử lúc này không cần ở vị trí giống nhau như hàm ở trên .
  VD. Mảng A={2,4,4,5} sẽ trùng với mảng B={4,5,4,2} nhưng không trùng với mảng C={5,5,4,2}.
- Hàm kiểm tra 2 mảng (có cùng số lượng phần tử) các phần tử liên tiếp có cùng thứ tự hay không.
  VD. Mảng A mà A[i]<=A[i+1] thì trong mảng B cũng phải tương tự B[i]<=B[i+1]
  A={1,3,2,7} sẽ cùng thứ tự với B={2,7,3,4}

Bài tập ứng dụng

```c
#include <stdio.h>

#define SIZE 5
/*function check if two arays are identical*/
int compare_arrays(int arr1[], int arr2[], int size)
{
    int i = 0;

    for (i = 0; i < size; ++i) {
        if (arr1[i] != arr2[i])
            return 0;
    }

    return 1;
}
```

```c
int main(void)
{
    int input1[SIZE], input2[SIZE], i;

    printf("Please enter a list of %d integers:\n", SIZE);
    for (i = 0; i < SIZE; ++i)scanf("%d", &input1[i]);

    printf("Please enter another list of %d integers:\n", SIZE);
    for (i = 0; i < SIZE; ++i) scanf("%d", &input2[i]);

    if (compare_arrays(input1, input2, SIZE) == 1)
        printf("Both lists are identical!\n");
    else
        printf("The lists are not identical...\n");

    return 0;
}
```

Lab test

- **Lab 01**. Find all perfect square in a sequence
  - Given a sequence of n integers a1, a2, . . ., an. Compute the number (Q) of perfect squares in that sequence.
- Input
  - Line 1: contains a positive integer n (1 <= n <= 100000)
  - Line 2: contains n positive integer a1, a2, . . ., an (1 <= ai <= 1000000)
- Output
  - Write the value Q

Input	Output
5 3 2 4 7 9	2

Lab test

- Lab 02. Sum Array
  - Given a sequence of integers a1, a2, ..., an. Compute the sum (Q) of elements of this sequence.
- Input
  - Line 1: contains n (1 <= n <= 10000)
  - Line 2: contains a1, a2, ..., an (-10000 <= ai <= 10000)
- Output
  - Write the value of Q

Input	Output
4 3 2 5 4	14

HUST

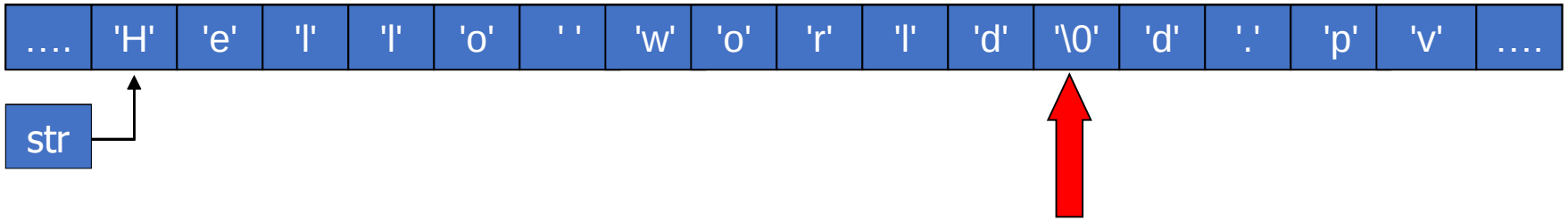
hust.edu.vn fb.com/dhbkhn

Xâu ký tự - String

Xâu ký tự trong C

- Mảng các ký tự.
- Kiểu dữ liệu chính để lưu trữ văn bản.
- Có thể khởi tạo khi khai báo:

```
char str[] = “Hello world";
```



- Để lưu trữ xâu độ dài N ta cần khai báo mảng N+1 phần tử
- Hai cách khởi tạo sau là tương đương

```
char str[] = { 'b', 'l', 'a', 'b', 'l', 'a', '\0' };
char str[] = "blabla";
```

Các hàm nhập xâu ký tự và ký tự

- getchar()
  - c = getchar()
- scanf
  - scanf("%s", str);
- gets()
  - gets(str);

- strlen(const char s[])
  trả về số ký tự của xâu s (không tính ký tự NULL)
- strcmp(const char s1[], const char s2[])
  so sánh s1 với s2
- strcpy(char s1[], const char s2[])
  sao chép nội dung s2 vào s1
- strcat(char s1[], char s2[])
  nối s2 vào s1, sau đó lưu kết quả vào s1

Ví dụ

- Ví dụ 1. Xây dựng hàm thay thế ký tự trong xâu
  - Hàm có tham số là một xâu ký tự và hai ký tự
  - Hàm sẽ duyệt xâu và thay thế tất cả các ký tự thứ nhất trong xâu bằng ký tự thứ hai.
- Viết chương trình để kiểm tra hàm nói trên:
  - Đọc một xâu không chứa ký tự trắng và hai ký tự, sau đó gọi hàm với các đối số trên và in ra kết quả.
- Ví dụ
  - Đầu vào: “papa”, ‘p’, ‘m’
  - Kết quả: “mama”

Ví dụ 1

```
/* function that replace all charaters <replace_what> by
<replace_with> in string str */
void replace(char str[], char replace_what, char replace_with)
{
    int i;

    for (i = 0; str[i] != '\0'; ++i)
    {
        if (str[i] == replace_what)
            str[i] = replace_with;
    }
}
```

Ví dụ 1

```c
#define STRING_LEN 100

int main(void)
{
    char str[STRING_LEN + 1];
    char replace_what, replace_with, tmp;

    printf("Please enter a string (no spaces)\n");
    scanf("%100s", str);

    printf("Letter to replace: ");
    scanf(" %c", &replace_what);
    do { tmp = getchar(); } while (tmp != '\n');

    printf("Letter to replace with: ");
    scanf(" %c", &replace_with);

    replace(str, replace_what, replace_with);
    printf("The result: %s\n", str);
    return 0;
}
```

Bài tập

- Bài tập 4. Viết chương trình đọc một xâu ký tự biểu diễn một câu từ người dùng. Sau đó chương trình hiển thị mỗi từ trong câu trên một dòng. Một từ là một dãy các ký tự liên tiếp không chứa ký tự trắng.
- Ví dụ:
  - Đầu vào: "The house nextdoor is   very old. "
  - Kết quả:
    - The
    - house
    - ….

Bài tập

- Bài tập 5. Viết chương trình yêu cầu người dùng nhập số lượng sinh viên trong một lớp học, sau đó nhập tên đầy đủ bằng tiếng Việt của mỗi sinh viên. Hiển thị danh sách sinh viên sắp xếp theo tên sinh viên. Ví dụ:
  - Nguyen Bao Anh
  - Tran Quang Binh
  - Vuong Quoc Binh
  - Dao Thi Ha
  - Ngo Anh Vu
- Nâng cao (không bắt buộc): Hiển thị số lượng lớn nhất các sinh viên cùng tên.

Lab test

- Lab 03. Count words
  - Given a Text, write a program to count the number of words (ignore characters SPACE, TAB, LineBreak) of this Text
- Input
  - The Text
- Output
  - Write the number of words

Input	Output
Hanoi University Of Science and Technology School of Information and Communication Technology	12

Lab test

- Lab 04. Text Replacement
  - Cho văn bản T và 2 mẫu P1, P2 đều là các xâu ký tự (không chứa ký tự xuống dòng, độ dài không vượt quá 1000). Hãy thay thế các xâu P1 trong T bằng xâu P2.
- Dữ liệu
  - Dòng 1: xâu P1
  - Dòng 2: xâu P2
  - Dòng 3: văn bản T
- Kết quả:
  - Ghi văn bản T sau khi thay thế
- Ví dụ

Input	Output
AI Artificial Intelligence Recently, AI is a key technology. AI enable efficient operations in many fields.	Recently, Artificial Intelligence is a key technology. Artificial Intelligence enable efficient operations in many fields.

HUST

hust.edu.vn fb.com/dhbkhn

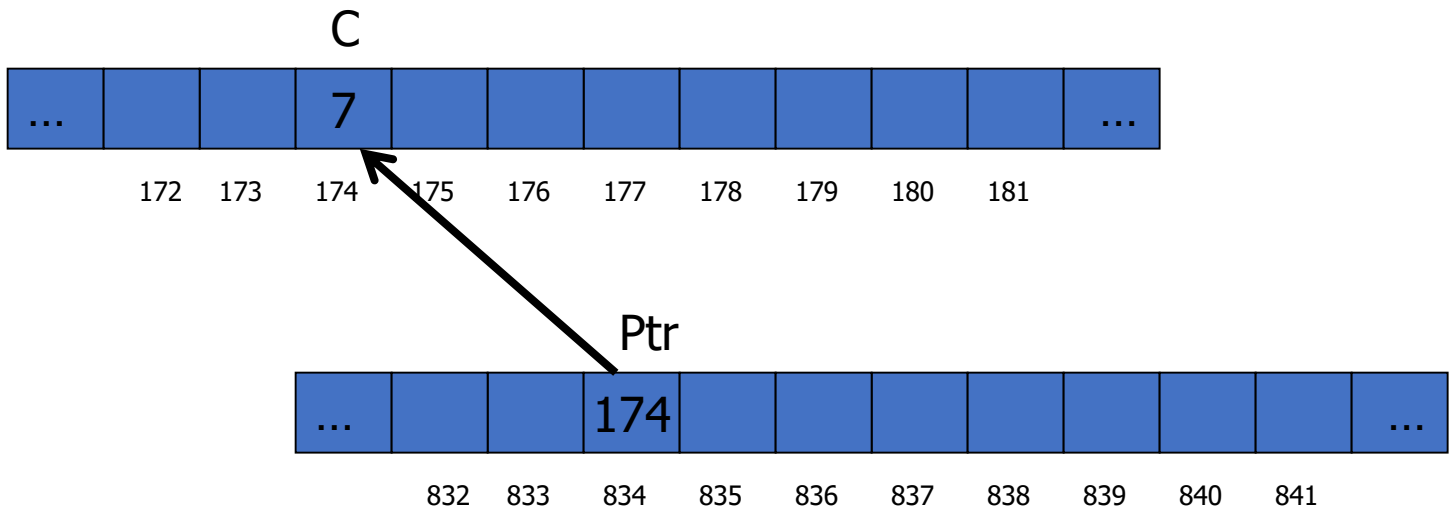
Con trỏ - pointer

Con trỏ

- Con trỏ là biến dùng để lưu giá trị một địa chỉ bộ nhớ.
- Địa chỉ của biến hoặc mảng.
- Được khai báo với ký tự * trước tên.

```
Kiểu_dữ_liệu *tên_biến;
```

- Con trỏ ptr được gọi là "trỏ" *point* tới biến c nếu giá trị của nó là địa chỉ của c



Toán tử tham chiếu (reference) và giải tham chiếu (dereference)

```
int n;
int* iptr; /* khai báo P là một con trỏ kiểu int */
n = 7;
iptr = &n;

printf(" % d", *iptr); /* Hiển thị ‘7’*/
*iptr = 177;
printf("% d", n); /* Hiển thị ‘177’ */
iptr = 177; /* Phép gán không đúng!! */
```

Ví dụ

Ví dụ 1. Viết hàm nhận đối số là một số thực (double) và trả về phần nguyên và phần thập phân của số đó.

Viết chương trình minh họa hàm trên, với số thực được nhập từ người dùng.

```
void split(double num, int* int_part,
double* frac_part)
{
    *int_part = (int)num;
    *frac_part = num - *int_part;
}
```

Ví dụ

```c
int main(void)
{
    double num, fraction;
    int integer;

    printf("Please enter a real number: ");
    scanf("%f", &num);

    split(num, &integer, &fraction);
    printf("The integer part is %d\n", integer);
    printf("The remaining fraction is %f\n", fraction);

    return 0;
}
```

Ví dụ

- Ví dụ 2. Viết hàm thay thế ký tự trong xâu với nguyên mẫu hàm:

void replace_char(char *str, char c1, char c2);

- Hàm sẽ thay thế tất cả các ký tự **c1** xuất hiện bởi **c2** trong xâu **str.**
- Minh họa việc sử dụng hàm

```
void replace_char(char* str, char c1, char c2)
{
    if (str == NULL)
        return;

    while (*str != '\0') {
        if (*str == c1) {
            *str = c2;
        }
        ++str;
    }
}
```

Bài tập

- Bài tập 6. 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 ký 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".

HUST

hust.edu.vn fb.com/dhbkhn

Tham số dòng lệnh

Chương trình với đối số dòng lệnh

- Các đối số dòng lệnh được định nghĩa trong hàm **main**
  - main bản thân là một hàm như các hàm khác
  - Nó có thể nhận các đối số truyền vào (từ dòng lệnh gõ bởi người dùng)
  - Vai trò của chương trình gọi nó (the calling function) trong trường hợp này là hệ điều hành, hoặc một chương trình khác

Nguyên mẫu của hàm main

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

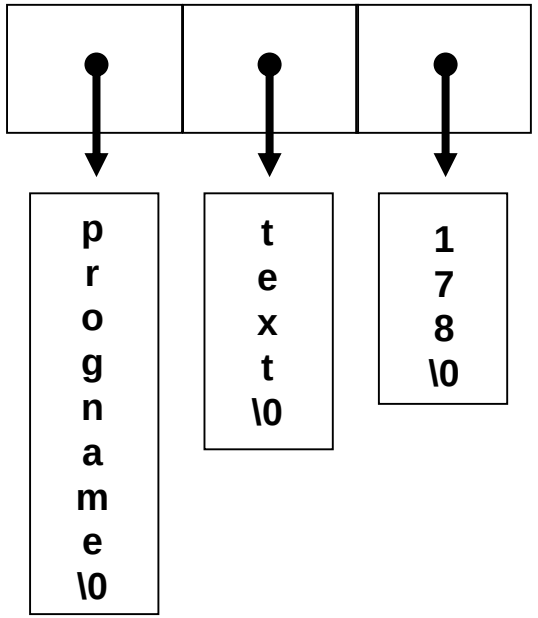
- Khi chúng ta muốn chương trình nhận các đối số tại dòng lệnh, cần định nghĩa hàm main như trên, với:
  - **argc** chứa số lượng các đối số
  - **argv** là một mảng các con trỏ kiểu char – mảng các xâu ký tự – nhận và lưu trữ các giá trị của các đối số dưới dạng dữ liệu văn bản.
- Đối số đầu tiên mặc định luôn là tên của chương trình.

Nguyên mẫu của hàm main

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

argc : 3

argv :



Các bước viết chương trình nhận đối số dòng lệnh

- Viết đúng cú pháp cho hàm **main**
- Kiểm tra giá trị **argc** để đảm bảo người dùng nhập đúng số đối số. Nếu sai, hiển thị thông báo cùng hướng dẫn về cú pháp sử dụng chương trình.
  - VD: Hello Hanoi ➔ 1 đối số thực sự, giá trị của argc nên là : 1+1 =2
- Lấy giá trị của các đối số từ mảng argv[], bắt đầu từ argv[1] và chuyển đổi sang đúng kiểu dữ liệu khi cần thiết.
  - Sử dụng các hàm atoi, atol, atof trong thư viện <stdlib.h>
- Xử lý, tính toán trên các đối số nói trên.

Ví dụ

- Ví dụ 1. Viết chương trình nhận hai số thực làm đối số dòng lệnh, đại diện cho chiều dài và chiều rộng của một hình chữ nhật.
- Chương trình dựa trên đó tính toán và in ra diện tích và chu vi của hình chữ nhật.

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    double width, height;
    if (argc != 3){
        printf("Wrong number of arguments!\n");
        printf("CORRECT SYNTAX : RECT <WIDTH> <HEIGHT>\n");
        return 1;
    }
    width = atof(argv[1]);
    height = atof(argv[2]);
    printf("The rectangle's area is %f\n", width* height);
    printf("The rectangle's perimeter is %f\n", 2 * (width + height));
    return 0;
}
```

Bài tập

- Bài tập 7. Đảo ngược câu
- 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

Bài tập

- Bài tập 8. Viết chương trình có tên sde nhận đối số dòng lệnh là các hệ số của một phương trình bậc 2 $ax^2 + bx + c = 0$ và giải phương trình, in ra màn hình các nghiệm.
- Cú pháp sử dụng sde a b c
- Ví dụ: ./sde 1 2 1 cho kết quả: x1=x2 =-1