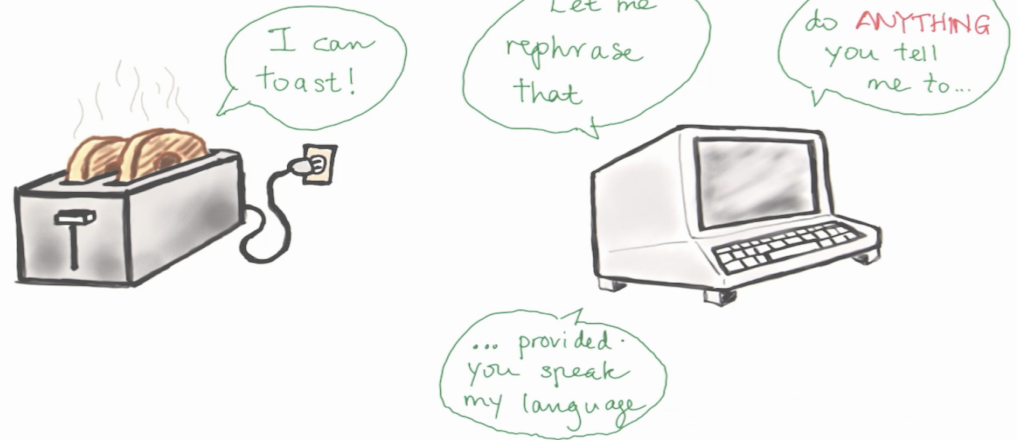


*What is programming?*



Tuần 7

# CHUỖI KÝ TỰ

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT

# Mục đích

- Cung cấp các khái niệm về chuỗi ký tự trong C
- Giới thiệu cách khai báo và khởi tạo chuỗi ký tự
- Giới thiệu cách nhập và xuất chuỗi

# Yêu cầu

- Nắm vững các khái niệm về chuỗi ký tự
- Biết cách khai báo và khởi tạo chuỗi
- Biết cách nhập và xuất chuỗi, cách sử dụng chuỗi trong chương trình
- Biết cách vận dụng chuỗi ký tự để giải quyết các bài toán thực tế
- Biết vận dụng các thuật toán cơ bản để thao tác trên chuỗi

# Nội dung

- Giới thiệu chuỗi ký tự trong C
- Cách khai báo và khởi tạo
- Nhập và xuất chuỗi
- Thao tác trên chuỗi
- Chuyển đổi chuỗi
- Các hàm xử lý chuỗi thông dụng

# Chuỗi ký tự

- C không có kiểu Chuỗi ký tự riêng nên trong C, chuỗi ký tự là một mảng các ký tự.
- Kết thúc bằng ký tự '\0' (ký tự NULL trong bảng mã ASCII)
- Các hằng chuỗi ký tự được đặt trong cặp dấu nháy kép ""

```
name[10] = "Pritesh";
```

```
name[0] = 'P';
```

```
name[1] = 'r';
```

```
name[2] = 'i';
```

```
name[3] = 't';
```

```
name[4] = 'e';
```

```
name[5] = 's';
```

```
name[6] = 'h';
```

```
name[7] = '\0';
```

P	r	i	t	e	s	h	\0
---	---	---	---	---	---	---	----

# Khai báo

- Khai báo chuỗi ký tự theo mảng với cú pháp

```
char variable_name [ SIZE ] ;
```

- Ví dụ:

```
char city[30];  
char name[20];  
char message[50];
```

## ***Ghi chú:***

- Kích thước tối đa của biến chuỗi là 256
- Không nên khai báo thừa để tránh lãng phí bộ nhớ

# Khởi tạo

- **Cách 1:** Dùng mảng có kích thước không xác định

```
char name [] = {'P','R','I','T','E','S','H','\0'};
```

- **Cách 2:** Dùng hằng chuỗi ký tự

```
char name [ ] = "PRITESH";
```

- **Cách 3:** Dùng biến con trỏ

```
char *name = "PRITESH";
```

name



P	R	I	T	E	S	H	\0
---	---	---	---	---	---	---	----

# Nhập chuỗi

- Dùng hàm **gets()** `gets( <variable-name> )`
- **Cú pháp:**
- Ta cũng có thể sử dụng hàm **scanf()** để nhập dữ liệu cho biến chuỗi, tuy nhiên lúc này ta chỉ có thể nhập được một chuỗi không có dấu khoảng trắng.
- Dùng hàm **cgets()** (trong **conio.h**)
- Hàm **getchar()** chỉ nhập một ký tự từ chuỗi

```
#include<stdio.h>
void main() {
    char name[20];
    printf("\nEnter the Name : ");
    gets(name);
}
```



# Xuất chuỗi

- Để xuất một chuỗi lên màn hình, ta sử dụng hàm ***puts()***.
- **Cú pháp:** `puts(variable_name) ;`
- Ngoài ra, ta có thể sử dụng hàm ***printf()***, ***cputs()*** (trong **conio.h**) để hiển thị chuỗi lên màn hình.
- Hàm ***putchar()*** chỉ xuất một ký tự

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

void main() {
    char string[] = "This is an example string\n";
    puts(string); // String is variable Here
    puts("String"); // String is in Double Quotes
    getch();
}
```

# Một số hàm xử lý chuỗi

## *string.h*

- **strcat** cộng chuỗi
- **strlen** xác định độ dài chuỗi
- **toupper**, **tolower** chuyển đổi 1 ký tự sang hoa/thường
- **strupr**, **strlwr** chuyển đổi một chuỗi sang hoa/thường
- **strcpy**, **strncpy** sao chép/sao chép một phần chuỗi
- **strchr**, **strrchr** tìm kiếm ký tự trong chuỗi theo thứ tự xuôi/ngược
- **strstr**, **strrstr** tìm kiếm chuỗi trong chuỗi theo thứ tự xuôi/ngược
- **strcmp**, **stricmp** so sánh chuỗi có/không phân biệt hoa thường
- **memset** khởi tạo chuỗi

## *stdlib.h*

- **atoi**, **atof**, **atol** đổi chuỗi ra số

# Cộng chuỗi strcat

- **Cú pháp:** `char * strcat(char * des, char * src)`
- Hàm cộng chuỗi nguồn (*src*) vào chuỗi đích (*des*)
- Trả lại *con trỏ* chỉ đến chuỗi kết quả

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

int main () {
    char str[80];
    strcpy(str, "these ");
    strcat(str, "strings ");
    strcat(str, "are ");
    strcat(str, "concatenated.");
    puts(str);
    return 0;
}
```

# Xác định độ dài chuỗi strlen

- **Cú pháp:** `size_t strlen ( const char * str );`
- Trả lại độ dài của một chuỗi

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

int main () {
    char szInput[256];
    printf("Enter a sentence: ");
    gets(szInput);
    printf ("The sentence entered is %u characters
            long.\n",
            (unsigned)strlen(szInput));
    return 0;
}
```

# Chuyển ký tự sang hoa/thường

- **Cú pháp:**

```
int toupper ( int c );  
int tolower ( int c );  
int isupper ( int c );  
int islower ( int c );  
int isalpha ( int c );
```

- Các hàm này trong thư viện **ctype.h**

```
#include <stdio.h>  
#include <ctype.h>  
int main (){  
    int i=0; char c;  
    char str[]="Test String.\n";  
    while (str[i]){  
        c=str[i];  
        putchar(toupper(c));  
        i++;  
    }  
    return 0;  
}
```

# Đổi chuỗi thành hoa/thường

- **Cú pháp:**

```
char *strupr(char * str)
char *strlwr(char * str)
```

- Chuyển đổi chuỗi sang hoa/thường, kết quả trả về là con tr

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
int main(){
    char str[100];
    printf("Input string: ");
    gets(str);
    printf("Output string: ");
    puts(strupr(str));
    getch();
    return 0;
}
```

# Sao chép chuỗi

- **Cú pháp:**

```
char * strcpy(char * des, const char * src)
char * strncpy(char * des, const char * src, size_t n)
```

- **Ví dụ:**

```
#include <stdio.h>
#include <string.h>
int main () {
    char str1[]="Sample string";
    char str2[40];
    char str3[40];
    strcpy(str2,str1);
    strcpy(str3,"copy successful");
    printf ("str1: %s\nstr2: %s\nstr3: %s\n",
           str1,str2,str3);
    return 0;
}
```

# Tìm kiếm chuỗi

- **Cú pháp:**

```
char * strchr ( char * str, int character );  
char * strrchr ( char * str, int character );  
char * strstr ( char * str1, char * str2 );  
char * strrstr ( char * str1, char * str2 );
```

- Dùng tìm kiếm một ký tự/chuỗi trong một chuỗi theo thứ tự xuôi/ngược

```
#include <stdio.h>  
#include <string.h>  
int main () {  
    char str[] = "This is a simple string";  
    char * pch;  
    pch = strstr(str, "simple");  
    strncpy(pch, "sample", 6);  
    puts (str);  
    return 0;  
}
```



# So sánh chuỗi

- **Cú pháp:**

```
int strcmp ( const char * str1, const char * str2 );  
int stricmp ( const char * str1, const char * str2 );  
int strncmp ( const char * str1, const char * str2, size_t num );
```

- **Kết quả trả về:**

```
<0   nếu   s1 < s2  
==0  nếu   s1 ==s2  
>0   nếu   s1 > s2
```

```
#include <conio.h>  
#include <stdio.h>  
#include <string.h>  
  
int main(){  
    char s1[]="apple",s2[]= Apple";  
    printf ("%d",strcmp(s1,s2));  
    getch();  
    return 0;  
}
```

# Đổi chuỗi sang số

- **Cú pháp:**

```
int    atoi(const char *s)
long   atol(const char *s)
float  atof(const char *s)
```

- Nếu kết quả không thành công, kết quả trả về của các hàm là 0

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

int main () {
    char str[100];
    printf ("Enter a number: "); gets(str);
    printf ("The value entered is %d", atoi (str));
    return 0;
}
```

# Tổng kết

- Trong C không có một kiểu chuỗi ký tự riêng mà đó là một mảng ký tự được kết thúc bằng ký tự '\0'.
- Các thao tác trên chuỗi ký tự giống như trên mảng
- Ta cũng có thể khai báo chuỗi ký tự bằng con trỏ, nhưng sẽ đề cập ở những chương sau.

# Kiểm tra kiến thức

1. Với hai chuỗi ký tự giống nhau, hàm **strcmp()** trả lại kết quả gì?
  - a. -1
  - b. 1
  - c. 0
  - d. yes

# Kiểm tra kiến thức

2. Hàm nào trong các hàm sau đây tìm một ký tự trong chuỗi và trả lại vị trí của kết quả sau cùng tìm kiếm được?
- a. `strchr()`
  - b. `strrchr()`
  - c. `strstr()`
  - d. `strnset()`

# Kiểm tra kiến thức

## 3. Cho biết kết quả chương trình sau:

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

int main() {
    char str1[20] = "Hello", str2[20] = " World";
    printf("%s\n", strcpy(str2, strcat(str1, str2)));
    return 0;
}
```

- a. Hello
- b. World
- c. Hello World
- d. WorldHello

# Kiểm tra kiến thức

4. Cho biết kết quả chương trình sau:

```
#include<stdio.h>

int main() {
    printf(5+"Good Morning\n");
    return 0;
}
```

- a. Good Morning
- b. Good
- c. M
- d. Morning

# Kiểm tra kiến thức

## 5. Cho biết kết quả chương trình sau:

```
#include <stdio.h>

int main() {
    static char s[25] = "The cocaine man";
    int i=0;
    char ch;
    ch = s[++i]; printf("%c", ch);
    ch = s[i++]; printf("%c", ch);
    ch = i++[s]; printf("%c", ch);
    ch = ++i[s]; printf("%c", ch);
    return 0;
}
```

a. hhe!  
b. de c

c. The c  
d. Hhec



# Kiểm tra kiến thức

6. Cho biết kết quả chương trình sau:

```
#include<stdio.h>

int main() {
    char str1[] = "Hello";
    char str2[10];
    char *t, *s;
    s = str1;
    t = str2;
    while(*t++=*s++);
    printf("%s\n", str2);
    return 0;
}
```

a. Hello

b. HelloHello

c. Không in ra gì cả

d. ello

# Kiểm tra kiến thức

7. Cho biết kết quả chương trình sau, biết rằng địa chỉ bộ nhớ của chuỗi “Hello1” là 1022

```
#include<stdio.h>

int main() {
    printf("%u %s\n", &"Hello1", &"Hello2");
    return 0;
}
```

- a. 1022 Hello2
- b. Hello1 1022
- c. Hello1 Hello2
- d. 1022 1022
- e. Xảy ra lỗi

# Bài tập

1. Viết chương trình nhập một chuỗi ký tự từ bàn phím, xuất mã ASCII của từng ký tự có trong chuỗi.
2. Viết chương trình con kiểm tra một chuỗi có đối xứng hay không, kiểm chứng trên hàm main.
3. Viết chương trình đảo ngược một chuỗi ký tự.
4. Viết chương trình nhập vào họ tên của một người, in lần lượt các từ trong họ tên đó trên mỗi dòng.
5. Viết chương trình đổi số tiền từ số thành chữ.



Mời các bạn đặt câu hỏi

CT101 – Lập trình căn bản

**Khoa CNTT&TT – ĐHCT**