MỘT SỐ LƯU Ý VỀ KIỂU CHUỖI KÍ TỰ TRONG C++

1/ Hiểu về chuỗi kí tự và con trỏ

Chuỗi kí tự là một mảng các kí tự chữ và chữ số, nó được kết thúc bởi kí tự '\0'

Chuỗi kí tự cũng là một con trỏ, nó luôn trỏ về vị trí đầu tiên của chuỗi.

Chú ý:

Khi thao tác trên chuỗi phải đảm bảo rằng chuỗi được xuất ra phải có kí tự '\0'

Khi con trỏ đã trỏ vào chuỗi nào đó thì nếu mọi thao tác trên con trỏ sẽ làm chuỗi thay đổi , ngoại trừ việc con trỏ trỏ vào chuỗi khác (xem Vd 3 bên dưới).

Có những dòng khai báo bắt buộc phải tuân theo.

Một số ví dụ về cách khai báo chuỗi và con trỏ

Quét khối màu xanh là chỗ sai

Vd1:

```
char a[50];
a="ngon ngu";
```

puts(a);-> báo lỗi

Bắt buộc phải khai báo như thế này

```
char a[50]="ngon ngu";
```

puts(a);-> ngon ngu

Vd2:

```
char *s,*a;
a="ngon ngu";
```

s=a;

puts(s);-> ngon ngu

hoặc khai báo là:

```
char *s,*a="ngon ngu";
```

s=a;

puts(s);-> ngon ngu

Vd 3:

```
char *s,a[50]= "ngon ngu";
```

s=a+5;

puts(s);-> ngu

Vd 4:

char s[50],a[50]= "ngon ngu";

s=a+5;

puts(s);-> báo lỗi

Vd 5:

```
char *s,a[50]= "ngon ngu";
```

s=a[5];

puts(s);-> báo lỗi

Vd 6:

char *s,a[50]= ”ngon ngu”;

s[2]=a+5;

puts(s);-> báo lỗi

Vd7:

```
char *s,a[50]= "ngon ngu";
```

s+2=a+5; hoặc s=a; s+2=a+5;

puts(s);-> báo lỗi

Vd 8:

```
char *s,a[20]="cho",b[20]="meo";
```

s=a;

```
s[0]='t';
```

s=b;

puts(s);->meo

puts(a);->tho

Vd 9:

```
char *s,a[20]="cho",*b;
```

s=a;

b=s;

s=s+1;

puts(s);->ho

puts(a);->cho

Vd10:

```
char *s,a[50]= "ngon ngu";
```

s+2=a+5; hoặc s=a; s+2=a+5;

puts(s);-> báo lỗi

Vd 11: bắt buộc phải khai báo như thế này(ở Vd 12,13,14 là khai báo sai)

```
char a[100]= {'n','g','o','n','1','2','3','n','g','u','\0'};
```

```
puts(a);->ngon123ngu
```

Vd 12:

```
char *s,a[100];
a= {'n','g','o','n','1','2','3','n','g','u','\0'};
```

puts(a);-> báo lỗi

Vd 13:

```
char *a= {'n','g','o','n','1','2','3','n','g','u','\0'};
```

puts(a);-> báo lỗi

Vd 14:

```
char *a;
a= {'n','g','o','n','1','2','3','n','g','u','\0'};
```

puts(a);-> báo lỗi

2/ Cách dùng một số hàm trong chuỗi kí tự

-Hàm strlen: tìm độ dài của xâu:

strlen(a); với a là một chuỗi hoặc a là con trỏ đã được trỏ vào 1 chuỗi nào đó.

strlen("conmeo");->6

-Hàm strcat: nối xâu b vào xâu a

strcat(a,b); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó, với b là chuỗi hoặc b là con trỏ đã được trỏ vào chuỗi nào đó.

Vd:

```
char b[20]="ngon",c[20]="ngu",*s=c;
strcat(b,s);
puts(b);
```

-Hàm strcpy: copy xâu b vào xâu a:

strcpy(a,b); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó, với b là chuỗi hoặc b là con trỏ đã được trỏ vào chuỗi nào đó.

Vd:

```
char b[20]="ngon",c[20]="ngu",*s=b;

strcpy(s,c);
puts(s);->ngu
```

nếu strcpy(s+4,c) ->kết quả là ngonngu

-Hàm strncpy: copy n kí tự xâu b vào xâu a:

strncpy(a,b,n); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó, với b là chuỗi hoặc b là con trỏ đã được trỏ vào chuỗi nào đó,n là số kí tự cần copy.

Vd:

Trường hợp a là con trỏ đã được trỏ(tương tự trường hợp a là chuỗi đã có gán giá trị)

```
char b[20]="ngon",c[20]="ngulaptrinh",*a=b;
strncpy(a,c,6);
puts(a);->ngulap
```

Trường hợp a là chuỗi trống

```
char c[20]="ngulaptrinh",s[20];
strncpy(s,c,6);
s[6]='\0';   lưu ý chỗ này
puts(s); ->ngulap
```

-Hàm strstr: tìm sự xuất hiện của xâu b trong xâu a

strstr(a,b); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó, với b là chuỗi hoặc b là con trỏ đã được trỏ vào chuỗi nào đó.

Vd:

```
char a[20]="ngonngulaptrinh",b[10]="lap",*s;
s=strstr(a,b);
puts(s);->laptrinh
```

-Hàm strchr: tìm sự xuất hiện của kí tự b trong xâu a

strchr(a,b); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó, với b là một kí tự.

Vd:

```
char a[20]="phanmem",b='m',*s;
s=strchr(a,b);
puts(s);->mem
```

-Hàm strrev:đảo ngược chuỗi a

strrev(a); với a là chuỗi hoặc a là con trỏ đã được trỏ vào chuỗi nào đó.

Vd:
```
char s[255]="con co";
Cout<<strrev(s);->oc noc
```

Chú ý: khi gọi strrev(s) thì s sẽ bị đảo ngược ngay lập tức

```
char *a,s[255]="con co";
a=strrev(s);
puts(a);->oc noc
puts(s);->oc noc ->vấn đề là ở chỗ này
```

-Hàm strlwr: đổi thành xâu thường

strlwr(a) : a là con trỏ được trỏ hoặc chuỗi

Vd:
```
char a[20]="BAI TAP";
puts(strlwr(a));->bai tap
```

-hàm strupr: đổi thành xâu hoa

strupr(a) : a là con trỏ được trỏ hoặc chuỗi

Vd:
```
char a[20]="bai tap";
```

```
puts(strlwr(a));->BAI TAP
```

-Hàm touper: đổi 1 kí tự thành kí tự hoa

strupr(a) : a là 1 kí tự

Vd:

```
char a='b',s;
s=toupper(a);
cout<<s;->B
```

-Hàm strcmp: So sánh 2 xâu a và b có phân biệt hoa thường

strcmp(a,b); với a là con trỏ đã được trỏ vào chuỗi hoặc a là chuỗi, b là con trỏ đã được trỏ vào chuỗi hoặc b là chuỗi.

kết quả trả về là: 0 nếu 2 xâu bằng nhau

giá trị âm nếu xâu a nhỏ hơn b

giá trị dương nếu xâu a lớn hơn b

Vd:

```
char a[10]="anh",b[10]="Anh";
cout<<strcmp(a,b);->1 vì a(97)>A(65) => a>b
```

-Tương tự hàm strcmpi (hoặc stricmp): so sánh 2 xâu a và b có không biệt hoa thường

Vd:

```
char a[10]="anh",b[10]="Anh";
cout<<strcmp(a,b);->0
```

-Hàm strncmp: So sánh n kí tự đầu tiên của 2 xâu a,b có phân biệt hoa thường

strcmp(a,b,n); với a là con trỏ đã được trỏ vào chuỗi hoặc a là chuỗi, b là con trỏ đã được trỏ vào chuỗi hoặc b là chuỗi,n số kí tự so sánh

-Hàm strncmpi: So sánh n kí tự đầu tiên của 2 xâu a,b không phân biệt hoa thường

strcmpi(a,b,n); với a là con trỏ đã được trỏ vào chuỗi hoặc a là chuỗi, b là con trỏ đã được trỏ vào chuỗi hoặc b là chuỗi,n số kí tự so sánh

-Hàm strcspn: Trả về vị trí đầu tiên của bất cứ kí tự nào trong a tìm thấy trong b. Nếu không tìm thấy thì trả về độ dài chuỗi a

Vd:

```
char *str="Xcross87 ";
char *key="123456789";
int pos = strcspn(str,key);
cout << "tim thay o vi tri" << pos; ->tim thay o vi tri 6
```

-Hàm strpbrk: Trả về vị trí của kí tự đầu tiên trong chuỗi a mà không khớp b.Trả về NULL nếu không tìm thấy.

Vd:

```
char *str1="con ga trong ko phai la con ga mai";
```

```
char *str2="1234956za";
char* found;  found = strpbrk(str1,str2);
 if (found!= NULL ) cout << " Tim thay ki tu dau tien: " << *found; ->Tim thay ki tu đau tien a
```

-Hàm memset: khởi tạo chuỗi a với n kí tự đầu tiên là kí tự b

memset(a,b,n); a chuỗi cần tạo, b kí tự cần tạo,n số kí tự cần tạo

Vd:

```
char  a[20];
memset(a,'k',5);
a[5]='\0';
cout<<a;->kkkkk
```

BẢNG MÃ ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com