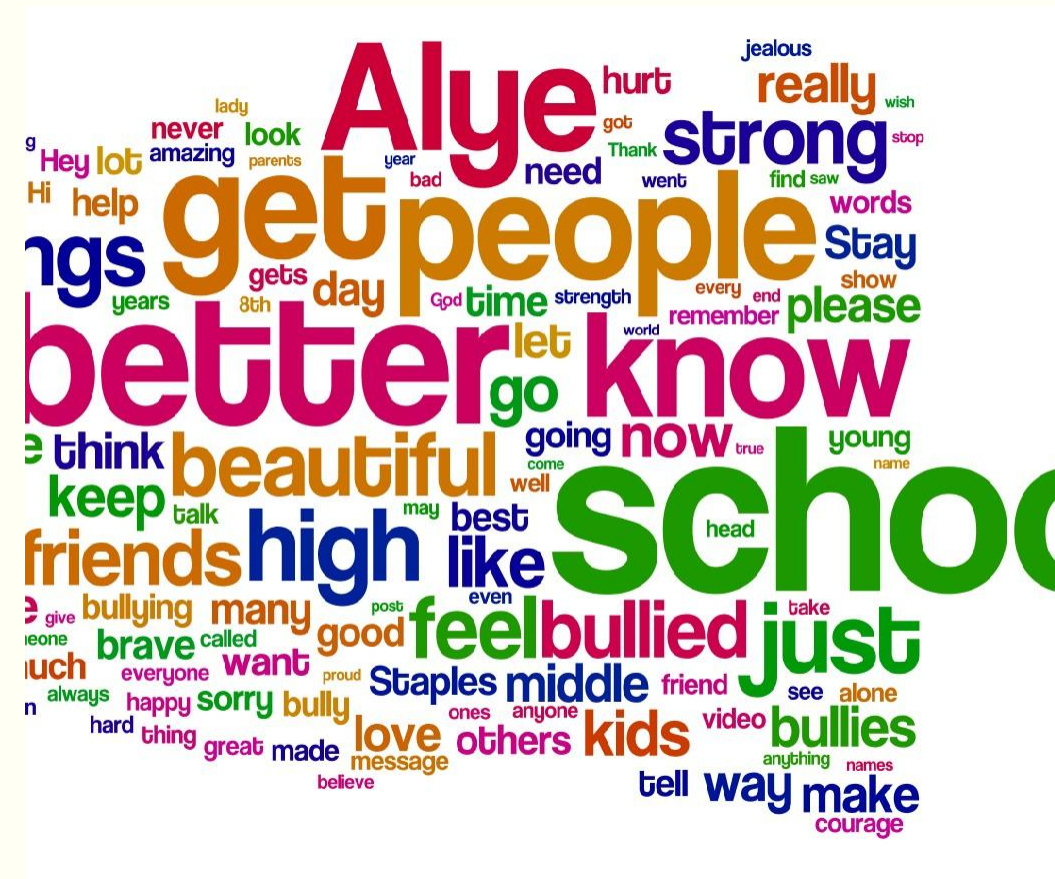


BÀI 8 XÂU HÀM CHUNG



Giảng viên: Nguyễn Quỳnh Diệp – Khoa CNTT – ĐH Thủy Lợi

Email: diepnnq@tlu.edu.vn

NỘI DUNG

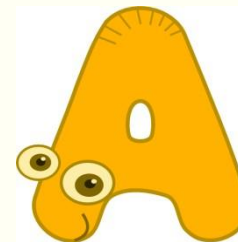
- Xâu kí tự
- Các hàm thao tác với xâu
- Hàm chung



XÂU KÍ TỰ

“Ngôn ngữ lập trình C++”

“Hôm nay trời rất đẹp!”



XÂU KÍ TỰ

▪ Xâu:

- Là một mảng (dãy) các kí tự
- Kết thúc bằng kí tự **'\0'** là kí tự NULL

▪ Ví dụ:

“Ngon ngu C++”

0	1	2	3	4	5	6	7	8	9	10	11	12	13
N	g	o	n		n	g	u		C	+	+	\0	



Kí tự đặt trong dấu nháy đơn ‘ ‘

Xâu kí tự đặt trong dấu nháy kép “ “

MÃ ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

KHAI BÁO XÂU KÍ TỰ

▪ Cách 1:

```
char tenxau [chieudai];
```

```
char tenxau [chieudai] = <Chuỗi kí tự>;
```

```
char tenxau [chieudai] = { 'kí tự 1', 'kí tự 2'....., '\0' };
```

▪ Ví dụ:

```
char xau[15] = "Xin chao!";
```

```
char xau[15] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

```
char xau[ ] = "Xin chao!";
```



Phép gán xâu = chỉ được dùng khi khai báo biến

Một xâu có n kí tự cần một mảng có kích thước $n+1$

KHAI BÁO XÂU KÍ TỰ

▪ Cách 2: sử dụng lớp **string**

```
string tenxau;
```

```
string tenxau = <Chuỗi kí tự>;
```

▪ Ví dụ:

```
string str;
```

```
str="Xin chao!";
```

```
string xau = "Xin chao!";
```

TRUY NHẬP VÀO PHẦN TỬ CỦA XÂU

▪ Cú pháp:

tenxau [chỉ số của kí tự]

▪ Ví dụ:

```
string str = "Ha Noi";
```

str[0] → lưu 'H'

str[1] → lưu 'a'

str[2] → lưu ' '

str[10] → ?

MỘT SỐ HÀM VỚI KÍ TỰ

Hàm	Mô tả
tolower(int ch)	Chuyển thành kí tự thường
toupper(int ch)	Chuyển thành kí tự hoa
islower(int ch)	Kiểm tra chữ thường
isupper(int ch)	Kiểm tra chữ hoa
isdigit(int ch)	Kiểm tra chữ số
isalpha(int ch)	Kiểm tra xem kí tự có là chữ cái không
isspace(int ch)	Kiểm tra kí tự dấu cách
isctrl(int ch)	Kiểm tra kí tự điều khiển

MỘT SỐ VÍ DỤ

- Ví dụ: Nhập vào một xâu kí tự. Đếm số kí tự viết hoa.

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string str;
7      cout<<"Nhập vào một xâu kí tự:";
8      getline(cin, str);
9      int i=0, dem=0;
10     while(str[i]!='\0')
11     {
12         if(isupper(str[i]))//Kiểm tra nếu là kí tự hoa
13             dem++;
14         i++;
15     }
16     cout<<"Số kí tự hoa là:"<<dem;
17     return 0;
18 }
```