

## 4. Lệnh nhập/xuất chuỗi

### 4.1. Nhập chuỗi từ bàn phím

- **Cú pháp:** có thể chọn 1 trong 2 cách sau:

```
gets(<biến chuỗi>) ;
```

```
scanf("%s", <biến chuỗi>) ;
```

- **Giải thích:**

- **gets :**

- Sẽ lấy tất cả các ký tự trong buffer (vùng đệm) của màn hình đưa vào chuỗi. Vì vậy chuỗi dự định nhận có thể chứa những ký tự không mong muốn.
- Để tránh điều này ta dùng hàm **flushall(void)** để xóa mọi buffer (vùng đệm) hoặc hàm **fflush(stdin)** để xóa vùng đệm bàn phím trước hàm nhập chuỗi **gets** (<biến chuỗi>).

## 4. Lệnh nhập/xuất chuỗi

### 4.1. Nhập chuỗi từ bàn phím (tt)

- *Có thể chọn 1 trong 3 cách sau :*

`gets (<biến chuỗi>) ;`

`fgets (<biến chuỗi>, kích thước, stdin) ;`

`scanf ("%s", <biến chuỗi>) ;`

- *Giải thích:*

- **scanf :**

- Nhập dữ liệu và kết thúc việc nhập dữ liệu bằng phím Enter. Lúc này phím Enter sẽ cho hai ký tự có mã ASCII là 13 và 10 trong vùng đệm. Như vậy nếu dùng hàm scanf() thì ký tự có mã ASCII 10 vẫn còn nằm trong vùng đệm.
- Chỉ có thể nhập được một chuỗi không có dấu khoảng trắng.

## 4. Lệnh nhập/xuất chuỗi

### 4.1. Nhập chuỗi từ bàn phím (tt)

- *Ví dụ* : dùng *gets*

```
#include<stdio.h>
#include<conio.h>
#define MAX 50
void main()
{
    char sHoTen[30];
    printf("Nhap ho va ten: ");
    flushall(); //hoặc dùng hàm fflush(stdin);
    gets(sHoTen);
    printf("Chao :");
    puts(sHoTen); //hoặc dùng printf("%s", sHoTen);
}
```

## 4. Lệnh nhập/xuất chuỗi

### 4.1. Nhập chuỗi từ bàn phím (tt)

- *Ví dụ* : dùng *fgets*

```
#include<stdio.h>
#include<conio.h>
#define MAX 50
void main()
{
    char sHoTen[MAX];
    while (getchar() != '\n'); /*sử dụng thêm dòng lệnh
    này nếu chương trình bị "trôi" (không cho nhận chuỗi*/
    printf("Nhap ho va ten: ");
    fflush(); //hoặc dùng hàm fflush(stdin);
    fgets(sHoTen, sizeof(sHoTen), stdin);
    printf("Chao :");
    puts(sHoTen);
}
```

## 4. Lệnh nhập/xuất chuỗi

### 4.2. Xuất chuỗi lên màn hình

- **Cú pháp**: có thể chọn 1 trong 2 cách sau:

```
puts(<biểu thức chuỗi>);  
printf("%s", <biến chuỗi>);
```

- **Ví dụ**: dùng *puts*

```
#include<stdio.h>  
#include<conio.h>  
#define MAX 50  
void main()  
{  
    char sHoTen[30];  
    printf("Nhap ho va ten: ");  
    fflush(); //hoặc dùng hàm fflush(stdin);  
    gets(sHoTen);  
    printf("Chao :");  
    puts(sHoTen);  
}
```

#### 4. Lệnh nhập/xuất chuỗi

### 4.3. Ví dụ minh họa hàm nhập chuỗi

- Cho khai báo thông tin của 1 sinh viên như sau:

```
char maso[5];  
char hoten[20];  
float dtb;
```

- Hàm nhập một sinh viên

```
void Nhap1SV(char maso[], char hoten[], float &dtb )  
{  
    float tam;  
    while (getchar() != '\n'); /*sử dụng thêm dòng lệnh này nếu  
                               chương trình bị "trôi" (không cho nhận chuỗi*/  
    printf("Ma so sinh vien: ");  
    fflush(stdin);  
    gets(maso);  
    printf("Ho ten: ");  
    fflush(stdin);  
    gets(hoten);  
    printf("Diem trung binh: ");  
    scanf("%f", &tam);  
    dtb = tam;  
}
```

## 5. Thư viện hàm xử lý chuỗi trong C

### 5.1. Các hàm sẵn có trong thư viện <string.h>

TT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
1	<b>int strlen</b> (char s[]);	Trả về độ dài của chuỗi s.	<pre>char *s = "Hello"; printf("Do dai s= %d",       strlen(s));</pre> <u>Kết quả: Do dai s= 5</u>
2	<b>strcpy</b> (char dest[], char src[]);	Sao chép nội dung chuỗi src vào chuỗi dest.	<pre>char dest[10]; char *src = "abcdefghi"; strcpy(dest, src); puts(dest);</pre> <u>Kết quả: abcdefghi</u>
3	<b>strncpy</b> (char dest[], char src[], int n);	Chép n ký tự từ chuỗi src sang chuỗi dest. Nếu chiều dài src < n thì hàm sẽ điền khoảng trống cho đủ n ký tự vào dest.	<pre>char dest[4]; char *src = "abcdefghi"; strncpy(dest, src, 3); puts(dest);</pre> <u>Kết quả: abc</u>
4	<b>strcat</b> (char s1[], char s2[]);	Nối chuỗi s2 vào chuỗi s1.	<pre>char *s1 = "Khoa "; char *s2 = "CNTT"; strcat(s1, s2); puts(s1);</pre> <u>Kết quả: Khoa CNTT</u>

## 5. Thư viện hàm xử lý chuỗi trong C

### 5.1. Các hàm sẵn có trong thư viện *<string.h>*

TT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
5	<b>strncat(char s1[], char s2[], int n)</b>	Nối n ký tự đầu tiên của chuỗi s2 vào chuỗi s1.	<pre>char *s1 = "Khoa "; char *s2 = "CNTT"; strncat(s1, s2, 2); puts(s1);</pre> <u>Kết quả: Khoa CN</u>
6	<b>int strcmp(char s1[], char s2[])</b>	So sánh 2 chuỗi s1 và s2 theo nguyên tắc <u>thứ tự từ điển</u> . Phân biệt chữ hoa và thường. Trả về: 0 : nếu s1 bằng s2. >0: nếu s1 lớn hơn s2. <0: nếu s1 nhỏ hơn s2.	<pre>char *s1 = "abcd"; char *s2 = "abCD"; if(strcmp(s1, s2)==0)     puts("Giống nhau"); else     puts("Khác nhau");</pre> <u>Kết quả: Khác nhau</u>
7	<b>int stricmp(char s1[], char s2[])</b>	Tương tự như <b>strcmp()</b> , nhưng không phân biệt hoa thường.	<pre>char *s1 = "abcd"; char *s2 = "abCD"; if(stricmp(s1, s2)==0)     puts("Giống nhau"); else     puts("Khác nhau");</pre> <u>Kết quả: Giống nhau</u>



## 5. Thư viện hàm xử lý chuỗi trong C

### 5.1. Các hàm sẵn có trong thư viện <string.h>

TT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
8	<b>int strncmp</b> (char s1[], char s2[], int n)	Tương tự như <b>strcmp()</b> , nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "abcd"; char *s2 = "abef"; if(strncmp(s1, s2, 2)==0)     puts("Giống nhau"); else     puts("Khác nhau");</pre> <u>Kết quả: Giống nhau</u>
9	<b>int strnicmp</b> (char s1[], char s2[], int n);	Tương tự như <b>stricmp()</b> , nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "aBcd"; char *s2 = "Abef"; if(strnicmp(s1, s2, 2)==0)     puts("Giống nhau"); else     puts("Khác nhau");</pre> <u>Kết quả: Giống nhau</u>

## 5. Thư viện hàm xử lý chuỗi trong C

### 5.1. Các hàm sẵn có trong thư viện <string.h>(tt)

TT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
10	<b>char *strchr</b> <b>(char s[], char c) ;</b>	Tìm lần xuất hiện đầu tiên của ký tự c trong chuỗi s. Trả về: - NULL: nếu không có. - Địa chỉ c: nếu tìm thấy.	<pre>char s[15]; char *ptr, c = 'm'; strcpy(s, "Vi du tim ky tu"); ptr = strchr(s, c); if (ptr)     printf("Ky tu %c xuat hien tai vi tri %d" ,c, ptr-s); else     printf("Khong tim thay");</pre> <p><u>Kết quả: Ky tu m tai: 8</u></p>
11	<b>char *strstr</b> <b>(char s1[], char s2[]) ;</b>	Tìm sự xuất hiện đầu tiên của chuỗi s2 trong chuỗi s1. Trả về: - NULL: nếu không có. - Ngược lại: Địa chỉ bắt đầu chuỗi s2 trong s1.	<pre>char *s1 = "Hello"; char *s2 = "ll", *ptr; ptr = strstr(s1, s2); printf("Chuoi con: "); puts(ptr);</pre> <p><u>Kết quả: Chuoi con: ll</u></p>

## 5. Thư viện hàm xử lý chuỗi trong C

### 5.2. Các hàm sẵn có trong thư viện <ctype.h>

#### 5.2.1. Các hàm dùng để kiểm tra

TT	TÊN HÀM	CHỨC NĂNG	Kết quả trả về của hàm	Ví dụ
1	<i>int</i> <i>isalpha</i> <i>(char_exp)</i>	<b>char_exp</b> có phải là một chữ cái hay không	<>0 khi <b>char_exp</b> là một chữ cái	<i>isalpha('x')</i>
2	<i>int</i> <i>isupper</i> <i>(char_exp)</i>	<b>char_exp</b> có phải là một chữ cái hoa hay không	<>0 khi <b>char_exp</b> là một chữ cái hoa	<i>isupper('a')</i>
3	<i>int</i> <i>islower</i> <i>(char_exp)</i>	<>0 khi <b>char_exp</b> có phải là một chữ cái thường hay không	<>0 khi <b>char_exp</b> là một chữ cái thường	<i>islower('a')</i>
4	<i>int</i> <i>isdigit</i> <i>(char_exp)</i>	<b>char_exp</b> có phải là một ký số hay không	<>0 khi <b>char_exp</b> là một ký số	<i>isdigit('a')</i>
5	<i>int</i> <i>isascii</i> <i>(char_exp)</i>	<b>char_exp</b> là một ký tự có mã ASCII <128	<>0 nếu <b>char_exp</b> là một ký tự có mã ASCII <128	<i>isascii('a')</i>

**5. Thư viện hàm xử lý chuỗi trong C**  
**5.2.Các hàm sẵn có trong thư viện <ctype.h> (tt)**  
**5.2.1. Các hàm dùng để kiểm tra (tt)**

TT	TÊN HÀM	CHỨC NĂNG	Kết quả trả về của hàm	Ví dụ
6	<i>int isspace (char_exp)</i>	<b>char_exp</b> là một khoảng trắng	<>0 khi <b>char_exp</b> là một khoảng trắng	<i>isspace(' ')</i>
7	<i>int isprint (char_exp)</i>	<b>char_exp</b> là một ký tự có thể in được	<>0 khi <b>char_exp</b> là một ký tự có thể in được	<i>isprint('a')</i>
8	<i>int iscntrl (char_exp)</i>	<b>char_exp</b> là một ký tự điều khiển	<>0 khi <b>char_exp</b> là một ký tự điều khiển	<i>iscntrl('a')</i>
9	<i>int ispunct (char_exp)</i>	<b>char_exp</b> là một ký tự dấu	<>0 khi <b>char_exp</b> là một ký tự dấu	<i>ispunct('!')</i>

## 5. Thư viện hàm xử lý chuỗi trong C

### 5.2. Các hàm sẵn có trong thư viện <ctype.h> (tt)

#### 5.2.2. Các hàm dùng để chuyển ký tự thường sang ký tự hoa và ngược lại (tt)

TT	TÊN HÀM	CHỨC NĂNG	Ví dụ
1	<i>int toupper (char_exp)</i>	Chuyển <b>char_exp</b> thành ký tự hoa tương ứng.	<i>toupper('a')</i>
2	<i>int tolower (char_exp)</i>	Chuyển <b>char_exp</b> thành ký tự thường tương ứng.	<i>tolower('A')</i>

**5. Thư viện hàm xử lý chuỗi trong C**  
**5.3. Các hàm sẵn có trong thư viện < stdlib.h >**  
**5.3.1. Các hàm thao tác chuỗi**

TT	TÊN HÀM	CHỨC NĂNG	Ví dụ
1	<i>int</i> <i>atoi(str_exp)</i>	Chuyển một chuỗi sang số nguyên. Việc chuyển đổi sẽ dừng khi gặp ký tự không phải là ký số	<i>atoi('123a45')</i>
2	<i>double</i> <i>atof(char_exp)</i>	Chuyển một chuỗi sang số double. Việc chuyển đổi sẽ dừng khi gặp ký tự không thể chuyển sang dạng double được	<i>toupper('a')</i>
3	<i>char*</i> <i>itoa(int</i> <i>value,</i> <i>char *st, int</i> <i>radix)</i>	Chuyển giá trị số nguyên sang dạng chuỗi và gán vào vùng nhớ mà con trỏ st đang trỏ đến. st là một con trỏ kiểu ký tự	<i>tolower('a')</i>