



BÀI THỰC HÀNH SỐ 12

1. MỤC TIÊU

Hiểu và cài đặt được các hàm xử lý trên kiểu dữ liệu chuỗi

2. LÝ THUYẾT CẦN GHI NHỚ

2.1. Ký tự (character)

2.1.1. Khái niệm

- Ký tự “ in được ” gồm :
 - 26 chữ thường ('a', 'b', 'c', ..., 'z'),
 - 26 chữ hoa ('A', 'B', 'C', ..., 'Z'),
 - 10 chữ số ('0', '1', '2', '3', ..., '9'),
 - Khoảng trắng, các ký tự: ! “ # \$ % & ‘ () * + , - . / : ; < = > ? @ [\] ^ _ { | } ~
- Các ký tự “ không in được ”: tab, lert (bell), newline, formfeed, ...
- Các ký tự “ in được ” đặc biệt: '\\', '\'', '\"' , ...

2.1.2. Nhập ký tự

(i). Hàm *getch*

- Nhận một ký tự từ bộ đệm bàn phím và không cho hiện ký tự này lên màn hình.
- Cú pháp: `int getch (void)`
- Ví dụ

```
char ch;
printf("Nhap 1 ky tu: ");
ch = getch();
printf("Ma ASCII cua ky tu %c la %d", ch,ch);
```
- Hàm trả về ký tự nhận được.
 - Nếu ký tự có sẵn trong bộ đệm bàn phím thì hàm nhận một ký tự trong đó.
 - Nếu bộ đệm rỗng thì chương trình tạm dừng cho đến khi ta gõ vào một ký tự. Ký tự gõ vào sẽ nhận được ngay, không cần phải gõ phím Enter và ký tự vừa nhập không được hiển thị lên màn hình.

(ii). Hàm *getche*

- Nhận một ký tự từ bộ đệm bàn phím (tương tự như hàm *getch*) và cho hiển thị ký tự này lên màn hình.
- Cú pháp: `int getche(void)`
- Ví dụ



```
printf("Nhap 1 ky tu: ");  
int ch = getche();  
printf("Ma ASCII cua ky tu %c la %d", ch,ch);
```

(iii). **Hàm getchar**

- Hàm cho chương trình dừng lại và chờ người dùng nhập 1 ký tự và nhấn enter.
- Cú pháp: `int getchar(void)`
- Ví dụ

```
printf("Nhap 1 ky tu: ");  
unsigned char ch = getchar();  
printf("Ma ASCII cua ky tu %c la %d", ch,ch);
```

2.1.3. *Xuất ký tự*

(i). **Hàm putchar**

- Khai báo thư viện `<string.h>` khi sử dụng hàm.
- Xuất một ký tự ra cửa sổ văn bản màn hình.
- Cú pháp: `int putchar (int ch)`

Trong đó, đối số `ch` chứa ký tự cần hiển thị.

- Hàm trả về ký tự đã hiển thị và xuất ký tự `ch` lên màn hình.

(ii). **Hàm printf**

- Khai báo thư viện `<string.h>` khi sử dụng hàm
- Cú pháp: `printf ("%c", ch)`
- Hàm có công dụng xuất ký tự `ch` lên màn hình.

(iii). **putc (ch)**

- Khai báo thư viện `<string.h>` khi sử dụng hàm
- Hàm có công dụng xuất ký tự `ch` lên màn hình.

2. 2. Chuỗi

2.2.1. *Khai báo chuỗi*

- Ví dụ

```
char s[30];  
char chuoi [] = "Lap Trinh C"  
char s [50]= "TP.Ho Chi Minh";  
char name []= {'Q','u','a','n',' ','3','\0'};  
char ten[10]={ 'S','a','i',' ','G','o','n','\0'};
```

2.2.2. *Truy xuất từng ký tự trong chuỗi*

Do chuỗi là một mảng ký tự vì vậy ta có thể truy xuất chuỗi bằng chỉ số như truy xuất mảng.



3. BÀI TẬP CÓ HƯỚNG DẪN

Hướng dẫn

Bước 1. Viết các hàm nhập số phần tử của mảng, tạo mảng, xuất mảng tương tự như bài thực hành trước.

Bước 2. Sau đó mới thực hiện cài đặt và gọi hàm vừa xây dựng từ hàm main?

Bài 1. Viết hàm đếm số nguyên âm của chuỗi được nhập từ bàn phím.

```
int DemNguyenAm(char S[])
{
    int dem=0,i=0;
    while (s[i]!='\0')
    {
        switch(s[i])
        {
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U':
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': dem++;
        }
        i++;
    }
    return dem;
}
```

Bài 2. Viết hàm nhận 2 tham số kiểu chuỗi. Copy nội dung chuỗi thứ nhất (sSource), đổi thành chữ hoa rồi đưa vào chuỗi thứ hai (sDest).

```
void DoiThanChuHoa(char sSource[], char sDest[])
{
    int dem=0,i=0;
    char s1[100],s2[100];
    printf("Nhap chuoi");
    gets(s1);
    i=0;
    while (sSource[i] != '\0')
    {
        sDest[i]=toupper(sSource[i]);
        i++;
    }
}
```

Bài 3. Viết hàm nhận 2 tham số kiểu chuỗi (sSource, sDest). Hàm thực hiện đảo ngược chuỗi sSource và đưa vào sDest. Giả sử sSource= “Sai gon” => sDest= “nog iaS”.



```
void DaoChuoi(char sSource[], char sDest[])
{
    int i, l=strlen(sSource);
    for(i=0; i<l; i++)
        sDest[i]=sSource[l-i-1];
    sDest[i]='\0';
}
```

4. BÀI TẬP DO SINH VIÊN TỰ THỰC HIỆN

Bài 1. Chèn thêm giá trị x vào vị trí k trong mảng một chiều nguyên (x, k là tham số đầu vào của hàm).

Bài 2. Nhập chuỗi

Bài 3. Xuất chuỗi

Bài 4. Đếm số ký tự 'a' có trong chuỗi

Bài 5. Đếm khoảng trắng trong chuỗi.

Bài 6. Đếm số từ có trong chuỗi (các từ cách nhau bởi khoảng trắng)

Bài 7. Viết hàm đổi tất cả các ký tự có trong chuỗi thành chữ thường (không dùng hàm `strlwr`)

Mở rộng: đổi tất cả thành chữ hoa/chữ in).

Bài 8. Viết hàm đổi tất cả ký tự đầu thành chữ hoa (chữ in), các ký tự còn lại là chữ thường.

Bài 9. Viết hàm đảo ngược các ký tự trong chuỗi.

Ví dụ: nhập ABCDE, chuỗi sau khi đảo ngược là:EDCBA

Bài 10. Một từ được gọi là *palindrome* khi các ký tự trong từ đối xứng nhau. Một số từ *palindrome* là: rotor, radar, madam, abba, php, www. Viết hàm kiểm tra xem từ có phải là từ đối xứng hay không? Kết quả trả về *true* hoặc *false*.

Mở rộng: Một chuỗi được gọi là *palindrome* khi các ký tự trong chuỗi đối xứng nhau.

Ví dụ: chuỗi sau là chuỗi *palindrome*: ABLE WAS I ERE I SAW ELBA

Bài 11. Viết chương trình cho nhập 1 số nguyên dương n (n>0). Chương trình thực hiện loại bỏ 1 số trong n sao cho giá trị các số còn lại trong n là nhỏ nhất. ¹

Ví dụ: n = 21 \Rightarrow bỏ số 2 \Rightarrow n = 1
 n = 132 \Rightarrow bỏ số 3 \Rightarrow n = 12
 n = 104 \Rightarrow bỏ số 2 \Rightarrow n = 4
 n = 23198 \Rightarrow bỏ số 3 \Rightarrow n = 2198

¹ Trích trong đề thi *ICPC Asia Regionals 2019 Online Preliminary Round* (<https://www.codechef.com/ICPCIN19>)



Bài 12. Giả sử chuỗi s được xem là hợp lệ khi giữa các từ chỉ có đúng 1 khoảng trắng và chuỗi không chứa các ký tự khoảng trắng ở đầu và cuối chuỗi. Viết hàm cắt khoảng trắng thừa đang có trong chuỗi.

Bài 13. Sắp xếp các ký tự trong chuỗi tăng dần.

Bài 14. Nhập 3 chuỗi, xuất 3 chuỗi theo thứ tự từ điển.

Ví dụ: nhập $s1 = "ABE"$, $s2 = "BAC"$, $s3 = "ABC"$. Sẽ in ra theo thứ tự $"ABC"$, $"ABE"$, $"BAC"$.

Bài 15. Cho nhập chuỗi S chỉ chứa các ký số 0 hoặc 1. Cho biết chuỗi có chứa chuỗi con có từ 5 ký tự trở lên giống nhau (cùng bằng 0 hoặc 1).

Ví dụ: $S = "0101000000010111" \Rightarrow$ có

$S = "0101000111101" \Rightarrow$ không

Bài 16. Trong mật mã học, mật mã Caesar (*Caesar Cipher*, còn được gọi là mật mã dịch chuyển - *Shift Cipher*, mã của Caesar hoặc dịch chuyển Caesar), là một trong những kỹ thuật mã hóa đơn giản và được biết đến rộng rãi nhất. Nó là một loại mật mã thay thế, trong đó mỗi chữ cái trong bản rõ (plain text) được thay thế bằng một chữ cái một số vị trí cố định trong bảng chữ cái dựa trên khóa k (key). Ví dụ với dịch chuyển trái 3, A sẽ được thay thế bằng D, B sẽ trở thành E, v.v. Phương pháp này được đặt theo tên của Julius Caesar, người đã sử dụng nó trong thư tín riêng của mình.

Thông điệp được mã hóa bằng cách dịch chuyển xoay vòng từng ký tự đi k vị trí trong bảng chữ cái. Trường hợp với $k=3$ gọi là phương pháp mã hóa Caesar.

Minh họa: Cho plain text = $x_1, x_2, x_3, x_4, x_5, x_6 = 'ATTACK'$, $k=17$, chiều dịch chuyển của ký tự là từ trái sang phải và bảng chữ cái được quy ước đánh số như sau:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

– Thực hiện mã hóa:

$$y_1 = x_1 + k \bmod 26 = 0 + 17 \bmod 26 = 17 \Rightarrow R$$

$$y_2 = 19 + 17 \bmod 26 = 10 \Rightarrow K$$

$$y_3 = 10 \Rightarrow K$$

$$y_4 = 17 \Rightarrow R$$

$$y_5 = 2 + 17 \bmod 26 = 19 \Rightarrow T$$

$$y_6 = 10 + 17 \bmod 26 = 1 \Rightarrow B$$

Bản mã hoàn chỉnh (Cipher text) $Y = y_1; y_2; y_3; y_4; y_5; y_6 = RKKRTB$



– Thực hiện giải mã: với $k=17$ và cipher text (Y) = $y_1; y_2; \dots; y_6 = RKKRTB$

$$x_1 = y_1 - k \bmod 26 = 17 - 17 \bmod 26 = 0 \Rightarrow A$$

$$x_2 = 10 - 17 \bmod 26 = -7 \bmod 26 = 19 \Rightarrow T$$

$$x_3 = 19 \Rightarrow T$$

$$x_4 = 0 \Rightarrow A$$

$$x_5 = 19 - 17 \bmod 26 = 2 \Rightarrow C$$

$$x_6 = 1 - 17 \bmod 26 = -16 \bmod 26 = 10 \Rightarrow K$$

Bản giải mã hoàn chỉnh (bản gốc - plan text) $X = x_1; x_2; \dots; x_6 = ATTACK$

🔗 **Yêu cầu:** viết chương trình tạo menu với các chức năng sau, trong đó khi thực hiện mã hóa hoặc giải mã, chương trình sẽ yêu cầu nhập khóa k

- Nhập chuỗi
- Xuất chuỗi
- Mã hóa
- Giải mã

🔗 **Mở rộng:**

- Văn bản bao gồm cả ký tự hoa và thường.
- Văn bản bao gồm tất cả các ký tự trong bộ mã ASCII nhưng bỏ qua các mã điều khiển (từ 0-30).
- Cho người dùng chọn chiều dịch chuyển (trái hoặc phải).