Big-O Blue

LECTURE 05

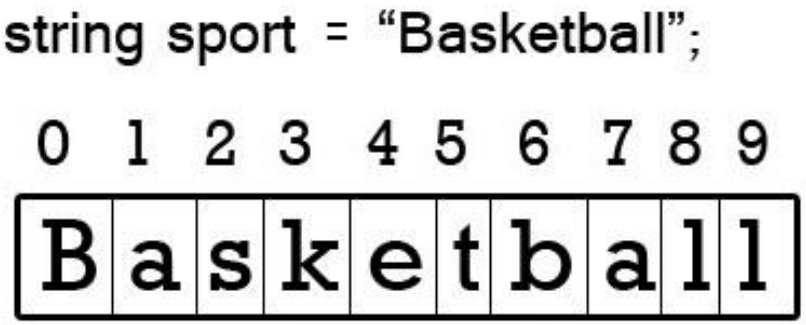
STRING & CHARACTER

Phạm Nguyễn Sơn Tùng

Email: sontungtn@gmail.com

String là gì?

String là một dãy gồm nhiều ký tự (character) liên tục nhau, các kí tự ở đây rất đa dạng có thể là chữ cái, số, dấu cách, hay các kí hiệu...



C++: string

Python: variable_name =

Java: String

Cách khai báo và sử dụng

Thư viện:

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

Khai báo:

```
string variable_name;
```

Ví dụ:

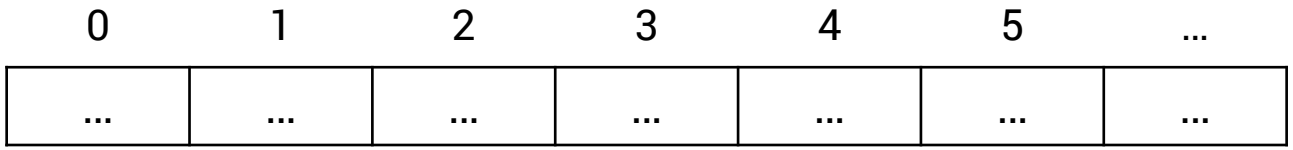
```
string s;
```

Khai báo: các biến trong python có thể không cần khai báo trước, tuy nhiên nên khai báo trước một chuỗi rỗng.

```
variable_name = ""
```

Ví dụ:

```
s = ""
```



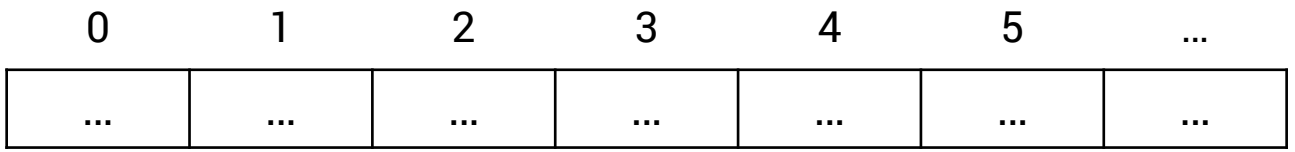
Cách khai báo và sử dụng

Khai báo:

```
String variable_name;
```

Ví dụ:

```
String s;
```



Cách khai báo có tham số đầu vào (1)

C++

```
string variable_name(“value”);
```

Ví dụ:

```
string s(“algorithm”);
```

python™

```
variable_name = "value"
```

Ví dụ:

```
s = "algorithm"
```

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

Cách khai báo có tham số đầu vào (1)

Khai báo: đầu vào là một chuỗi.

```
String variable_name = “value”;
```

Ví dụ:

```
String s = "algorithm";
```

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

Cách khai báo sao chép – toàn bộ (2)


```
string variable_name0(“value”);
string
variable_name1(variable_name0);
```

Ví dụ:

```
string s0(“algorithm”);
string s1(s0);
```

```
variable_name0 = "value"
variable_name1 = variable_name0
```

Ví dụ:

```
s0 = "algorithm"
s1 = s0
```

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

Cách khai báo sao chép – toàn bộ (2)


```
String variable_name0 = “value”;
String variable_name1 = new String(variable_name0);
```

Ví dụ:

```
String s0 = "algorithm";
String s1 = new String(s0);
```

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

Cách khai báo sao chép – một phần (3)


```
string variable_name0("value");
string variable_name1(variable_name0,start_pos,num_chars);
```

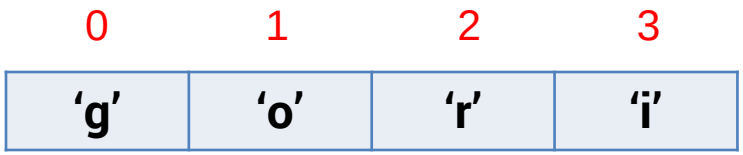
Ví dụ:

```
string s0("algorithm");
string s1(s0, 2, 4);
```

```
string variable_name0("value");
string variable_name1(iterator1, iterator2);
```

Ví dụ:

```
string s0("algorithm");
string s1(s0.begin() + 2, s0.begin() + 6);
```



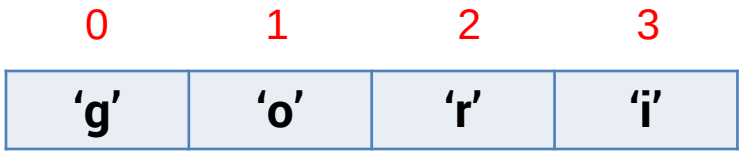
Cách khai báo sao chép – một phần (3)

Python cho phép lấy một đoạn con trong string/list bằng cú pháp <variable_name>[start_pos:end_pos], nó sẽ lấy ra các phần tử trong đoạn khoảng [start_pos, end_pos) (lấy chuỗi con).

```
variable_name0 = "value"
variable_name1 = variable_name0 [start_pos:end_pos]
```

Ví dụ:

```
s0 = "algorithm"
s1 = s0[2:6]
```



Cách khai báo sao chép – một phần (3)

Java không có copy constructor để khai báo sao chép một phần. Ta có thể kết hợp với phương thức **substring** trong String để khởi tạo.

```
String str0 = "value";
String str1 = str0.substring(start_pos, end_pos);
```

Ví dụ:

```
String s0 = "algorithm";
String s1 = s0.substring(2, 6);
```

0	1	2	3
'g'	'o'	'r'	'i'

Hàm chèn chuỗi/ký tự vào chuỗi có sẵn

insert(position, string)

```
string s0("I you");
s0.insert(2, "love ");
cout<<s0;
```

insert(iterator_init, iterator_begin, iterator_end)

```
string s0("I you"); string s1("love ");
string::iterator it;
it = s0.begin() + 2;
s0.insert(it, s1.begin(), s1.begin() + 5);
cout<<s0;
```

I love you

Hàm chèn chuỗi/ký tự vào chuỗi có sẵn

String Python không có hàm chèn chuỗi vào chuỗi, nhưng ta có thể sử dụng cú pháp sau để chèn:

string0 = string0[:position] + string1 + string0[position:]

```
s0 = "I you"
s0 = s0[:2] + "love " + s0[2:]
print(s0)
```

I love you

Chèn chuỗi/ký tự vào chuỗi có sẵn

Java không có hàm insert, có thể chèn chuỗi/ký tự vào String bằng 1 trong 2 cách sau:

- Sử dụng insert của **StringBuilder.**
- Kết hợp cách lấy chuỗi con và nối chuỗi.

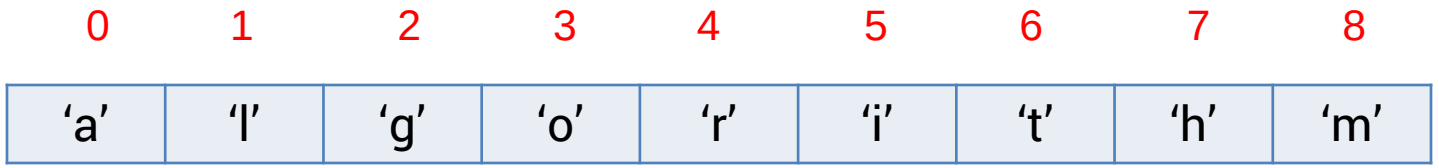
```
String s0 = "I you";
s0 = (new StringBuilder(s0)).insert(2, "love ").toString();
System.out.println(s0);
```

```
String s0 = "I you";
s0 = s0.substring(0, 2) + "love " + s0.substring(2);
System.out.println(s0);
```

Kết quả

I love you

Hàm xóa một đoạn trong chuỗi



C++

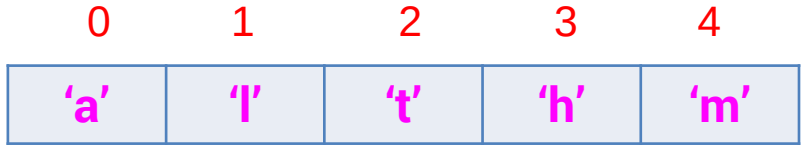
```
string s(“algorithm”);
s.erase(2, 4);
//Hoặc
s.erase(s.begin()+2, s.begin() + 6);
```

python

string = string[:start_position] + string[end_position:]

```
s = "algorithm"
s = s[:2] + s[6:]
```

Kết quả



Hàm xóa một đoạn trong chuỗi

Java không có hàm xóa một đoạn trong chuỗi. Nhưng tương tự như việc chèn, ta có thể kết hợp việc lấy chuỗi con và nối chuỗi để xóa:

```
s = s.substring(0, start_pos) + s.substring(end_pos);
```

0	1	2	3	4	5	6	7	8
'a'	'l'	'g'	'o'	'r'	'i'	't'	'h'	'm'

```
String s0 = "algorithm";
s0 = s0.substring(0, 2) + s0.substring(6);
```

Kết quả

0	1	2	3	4
'a'	'l'	't'	'h'	'm'

Tìm kiếm chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

C++

find(string)

```
string s0(“algorithm”);
string s1(“go”);
int pos = s0.find(s1);
if(pos==-1)
    cout<<“not found";
else
    cout<<pos;
```

python

find(string)

```
s0 = "algorithm"
s1 = "go"
pos = s0.find(s1)
if pos == -1:
    print("not found")
else:
    print(pos)
```

Kết quả

2

Tìm kiếm chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

indexOf(string);

```
String s0 = "algorithm";
String s1 = "go";
int pos = s0.indexOf(s1);
if (pos == -1)
    System.out.print("not found");
else
    System.out.print(pos);
```

Kết quả

2

Lấy chuỗi con

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

substr(pos, len)

```
string s0(“algorithm”);
string s1;
s1 = s0.substr(2, 4);
cout<<s1;
```

Python không hỗ trợ hàm substring, lấy thông qua index: variable_name [start_pos:end_pos]

```
s0 = "algorithm"
s1 = s0[2:6]
print(s1)
```

gori

Lấy chuỗi con

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

substring(start_index, end_index = length());

```
String s0 = "algorithm";
String s1 = s0.substring(2, 6);
System.out.println(s1);
```

gori

Nối chuỗi lại với nhau

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

C++

python

append(string, pos, len)

```
string s0(“algorithm”);
string s1(“the ”);
s1.append(s0, 2, 4);
cout<<s1;
```

Dùng phép “+” nối chuỗi

```
s0 = "algorithm"
s1 = "the "
s1 += s0[2:6]
print(s1)
```

Kết quả

the gori

Nối chuỗi lại với nhau

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

Dùng operator +

```
String s0 = "algorithm";
String s1 = "the ";
s1 += s0.substring(2,6);

System.out.print(s1);
```

Kết quả

the gori

So sánh chuỗi

compare(string): phân biệt hoa thường.

```
string s0("algorithm");
string s1("ALGORITHM");
if(s0.compare(s1)!=0) //if(s0!=s1)
    cout<<"Strings are not equal";
else
    cout<<"Strings are equal";
```

Dùng dấu == hoặc !=

```
s0 = "algorithm"
s1 = "ALGORITHM"
if s0 != s1:
    print("Strings are not equal")
else:
    print("Strings are equal")
```

Strings are not equal

So sánh chuỗi

Dùng phương thức equals(string)

```
String s0 = "algorithm";
String s1 = "ALGORITHM";
if (s0.equals(s1))
    System.out.println("Strings are equal");
else
    System.out.println("Strings are not equal");
```

Kết quả

Strings are not equal

Các hàm thành viên khác của string

size()/length(): Trả về số lượng phần tử hiện tại có trong chuỗi.

empty(): Kiểm tra chuỗi có rỗng hay không.

clear(): Xóa hết tất cả các giá trị trong chuỗi.

len(s): Trả về số lượng phần tử hiện tại trong chuỗi.

Sử dụng phép so sánh để kiểm tra chuỗi có rỗng.

if len(s) == 0:

Xóa hết tất cả các giá trị trong chuỗi: sử dụng phép gán.

```
s = ""
```

Các hàm thành viên khác của string

length(): Trả về số lượng phần tử hiện tại có trong chuỗi.

isEmpty(): Kiểm tra chuỗi có rỗng hay không.

Xóa hết tất cả các giá trị trong chuỗi: sử dụng phép gán.

```
s = "";
```

CÁC HÀM KIỂM TRA KÝ TỰ

Bảng mã ASCII

Big-O Blue

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciichars.com

Kiểm tra ký tự

Dùng mã ASCII kiểm tra

```cpp
#include <iostream>
using namespace std;
int main() {
    string s("123abc");
    if (s[0] > 48 && s[0] < 57)
        cout << "number";
    return 0;
}
```

Trong Python sử dụng hàm ord() để lấy mã của chuỗi có 1 kí tự.

Để chuyển mã ASCII thành chuỗi, sử dụng hàm chr().

```python
s = "123abc"
if ord(s[0]) > 48 and
ord(s[0]) < 57:
    print("number")
```

number

Kiểm tra ký tự

Dùng mã ASCII kiểm tra

```
String s = "123abc";
if (s.charAt(0) >= 48 && s.charAt(0) < 58)
    System.out.println("number");
```

Một số hàm kiểm tra ký tự

- isalpha
- isdigit
- islower
- isupper

Lưu ý: *python không chỉ kiểm tra từng ký tự mà có thể kiểm tra toàn bộ chuỗi.*

Một số hàm kiểm tra ký tự, C++ sử dụng thư viện <ctype.h>

Kiểm tra ký tự có phải chữ cái không

isalpha

Cú pháp: int result = **isalpha**(char c)

Kết quả trả về:

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

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

Kết quả **2**

isalpha

Cú pháp: result = s.**isalpha**()

Kết quả trả về:

- False: chuỗi rỗng hoặc ký tự đó không phải chữ cái.
- True: ký tự đó là ký tự chữ cái.

```
s = "Ky Thuat Lap Trinh"
result = s[1].isalpha()
print(result)
```

Kết quả **True**

Kiểm tra ký tự có phải chữ cái không

isLetter

Cú pháp: isLetter(character c). Thư viện: java.lang.Character

Kết quả trả về:

- false: ký tự đó không phải chữ cái.
- true: ký tự đó là chữ cái.

```
System.out.println(Character.isLetter('c'));
System.out.println(Character.isLetter('5'));
```


```
true
false
```

Kiểm tra ký tự có phải số không

isdigit

Cú pháp: int result = **isdigit**(char c)

Kết quả trả về:

- 0: ký tự đó không phải số.
- Khác 0: ký tự đó là số.

```
string s = "Thu 6";
int result = isdigit(s[4]);
cout<<result;
```

Kết quả → 4

isdigit

Cú pháp: result = s.**isdigit**()

Kết quả trả về:

- False: chuỗi rỗng hoặc ký tự đó không phải là số.
- True: ký tự đó là số.

```
s = "Thu 6"
result = s[4].isdigit()
print(result)
```

Kết quả → True

Kiểm tra ký tự có phải số không

isDigit

Cú pháp: isDigit(character c)

Kết quả trả về:

- false: ký tự đó không phải số.
- true: ký tự đó là số.

```
System.out.println(Character.isDigit('c'));
System.out.println(Character.isDigit('5'));
```

Kết quả

```
false
true
```

Kiểm tra ký tự có phải viết thường không

islower

Cú pháp: int result = **islower**(char c)

Kết quả trả về:

- 0: ký tự đó không phải viết thường.
- Khác 0: ký tự đó viết thường.

```
string s = "Thu 6";
int result = islower(s[1]);
cout<<result;
```

Kết quả → 2

islower

Cú pháp: result = s.**islower**()

Kết quả trả về:

- False: chuỗi rỗng hoặc ký tự đó không phải viết thường.
- True: ký tự đó viết thường.

```
s = "Thu 6"
result = s[1].islower()
print(result)
```

Kết quả → True

Kiểm tra ký tự có phải viết thường không

isLowerCase

Cú pháp: isLowerCase(character c)

Kết quả trả về:

- false: Ký tự đó không phải viết thường.
- true: Ký tự đó viết thường.

```
System.out.println(Character.isLowerCase('c'));
System.out.println(Character.isLowerCase('C'));
```


```
true
false
```

Kiểm tra ký tự có phải viết hoa không

isupper

Cú pháp: int result = **isupper**(char c)

Kết quả trả về:

- 0: ký tự đó không phải viết hoa.
- Khác 0: ký tự đó viết hoa.

```
string s = "Thu 6";
int result = isupper(s[0]);
cout<<result;
```

Kết quả **1**

isupper

Cú pháp: result = s. **isupper**()

Kết quả trả về:

- False: chuỗi rỗng hoặc ký tự đó không phải viết hoa.
- True: ký tự đó viết hoa.

```
s = "Thu 6"
result = s[0].isupper()
print(result)
```

Kết quả **True**

Kiểm tra ký tự có phải viết hoa không

isUpperCase

Cú pháp: isUpperCase(character c)

Kết quả trả về:

- false: Ký tự đó không phải viết hoa.
- true: Ký tự đó viết hoa.

```
System.out.println(Character.isUpperCase('c'));
System.out.println(Character.isUpperCase('C'));
```

CÁC HÀM CHUẨN HÓA TRONG STRING

Chuyển chuỗi thành kiểu số

atoi(char *str): Chuyển chuỗi thành số nguyên.

atof(char *str): Chuyển chuỗi thành số thực.

```
string s("12");
int number = atoi(s.c_str());
cout << number;
```

Chuyển chuỗi thành số nguyên:

int(string)

Chuyển chuỗi thành số thực:

float(string)

```
s = "12"
number = int(s)
print(number)
```

12

Chuyển chuỗi thành kiểu số

Integer.parseInt (string, base = 10)

```
String s = "12";
int number = Integer.parseInt(s);
// int number = Integer.parseInt(s, 10);
System.out.println(number);
```

Kết quả

12

Chuyển số thành chuỗi

to_string(value)

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

str(value)

```python
number = 15789
s = str(number)
print(s)
```

15789

Chuyển số thành chuỗi

Integer.toString(value)

```
int number = 15789;
String s = Integer.toString(number);
System.out.println(s);
```

15789

In hoa và in thường ký tự - dùng hàm

toupper(char c): Chuyển ký tự thành ký tự in hoa.

tolower(char c): Chuyển ký tự thành ký tự in thường.

```
string s("algorithm");
char c = toupper(s[2]);
cout << c;
```

upper(): Chuyển ký tự thành ký tự in hoa.

lower(): Chuyển ký tự thành ký tự in thường.

Lưu ý: trong Python 2 hàm này có thể chuyển nguyên chuỗi thành hoa/thường.

```
s = "algorithm"
c = s[2].upper()
print(c)
```

G

In hoa và in thường ký tự - dùng hàm

toUpperCase(): Chuyển ký tự thành ký tự in hoa.

toLowerCase(): Chuyển ký tự thành ký tự in thường.

```
String s = "algorithm";
char c = Character.toUpperCase(s.charAt(2));
System.out.print(c);
```

G

In hoa và in thường ký tự - ASCII

In hoa và In thường ký tự (**dùng mã ASCII**)

- ký tự - 32: Chuyển ký tự thành ký tự in hoa.
- Ký tự + 32: Chuyển ký tự thành ký tự in thường.

```
string s("algorithm");
s[2] = s[2] - 32;
cout << s[2];
```

G

Lưu ý: *In hoa và In thường ký tự (**dùng mã ASCII**). String trong Python là immutable, không thể cập nhật trực tiếp vào thành phần của string để gán.*

In hoa và in thường ký tự - ASCII

In hoa và In thường ký tự (**dùng mã ASCII**)

- ký tự - 32: Chuyển ký tự thành ký tự in hoa.
- Ký tự + 32: Chuyển ký tự thành ký tự in thường.

```
String s = "algorithm";
char c = s.charAt(2);
c -= 32;
System.out.print(c);
```

G

MỘT SỐ LƯU Ý KHI SỬ DỤNG STRING

Truy cập ngẫu nhiên vào chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

C++

python

Có thể truy cập vào thành phần của chuỗi để thao tác.

```
string s("algorithm");
s[0] = 'A';
s[8] = 'M';
char c = s[6];
cout<<s<<" "<<c;
```

Kết quả

AlgorithM t

Lưu ý: *In hoa và In thường ký tự (**dùng mã ASCII**). String trong Python là immutable, không thể cập nhật trực tiếp vào thành phần của string để gán.*

Truy cập ngẫu nhiên vào chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

String trong java là immutable. Để thay đổi giá trị thì ta cần sử dụng StringBuilder.

```
StringBuilder s = new StringBuilder("algorithm");
s.setCharAt(0, ‘A’);
s.setCharAt(8, 'M’);
Character c = s.charAt(6);
System.out.print(s);
System.out.print(' ');
System.out.print(c);
```

Kết quả

AlgorithM t

Duyệt chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’

C++

python

```cpp
for(int i=0; i<s.size(); i++)
{
    cout << s[i];
}
```

```cpp
string::iterator it;
for(it=s.begin(); it!=s.end(); it++)
{
    cout<<*it;
}
```

```python
s = "algorithm"
for i in range(len(s)):
    print(s[i], end='')
```

```python
s = "algorithm"
for ch in s:
    print(ch, end='')
```

Kết quả

algorithm

Duyệt chuỗi

0	1	2	3	4	5	6	7	8
‘a’	‘l’	‘g’	‘o’	‘r’	‘i’	‘t’	‘h’	‘m’


```java
String str = "algorithm";
for (int i = 0, n = str.length(); i < n; i++) {
    System.out.print(str.charAt(i));
}
```

algorithm

Đọc từng từ và nguyên dòng

nothing is impossible


```
string s0;
cin >> s0;
cout << s0;
```

nothing

```
string s1;
getline(cin, s1);
cout << s1;
```

nothing is impossible

Đọc từng từ và nguyên dòng

nothing is impossible

Trong Python mặc định sẽ đọc theo từng dòng nên không có sự phần biệt giữa đọc từng từ và đọc nguyên dòng. Để phần tách lấy từng từ thì ta có thể dùng hàm split sau khi đọc.

```
line = input().split()
s0 = line[0]
print(s0)
```

nothing

Đọc từng từ và nguyên dòng

nothing is impossible


```
Scanner sc = new Scanner(System.in);
String s0 = sc.next();
System.out.print(s0);
```

nothing

```
Scanner sc = new Scanner(System.in);
String s1 = sc.nextLine();
System.out.print(s1);
```

nothing is impossible

Hỏi đáp