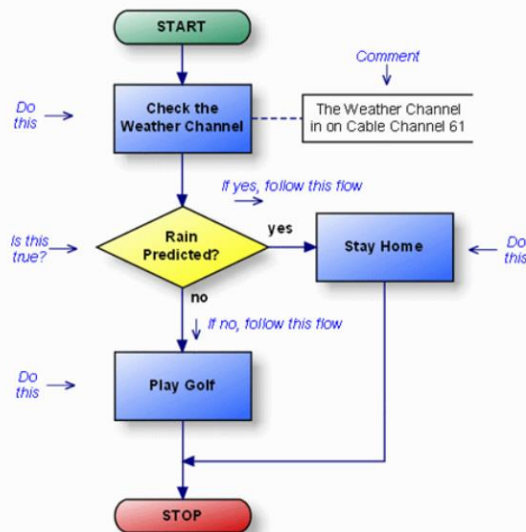


NHẬP MÔN LẬP TRÌNH

CHƯƠNG 7: CHUỖI KÝ TỰ (STRING)



GV: Phạm Nguyễn Sơn Tùng

Email: pnstung@fit.hcmus.edu.vn

THÔNG TIN CHUNG

1

Giới thiệu tổng quan

2

Các cách khai báo chuỗi

3

Các hàm xử lý trên chuỗi

4

Bài tập ứng dụng tại lớp

5













Bài tập về nhà

ĐỊNH NGHĨA CHUỖI TRONG STL C++

Định nghĩa: Chuỗi là một dãy các ký tự được ghép lại với nhau.

```
string s = "football";
```

0	1	2	3	4	5	6	7
f	o	o	t	b	a	l	l

	[size]	8
	[capacity]	15
▸ 	[allocator]	allocator
	[0]	102 'f'
	[1]	111 'o'
	[2]	111 'o'
	[3]	116 't'
	[4]	98 'b'
	[5]	97 'a'
	[6]	108 'l'
	[7]	108 'l'
▸ 	[Raw View]	{_Mypair=allocator }

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		DEL

HÀM NHẬP VÀ XUẤT CHUỖI TỪ BÀN PHÍM

Cách 1: sử dụng cin, cout để nhận 1 chuỗi từ bàn phím, cách này không lấy khoảng trắng.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s;
    cout << "Nhap vao chuoai: ";
    cin >> s;
    cout << s;
    return 0;
}
```

HÀM NHẬP VÀ XUẤT CHUỖI TỪ BÀN PHÍM

Cách 2: sử dụng getline để lấy chuỗi vào, cách này lấy toàn bộ chuỗi luôn cả khoảng trắng.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s;
    cout << "Nhap vao chuoai: ";
    getline(cin, s);
    cout << s;
    return 0;
}
```

CÁC HÀM THAO TÁC TRÊN CHUỖI

1. Lấy kích thước chuỗi: **size()**, **length()**
2. Lấy chuỗi con: **substr**(int batdau, int dodai)
3. Xóa chuỗi: **clear()**
4. Xóa chuỗi: **erase**(s.begin() + int vitri)
5. Thêm ký tự vào chuỗi: **push_back**(char kytu)
6. Xóa ký tự khỏi chuỗi: **pop_back()**
7. Chèn ký tự vị trí bất kỳ: **insert**(s.begin() + int vitri, char kytu)
8. Các phép toán so sánh: **==**, **!=**

CÁC HÀM THAO TÁC TRÊN CHUỖI

1. **size**: Hàm dùng để lấy độ dài của chuỗi

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "nhap mon lap trinh";
    int dodai = s.size();
    cout << dodai;
    return 0;
}
```

Kết quả

18

CÁC HÀM THAO TÁC TRÊN CHUỖI

2. substr: Hàm dùng để sao chép chuỗi này sang chuỗi kia 1 đoạn nào đó bất kỳ.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "nhap mon lap trinh";
    string k = s.substr(0, 8);
    cout << k;
    return 0;
}
```

Kết quả

nhap mon

CÁC HÀM THAO TÁC TRÊN CHUỖI

3. **clear**: Hàm dùng để xóa toàn bộ chuỗi.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "nhap mon lap trinh";
    s.clear();
    cout << s;
    return 0;
}
```

Kết quả



CÁC HÀM THAO TÁC TRÊN CHUỖI

4. **erase**: Hàm dùng để xóa một ký tự bất kỳ trong chuỗi.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "nhap mon lap trinh";
    s.erase(s.begin() + 4);
    cout << s;
    return 0;
}
```

Kết quả

nhapmon lap trinh

CÁC HÀM THAO TÁC TRÊN CHUỖI

push_back: Hàm dùng thêm một ký tự vào cuối chuỗi.

Ví dụ minh họa:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "Ky thuat lap trinh";
    s.push_back('.');
    cout << s;
    return 0;
}
```

Kết quả

Ky thuat lap trinh.

CÁC HÀM THAO TÁC TRÊN CHUỖI

pop_back: Hàm dùng lấy(xóa) một ký tự ở cuối chuỗi.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "ky thuat lap trinh";
    s.pop_back();
    cout << s;
    return 0;
}
```

Kết quả

ky thuat lap trin

CÁC HÀM THAO TÁC TRÊN CHUỖI

insert: Hàm dùng thêm một ký tự bất kỳ vào vị trí bất kỳ của chuỗi:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s = "Ky thuat lap trinh ";
    s.insert(s.begin() + 1, 'U');
    cout << s;
    return 0;
}
```

Kết quả

kUy thuat lap trinh

BÀI TẬP VỀ HÀM CƠ BẢN

Bài tập 1: Viết chương trình nhập vào chuỗi, sau đó kiểm tra xem chuỗi đó có chứa ký tự 'N' hay không? Nếu có trả về 1 ngược lại trả về 0.

Bài tập 2: Viết chương trình đếm số lần xuất hiện của 1 ký tự trong chuỗi (ký tự do người dùng nhập vào)

Bài tập 3: Viết chương trình tìm ký tự xuất hiện ít nhất trong chuỗi.

BÀI TẬP VỀ HÀM CƠ BẢN

Bài tập 4: Cho chuỗi sau “I would like to buy a motor bike”

Viết chương trình xóa chuỗi trên thành chuỗi
“like a motor bike”

Viết chương trình sao chép chuỗi trên qua chuỗi
mới “motor bike”

Bài tập 5: Viết chương trình đếm ký tự phân biệt xuất
hiện trong chuỗi theo 2 cách:

- Phân biệt ký tự hoa – thường.
- Không phân biệt ký tự hoa thường.

BÀI TẬP VỀ HÀM CƠ BẢN

Bài tập 6: Viết chương trình mô phỏng lại hàm insert của chuỗi.

Bài tập 7: Viết chương trình tìm các ký tự nào trong 26 chữ cái không xuất hiện trong chuỗi “**Nhap mon lap trinh**”.

Bài tập 8: Viết sao chép n ký tự từ chuỗi 1 sang chuỗi 2.

BÀI TẬP VỀ HÀM CƠ BẢN

Bài tập 9: Kiểm tra chuỗi 1 có xuất hiện trong chuỗi 2 hay không?

■ Ví dụ:

- Chuỗi 1: “hoc”
- Chuỗi 2: “hom nay toi di **hoc** o truong dai hoc”
- Chuỗi 1 có xuất hiện trong chuỗi 2.

Bài tập 10: Đếm số lần xuất hiện nhiều nhất của 1 từ trong chuỗi. Nếu nhiều từ xuất hiện bằng nhau, chỉ cần in ra 1 từ.

■ Ví dụ:

- Chuỗi 2: “hom nay toi di **hoc** o truong dai **hoc**”

➔ từ “hoc”

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU TRONG CHUỖI

1. `atof`: chuyển chuỗi sang kiểu số **thực**.
2. `atoi`: chuyển chuỗi sang kiểu số **nguyên**.
3. `atol`: chuyển đổi chuỗi sang kiểu số **nguyên dài**.
4. `to_string`: chuyển từ số về chuỗi.
5. `toupper`: Chuyển đổi ký tự viết thường thành hoa.
6. `tolower`: Chuyển đổi ký tự viết hoa thành thường.

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

atof: Hàm dùng để chuyển chuỗi sang kiểu dữ liệu số thực.

```
double atof (const char *_String)
```

```
string s = "3.14";  
double kq = atof(s.c_str());  
cout<<kq;
```

Kết quả

3.14

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

atoi: Hàm dùng để chuyển chuỗi sang kiểu dữ liệu số nguyên.

```
int atoi (const char *_Str)
```

```
string s = "358";  
int kq = atoi(s.c_str());  
cout<<kq;
```

Kết quả

358

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

atoi: Hàm dùng để chuyển chuỗi sang kiểu dữ liệu số nguyên dài.

```
long atoi (const char *_Str)
```

```
string s = "35899";  
long kq = atoi(s.c_str());  
cout<<kq;
```

Kết quả

35899

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

to_string: Hàm dùng để chuyển kiểu số nguyên sang kiểu dữ liệu chuỗi.

▲ 1 of 9 ▼ `std::string to_string(int _Val)`

```
string s;  
int k = 15789;  
s = to_string(k);  
cout<<s;
```

Kết quả

15789

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

toupper: Hàm dùng để chuyển kiểu số nguyên sang kiểu dữ liệu chuỗi.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s = "football";
    for(int i = 0; i < s.size(); i++)
        s[i] = toupper(s[i]);
    cout << s;
    return 0;
}
```

Kết quả

FOOTBALL

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

Dùng mã ASCII để chuyển đổi hoa thường.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s = "football";
    for(int i = 0; i < s.size(); i++)
        s[i] = s[i] - 32;
    cout << s;
    return 0;
}
```

Kết quả

FOOTBALL

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

tolower: Hàm dùng để chuyển kiểu số nguyên sang kiểu dữ liệu chuỗi.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s = "footBall";
    s[4] = tolower(s[4]);
    cout << s;
    return 0;
}
```

Kết quả

football

CÁC HÀM CHUYỂN ĐỔI DỮ LIỆU

Dùng mã ASCII để chuyển đổi hoa thường.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s = "footBall";
    s[4] = s[4] + 32;
    cout << s;
    return 0;
}
```

Kết quả

football

BÀI TẬP VỀ HÀM KIỂM TRA KÝ TỰ

Bài tập 1: Viết chương trình nhập vào chuỗi, sau đó sau mỗi dấu chấm viết in hoa ký tự đầu tiên của chuỗi.

Bài tập 2: Viết chương trình đếm xem có bao nhiêu ký tự viết hoa trong chuỗi nhập vào.

CÁC HÀM KIỂM TRA KÝ TỰ

Sử dụng các hàm sau thuộc thư viện <ctype.h>

- isalpha: kiểm tra có phải chữ cái không.
- isdigit: kiểm tra có phải số hay không.
- islower: kiểm tra viết thường.
- isupper: kiểm tra viết hoa.

CÁC HÀM KIỂM TRA KÝ TỰ

isalpha: Hàm dùng để kiểm tra ký tự có phải là ký tự chữ cái hay không.

```
string s = "Ky Thuat Lap Trinh";  
int kq = isalpha(s[1]);  
cout<<kq;
```

Kết quả trả về:

- bằng 0: ký tự đó không phải chữ cái.
- khác 0: ký tự đó là chữ cái.

Kết quả

2

BÀI TẬP VỀ HÀM KIỂM TRA KÝ TỰ

Bài tập 1: Đếm chuỗi sau có bao nhiêu ký tự là số s = “thu 2 la ngay 19 thang 5”.

Bài tập 2: Đếm xem có bao nhiêu ký tự viết hoa và bao nhiêu ký tự viết thường.

Bài tập 3: Viết hàm đảo ngược chuỗi.

Bài tập 4: Viết chương trình đảo chữ đầu và cuối của chuỗi.

■ Ví dụ: “Co an bo” → “Bo an co”