



Phần 7

Nội dung thực hành

- Xâu kí tự: *khai báo biến xâu kí tự, một số thuật toán cơ bản trên xâu kí tự*
- Thao tác trên xâu kí tự: *tách, ghép, tìm kiếm, xóa, thay thế, đếm, chuyển đổi*
- Vận dụng một số hàm thư viện chuyên dụng để xử lý xâu kí tự: *gets, puts, strlen, strcpy, strcat, strcmp, tolower, toupper, strchr, strstr, strncat, ...*

Ví dụ

1. Các cách khai báo và in ra xâu ký tự

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

int main() {
    char xau_1[] = {'K', 'H', 'T', 'N', '\0'};
    char xau_2[] = "KHTN";
    char *xau_3 = "KHTN";

    printf("%s\n", xau_1);
    puts(xau_2);
    int i, L = strlen(xau_3);
    for (i = 0; i < L; i++) putchar(xau_3[i]);

    return 0;
}
```

2. Hàm copy và nối xâu: *strcpy* và *strcat*

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

int main () {
    char xau_1[] = "Bo mon";
    char xau_2[] = " Tin hoc Vat ly";
    char xau_3[100];

    strcpy (xau_3, xau_1);
    puts(xau_3);
    strcat (xau_3, xau_2);
    puts(xau_3);

    return 0;
}
```

3. Trả lời câu đố: “Ai là cha đẻ của ngôn ngữ lập trình C?”

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

int main() {
    char dapan[] = "dennis m. ritchie";

    // Viet hoa chu cai dau cua cac tu trong ten
    // Cach 1:
    int i, L = strlen(dapan);
    for (i = 0; i < L; i++) {
        if (i == 0 || dapan[i - 1] == ' ') {
            dapan[i] = toupper(dapan[i]);
        }
    }

    // Cach 2:
    char *ctr = dapan - 1;
    while (ctr != NULL) {
        int d = ctr - dapan + 1;
        *(dapan + d) = toupper(*(ctr + 1));
        ctr = strchr(ctr + 1, ' ');
    }

    char traloi[100];

    do {
        puts("Ai la cha de cua ngon ngu lap trinh C?");
        gets(traloi);
    } while (strcmp(traloi, dapan) != 0);

    puts("Chính xác!");

    return 0;
}
```

🔖 Lưu ý:

- Xâu ký tự, về bản chất, là một mảng mà mỗi phần tử của mảng là một ký tự (kiểu *char*) thay vì một số (kiểu *int* hoặc *float*) như ta vẫn thường sử dụng.
- Ký tự kết thúc của một xâu là ký tự đặc biệt: `'\0'`. Nó không được tính vào số ký tự có trong xâu khi dùng hàm *strlen* để kiểm tra.

Bài tập

1. Cho một xâu ký tự được tạo thành bởi các chữ cái trong bảng chữ cái tiếng Anh. Tính xem có nhiều loại chữ cái xuất hiện trong xâu, và mỗi loại xuất hiện bao nhiêu lần. (Viết hoa hay viết thường đều tính là 1 loại)

Input

1 xâu ký tự

Output

dòng 1 là số n là số loại chữ cái xuất hiện trong xâu

n dòng tiếp, mỗi dòng chứa 1 chữ cái và số lần xuất hiện

Input	Output
DhKHtNdHQGHn	7
	d 2
	g 1
	h 4
	k 1
	n 2
	q 1
	t 1

🔖 Gợi ý (một trong nhiều cách): Sắp xếp các ký tự trong xâu.

2. Một xâu palindrome là một xâu mà đọc xuôi (từ trái sang phải) hoặc ngược (từ phải sang trái) đều giống nhau, chẳng hạn như “radar”, “Hannah” hay “madam”. Cho một xâu ký tự (viết in thường). Lập một hàm kiểm tra xem xâu này có phải là xâu palindrome không. Hàm này trả về 2 giá trị: 0 hoặc 1, tương ứng với xâu đã cho “không” hoặc “có” là xâu palindrome.

Input

1 xâu ký tự

Output

nếu hàm trả về 1, in ra “YES”, còn lại, in ra “NO”

Input	Output
kayak	YES

🔖 Gợi ý: Một số xâu palindrome: *tacocat*, *racecar*, *neveroddoreven*, *steponnopets*, *ufotofu*.

3. Xâu Fibonacci được định nghĩa như sau:

$$F_n = \begin{cases} 'A' & \text{nếu } n = 0 \\ 'B' & \text{nếu } n = 1 \\ F_{n-1} \text{ nối với } F_{n-2} & \text{nếu } n > 1 \end{cases}$$

Cho số n . Tính xâu Fibonacci thứ n .

Input

1 số nguyên n

Output

Xâu Fibonacci F_n

Input	Output
6	BABBABABBABBA

Giải thích:

$F_0 = 'A'$, $F_1 = 'B'$, $F_2 = 'BA'$, $F_3 = 'BAB'$, $F_4 = 'BABBA'$, $F_5 = 'BABBABAB'$, $F_6 = 'BABBABABBABBA'$

4. Phép chuẩn hóa xâu bao gồm những thao tác sau:

- Xóa hết ký tự trống ở đầu xâu
- Nếu trong xâu có hai ký tự trống liên tiếp nhau thì xóa bớt đi một
- Xóa hết ký tự trống ở cuối xâu

Cho một xâu ký tự. Thực hiện phép chuẩn hóa trên xâu đã cho.

Input

1 xâu ký tự

Output

xâu ký tự sau khi đã chuẩn hóa

Input	Output
␣Bo␣mon␣Tin␣␣hoc␣Vat␣␣uly␣	Bo␣mon␣Tin␣hoc␣Vat␣uly

📌 Gợi ý:

- + Xem lại slides bài giảng về *Xâu ký tự*.
- + Nên viết phép chuẩn hóa xâu thành một hàm để tiện cho việc xử lý xâu ở các bài sau.

5. Cho một xâu ký tự bao gồm các từ. Tách các từ của xâu ban đầu ra rồi lưu lại vào một mảng các xâu, sau đó in ra các từ đã lưu.

Input

1 xâu ký tự

Output

mỗi dòng in ra 1 từ của xâu ban đầu

Input	Output
Tin hoc Vat ly	Tin hoc Vat ly

6. Cho một xâu ký tự bao gồm các từ. Cho số n . Xóa từ thứ n của xâu đã cho.

Input

dòng 1 là 1 xâu ký tự

dòng 2 là số n

Output

xâu ký tự sau khi đã xóa từ thứ n

Input	Output
Tin hoc Vat ly 2	Tin Vat ly

📌 Mở rộng bài toán: Tương tự, cho một từ, hãy đếm số lần xuất hiện của từ đó trong xâu.

7. Cho một xâu ký tự mô tả một đoạn văn bao gồm các câu (ngăn cách nhau bởi các dấu ngắt câu thông thường: ‘,’ , ‘?’ , ‘!’). Tách các câu của xâu ban đầu ra rồi lưu lại vào một mảng các xâu, sau đó in ra các câu đã lưu.

Input

1 xâu ký tự

Output

mỗi dòng in ra 1 câu của xâu ban đầu

Input	Output
You enter the high school lab. You see an experiment. How will you know which class is it? If it's green and wiggles, it's biology. If it stinks, it's chemistry. If it doesn't work, it's physics!	You enter the high school lab. You see an experiment. How will you know which class is it? If it's green and wiggles, it's biology. If it stinks, it's chemistry. If it doesn't work, it's physics!

*Xâu ký tự ở ví dụ *Input* không có dấu xuống dòng.

8. Cho n xâu ký tự ứng với n cái tên. Sắp xếp lại n tên này theo chiều tăng dần của bảng chữ cái.

Input

dòng 1 là số n

n dòng tiếp, mỗi dòng chứa 1 xâu

Output

n dòng chứa n cái tên sau khi đã sắp xếp

Input	Output
5 Vy Huy Anh Minh Khang	Anh Huy Khang Minh Vy

📌 Gợi ý: Nhớ lại bài toán sắp xếp đã học, kết hợp với hàm *strcmp* của thư viện *string* để so sánh hai xâu.

9. Một xâu X là xâu con (substring) của một xâu Y nếu X là một chuỗi các ký tự liên tiếp của Y . Cho một xâu ký tự Y (viết in thường). Tìm xâu ký tự con palindrome X lớn nhất của Y .

Input

1 xâu ký tự Y

Output

xâu ký tự con palindrome X lớn nhất của Y

Giải thích: Xâu đã cho có nhiều xâu palindrome con như: “ss”, “ippi”, “ssiss”, ... nhưng xâu “ississi” là dài nhất.

Input	Output
Mississippi	ississi

📌 Gợi ý: Áp dụng hàm đã lập ở bài 2.

Mã Morse ([Morse code](#)) là một phương pháp được sử dụng trong viễn thông để mã hóa các ký tự văn bản dưới dạng chuỗi chuẩn hóa của hai loại tín hiệu khác nhau là *dit* (dấu chấm) và *dah* (dấu gạch ngang). Không có sự phân biệt giữa chữ hoa và chữ thường. Mỗi ký tự được mã hóa một chuỗi các *dit* và *dah*. Thời lượng một *dah* gấp ba lần thời lượng một *dit*. Mỗi *dit* hoặc *dah* trong một ký tự mã hóa được theo sau bởi một khoảng thời gian không có tín hiệu, được gọi là khoảng trắng, bằng thời lượng *dit*. Các chữ cái của một từ được phân tách bằng một khoảng trắng có thời lượng bằng ba *dit*. Các từ được phân tách bằng một khoảng trắng bằng bảy *dit*.

A	•—	K	—••	U	••—	1	•—•—•—
B	••••	L	—•••	V	•••—	2	••—•—•—
C	••—•	M	—•—	W	•—•—	3	••••—•—
D	••••	N	—••	X	—•••	4	•••••—
E	•	O	—•—•	Y	•—•—•	5	•••••
F	••••	P	•—•••	Z	—•—••	6	•—••••
G	•—••	Q	—••—•			7	—•—•••
H	••••	R	•—••			8	—•—•••
I	••	S	•••			9	—•—•—••
J	•—•—•	T	—•			0	—•—•—•—

10. Cho một xâu ký tự. Chuyển xâu ký tự này thành mã Morse rồi in ra.
Để thuận tiện cho việc in ra, ta quy ước: *dit* là dấu ‘.’, *dah* là dấu ‘-’. Mỗi *dit* hoặc *dah* của một ký tự không cần khoảng trắng theo sau. Các ký tự cách nhau bởi 1 dấu cách. Các từ cách nhau bởi 3 dấu cách.

Input

1 xâu ký tự

Output

mã Morse của xâu ký tự đã cho

Input	Output
SOS	...- - - - -

Input	Output
I love HUS	..- - - - - - - - - - - - - - - -

11. Cho một xâu ký tự mô tả mã Morse. Giải mã mã Morse này thành xâu ký tự rồi in ra.
(Quy ước *Input* giống như quy ước *Output* của bài trước)

Input

1 xâu ký tự mã Morse

Output

xâu ký tự sau khi đã giải mã

Input	Output
...- - - - -	SOS

Input	Output
..- - - - - - - - - - - - - - - -	I love HUS

12. Cho hai xâu ký tự mô tả hai số dương a và b . Tính tổng $a + b$.

Input

dòng 1 là xâu ký tự mô tả số a

dòng 2 là xâu ký tự mô tả số b

Output

xâu ký tự mô tả tổng $a + b$

Input	Output
987654321 876543210	1864197531

Input	Output
99999888887777766666555544444 9998979695949392919089888786858483	9999979694838270696756554342402927

🔗 Gợi ý:

+ `int d = '7' - '0'; // d = 7`
+ `char c = 8 + '0'; // c = '8'`

🔗 Mở rộng bài toán:

+ Ngoài phép [cộng](#), tìm cách thực hiện các phép toán tương tự: [trừ](#), [nhân](#), và [chia](#).
+ Để tăng tốc độ thực hiện cũng như tiết kiệm bộ nhớ, sử dụng cơ số 1,000,000,000 thay vì cơ số 10.
+ Tham khảo thêm các [thuật toán nhân](#) khác như thuật toán [Karatsuba](#).

13. Cho một xâu ký tự mô tả một công thức toán học ([formula](#)). Biết rằng xâu này chỉ chứa các ký tự số, dấu cách, dấu ‘=’, dấu của các phép toán cơ bản như ‘+’, ‘-’, ‘*’, ‘/’, hãy xác định xem công thức toán học đã cho là đúng hay sai bằng cách in ra “TRUE” hoặc “FALSE” tương ứng.

Input

1 chuỗi ký tự mô tả biểu thức

Output

in ra câu trả lời: "TRUE" hoặc "FALSE"

Input	Output
$-3 + 5 * 4 = 17$	TRUE

Input	Output
$17 + 3 / 4 = 5$	FALSE

🔖 Mở rộng bài toán:

- + Trong trường hợp chuỗi mô tả một biểu thức ([expression](#)), trả về giá trị biểu thức đó.
- + Xét thêm các phép so sánh khác: '>', '<', '>=', '<=', '!='
- + Xét thêm các ký tự ngoặc: '(' và ')'
- + Xét thêm các phép toán phức tạp khác như: '!', '^', ...

14. Cho một số dương n , $1 \leq n \leq 3888$. Tìm chuỗi ký tự mô tả n dưới dạng số La Mã ([Roman numerals](#)).

Input

1 số n

Output

chuỗi ký tự mô tả n dưới dạng số La Mã

Input	Output
207	CCVII

Input	Output
1918	MCMXVIII

15. Cho một chuỗi ký tự mô tả một số n nào đó dưới dạng số La Mã. Tìm n , biết $1 \leq n \leq 3888$.

Input

1 chuỗi ký tự mô tả n dưới dạng số La Mã

Output

số n

Input	Output
CCVII	207

Input	Output
MCMXVIII	1918

16. Giả sử có một hóa chất chỉ gồm các nguyên tố C, H, N, và O có nguyên tử khối là 12, 1, 14, và 16 (u) tương ứng. Cho một chuỗi mô tả công thức cấu tạo hóa học ([condensed structure formula](#)) của hóa chất. Tính khối lượng của hóa chất đã cho.

Input

1 chuỗi ký tự mô tả hóa chất

Output

1 số là khối lượng hóa chất

Input	Output
(CH ₃) ₂ CHOH	60

Input	Output
(CH ₃) ₂ CHCH ₂ CO ₂ H	102

Input	Output
((CH) ₂ (OH ₂ H) (C(H)) O) ₃	222

Input	Output
CH ₃ CH ₂ CCCH ₂ CO (CH ₂) ₂ N (CH ₃) ₂	167

Input	Output
CH ₃ (CH ₂) ₂ CONHCH ₂ CH (CH ₃) OCH ₃	159

17. Hãy xem xét chương trình sau:

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

int main(int argc, char *argv[]) {
    char *string1;
    string1 = "hello";
    printf("string1 %s\n", string1);
    printf("string1[2] %c\n", string1[2]);
    string1[2] = 'L';
    printf("string1 %s\n", string1);
    return 0;
}
```

```
string1 hello
string1[2] l
[1] 581260 segmentation fault (core dumped)
```

Chương trình này ra kết quả như trên (chạy bị lỗi) hoặc là không thực thi dòng lệnh `printf` cuối cùng. Hỏi rằng chương trình bị sai ở đâu? Tại sao? Ta cần sửa lại chương trình thế nào cho đúng?