



KHOA CÔNG NGHỆ THÔNG TIN

Faculty of Computer Science and Engineering-Thuyloi University

# LẬP TRÌNH NÂNG CAO

Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Bộ môn: Tin học và KTTT

Năm học: 2020-2021

## Chương 2: Mảng và chuỗi ký tự

### 2.1. Kiểu mảng

2.1.1. Mảng dữ liệu một chiều, hai chiều

2.1.2. Tìm kiếm và sắp xếp dữ liệu trong mảng

2.1.3. Truyền tham số là mảng

### 2.2. **Kiểu chuỗi ký tự**

2.2.1. Chuỗi ký tự và các phép toán trên chuỗi

2.2.2. Một số bài toán trên chuỗi ký tự

2.2.3. Chèn/ xóa chuỗi ký tự

# Xâu kí tự



*“Ngôn ngữ lập trình C++”*

*“Hôm nay trời rất đẹp!”*





# Xâu kí tự

- Nhiều phần mềm có nhu cầu xử lý dãy các kí tự, chẳng hạn như làm việc với tên của khách hàng, địa chỉ, email, chức vụ công tác,...
  - Xuất hiện nhu cầu xử lý các kí tự theo loạt
- Thời kì ban đầu, các lập trình viên tự tạo kiểu dữ liệu **string**, bản chất là dãy các kí tự, để xử lý các nhu cầu đó
- Khi việc sử dụng trở nên quá phổ biến, người ta chuẩn hóa đưa vào trong thư viện của C++
- “string” là kiểu dữ liệu tự tạo phổ biến nhất

# Khái niệm chuỗi ký tự

'T'	'i'	' n '	' '	'h'	'o'	'c'	'\0'
-----	-----	-------	-----	-----	-----	-----	------

- Chuỗi ký tự (string) là một dãy các ký tự viết liên tiếp nhau
  - Độ dài chuỗi là số ký tự có trong chuỗi
  - Chuỗi không có ký tự nào: Chuỗi rỗng
- Ví dụ: "Tin học", "String"
- Lưu trữ: kết thúc chuỗi bằng ký tự '\0' hay NULL (mã ASCII là 0)

## ❑ Chuỗi ký tự:

- Là một mảng (dãy) các ký tự liên tiếp nhau
- Kết thúc bằng ký tự '\0' là ký tự NULL

# Khái niệm chuỗi ký tự → Lưu ý

## ➤ Chuỗi ký tự $\neq$ mảng ký tự

- Tập hợp các ký tự viết liên tiếp nhau
  - Truy nhập một phần tử của chuỗi ký tự (*là một ký tự*) giống như truy nhập vào một phần tử của mảng:

Tên\_chuỗi [Chỉ\_số]

- Chuỗi ký tự có ký tự kết thúc chuỗi, mảng ký tự không có ký tự kết thúc chuỗi
- Chuỗi ký tự độ dài 1  $\neq$  ký tự ("A"  $\neq$  'A' ?)
  - 'A' là 1 ký tự, được lưu trữ trong 1 byte
  - "A" là 1 chuỗi ký tự, ngoài ký tự 'A' còn có ký tự '\0'  $\Rightarrow$  được lưu trữ trong 2 byte

# Xâu kí tự

Ví dụ: Xâu ký tự “Ngon ngu C++”

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 chuỗi 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 chuỗi = chỉ được dùng khi khai báo biến***

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

# Khai báo xâu kí tự: Cách 2

## Cách 2: Sử dụng lớp **string**

Muốn sử dụng, cần: **#include <string>**

```
string tenxau;
```

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

### ■ Ví dụ:

```
string str;
```

```
str = "Xin chào!";
```

```
string xau = "Xin chào!";
```

# 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
<b>tolower(int ch)</b>	Chuyển thành kí tự thường
<b>toupper(int ch)</b>	Chuyển thành kí tự hoa
<b>islower(int ch)</b>	Kiểm tra chữ thường
<b>isupper(int ch)</b>	Kiểm tra chữ hoa
<b>isdigit(int ch)</b>	Kiểm tra chữ số
<b>isalpha(int ch)</b>	Kiểm tra xem kí tự có là chữ cái không
<b>isspace(int ch)</b>	Kiểm tra kí tự dấu cách
<b>isctrl(int ch)</b>	Kiểm tra kí tự điều khiển

## Một số ví dụ

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

# Ví dụ: Nhập vào một chuỗi 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 chuỗi kí tự:";
8      getline(cin, str);
9      int i, dem=0;
10     for(i=0;i<=str.size(); i++)
11     {
12         if(isupper(str[i]))//Kiểm tra nếu là kí tự hoa
13             dem++;
14     }
15     cout<<"Số kí tự hoa là:"<<dem;
16     return 0;
17 }
```

# Chuyển xâu thành chữ thường

Ví dụ: Nhập vào một xâu kí tự. Chuyển xâu đó thành chữ thường

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main ()
6  {
7      string s;
8      cout<<"Nhap vao mot xau: "<<endl;
9      getline(cin, s);
10
11     for(int i=0; i<s.size();i++)
12         s[i] = tolower(s[i]);
13
14     cout<<"\nChuyen thanh chu thuong:\n"<< s;
15     return 0;
16 }
```

# Một số ví dụ

- Ví dụ: Nhập vào một chuỗi kí tự và một kí tự. Đếm số lần xuất hiện của kí tự đó trong chuỗi.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main ()
5  {
6      string s; char c;
7      cout<<"Nhập vào một chuỗi: "<<endl;
8      getline(cin, s);
9      cout<<"Nhập vào một kí tự ";
10     cin>>c;
11     int dem =0;
12     for(int i=0; i<s.size();i++)
13         if(s[i]==c)
14             dem++;
15
16     cout<<"\nSố lần xuất hiện kí tự "<<c<<" là:"<<dem;
17     return 0;
18 }
```

TS. Bùi Thị Thanh Xuân



# Một số ví dụ

- **Ví dụ:** Lập trình đọc vào một xâu chuẩn từ bàn phím rồi đưa màn hình dưới dạng cột.

Ví dụ:

Dai Hoc Thuy Loi

Đưa ra:

Dai  
Hoc  
Thuy  
Loi

# Một số ví dụ

## Ví dụ:

Lập trình đọc vào một  
xâu chuẩn từ bàn phím  
rồi đưa màn hình dưới  
dạng cột.

## Ví dụ:

Dai Hoc Thuy Loi

Đưa ra:

Dai  
Hoc  
Thuy  
Loi

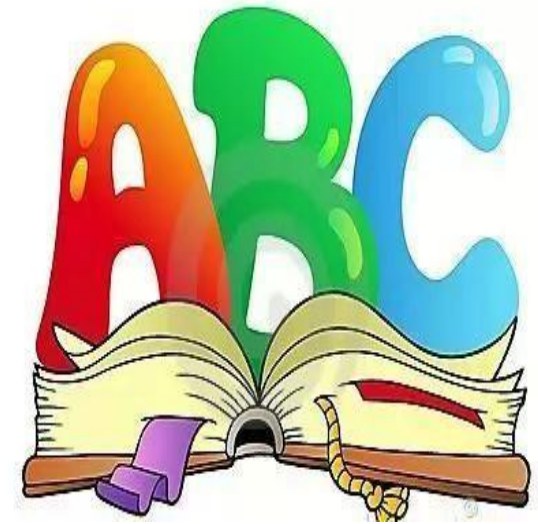
```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s;
    cout<<"Nhap vao mot xau (chuan): "<<endl;
    getline(cin, s);

    for(int i=0; i<s.size();i++)
        if(s[i]==' ')
            cout<<endl;
        else
            cout<<s[i];
    return 0;
}
```

# Một số thao tác với âm

- So sánh âm
- Ghép âm
- Chèn âm
- Xóa âm
- Tìm kiếm âm kí tự



# Các phép toán và phương thức cơ bản

Phép toán/Phương thức	Mô tả
<code>+</code> , <code>+=</code>	Ghép 2 chuỗi xâu hoặc ghép một kí tự vào xâu
<code>==</code> , <code>!=</code> , <code>&gt;</code> , <code>&gt;=</code> , <code>&lt;</code> , <code>&lt;=</code>	So sánh theo thứ tự từ điển
<code>.length()</code> , <code>.size()</code>	Trả về độ dài của xâu
<code>.clear()</code>	Xóa nội dung của xâu
<code>.append(str)</code>	Thêm các kí tự/xâu vào cuối xâu hiện tại
<code>.insert(pos, str)</code>	Chèn các kí tự/xâu con vào xâu tại vị trí bất kì
<code>.replace(pos, len, str)</code>	Thay thế xâu con trong xâu hiện tại bằng 1 xâu con mới
<code>.substr(pos, len)</code>	Trích xâu con từ xâu ban đầu
<code>.compare(str)</code>	So sánh xâu với xâu hiện tại
<code>.find(str)</code>	Tìm xâu con trong xâu hiện tại

# Ghép xâu

## ▪ Dùng toán tử +

```
string str1 = "Hello. ";  
string str2;  
str2 = "How are you?";  
str1 = str1 + str2;  
cout<<str1;
```

Hello. How are you?

## ▪ Dùng phương thức **append**

```
string str1 = "Hello. ";  
str1.append("How are you?");  
cout<<str1;
```

Hello. How are you?

# Chèn xâu

Cú pháp	Mô tả
<code>.insert(pos, str2)</code>	
<code>.insert(pos, str2, subpos, sublen)</code>	
<code>.insert(pos, n, c)</code>	

- **pos:** vị trí cần chèn
- **str2:** xâu chèn vào xâu gốc
- **subpos:** vị trí kí tự đầu tiên trên xâu *str2* được chèn
- **sublen:** độ dài xâu con cần chèn
- **n:** số kí tự cần chèn
- **c:** kí tự cần chèn

# Xóa sâu

Cú pháp	Mô tả
<code>.erase (pos, len)</code>	
<code>.clear()</code>	

- **pos:** vị trí bắt đầu xóa
- **len:** số kí tự xóa

# Tìm kiếm chuỗi

Cú pháp	Mô tả
<code>.find(str)</code>	
<code>.find(str, pos)</code>	

- **str:** chuỗi kí tự/ kí tự cần tìm
- **pos:** vị trí bắt đầu tìm



# Xâu kí tự - Hàm chuyển đổi xâu – số

- **atoi(str), atol(str), atof(str)**: hàm trả về số nguyên kiểu int, long và số thực float.
- Hàm trả về 0 nếu trong xâu không chứa số nào
- int **atoi**(const char \* str):
  - Chuyển một xâu kí tự thành một số nguyên tương ứng
  - Ví dụ: atoi("1234") → 1234
- int **atol**(const char \* str):
  - Chuyển xâu kí tự thành số long int
- float **atof**(const char \* str):
  - Chuyển xâu kí tự thành số thực
  - Ví dụ: atof("123.456E-2") → 1.23456
- Thất bại cả 3 hàm: trả về 0

# Xâu kí tự - Hàm chuyển đổi xâu – số

- **atoi(str), atol(str), atof(str)**: hàm trả về số nguyên kiểu int, long và số thực float.
- Hàm trả về 0 nếu trong xâu không chứa số nào

```
1  #include <iostream>
2  #include <string>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      cout<<"Chuyen xau '123' sang so: "<<atoi("123");
9      cout<<"\nChuyen xau '123L' sang so: "<<atol("123L");
10     cout<<"\nChuyen xau '123.123' sang so: "<<atof("123.123");
11     return 0;
12 }
```

```
Chuyen xau '123' sang so: 123
Chuyen xau '123L' sang so: 123
Chuyen xau '123.123' sang so: 123.123
-----
```

# Đếm số lần xuất hiện của một xâu

Nhập vào một xâu kí tự và một xâu con. Đếm số lần xuất hiện của xâu con đó trong xâu.

# Đếm số lần xuất hiện của một chuỗi

**Ví dụ:** Nhập vào một chuỗi kí tự và một chuỗi con. Đếm số lần xuất hiện của chuỗi con đó trong chuỗi.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main ()
5  {
6      string s1, s2;
7      cout<<"Nhập vào một chuỗi gốc:"; getline(cin,s1);
8      cout<<"Nhập vào một chuỗi con:"; getline(cin, s2);
9      int i=0, dem=0, pos;
10     while(i<s1.size())
11     {
12         pos = s1.find(s2,i);
13         if(pos<0)
14             break;
15         else
16         {
17             dem++;
18             i=pos + s2.size();
19         }
20     }
21     cout<<"Số lần xuất hiện của chuỗi '"<<s2<<"' là:"<<dem;
22     return 0;
23 }
```

# Đếm số lần xuất hiện của một xâu

**Ví dụ:** Nhập vào một xâu kí tự và một xâu con. Đếm số lần xuất hiện của xâu con đó trong xâu.

```
void nhap_xau(string &s)
void hien_thi(string s)
int tim_xau(string s, string w)
int main()
{
    string w,s;
    nhap_xau(s);
    nhap_xau(w);
    cout<<"So lan xuat hien xau '"<<s<<"' trong xau '"<<w<<"'
    la: "<<tim_xau(s,w);
    return 0;
}
```

```
Nhap xau: xuan
Nhap xau: xuan oi xuan da ve
So lan xuat hien xau 'xuan' trong xau 'xuan oi xuan da ve' la: 2
-----
```

# Đếm số lần xuất hiện của một chuỗi

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

void nhap_xau(string &s)
{
    cout<<"Nhập chuỗi: "; getline(cin,s);
}

void hien_thi(string s)
{
    cout<<"In chuỗi: "<<s;
}

int main()
{
    string w,s;
    nhap_xau(s);
    nhap_xau(w);
    cout<<"Số lần xuất hiện chuỗi '"<<s<<"' trong chuỗi '"<<w<<"' là: "<<tim_xau(s,w);
    return 0;
}
```

```
int tim_xau(string s, string w)
{
    int dem = 0;
    if(w.size()>0 && s.size()>0)
    {
        int k = 0, vt;
        while(k < w.size())
        {
            vt = w.find(s,k);
            if(vt<0)
                break;
            else
            {
                dem++;
                k = vt + s.size();
            }
        }
    }
    return dem;
}
```

# Bài tập



- ☐ Bài 1: Lập trình đọc vào một xâu và đưa ra màn hình số từ của xâu đó.
- ☐ Bài 2: Nhập vào 1 xâu ký tự. Kiểm tra tính đối xứng của xâu. Đưa ra xâu đảo ngược

# Bài tập



Bài 1: Lập trình đọc vào một câu và đưa ra màn hình số từ của câu đó.

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

int dem_tu(string s)
{
    int dem = 0;
    for(int i=0; i<s.size();i++)
        if(s[i]!=' ' && s[i+1] == ' ' || s[i] != ' ' && s[i+1] == '\0')
            dem++;
    return dem;
}

int main()
{
    string s;
    cout<<"Nhập vào S: ";getline(cin,s);
    cout<<"Số từ trong câu = "<<dem_tu(s);
    return 0;
}
```



# Bài tập



Bài 2: Nhập vào 1 xâu ký tự. Kiểm tra tính đối xứng của xâu.  
Đưa ra xâu đảo ngược.

```
bool xau_doi_xung(string s)
{
    int i, n=s.length();
    for(i=0; i<n/2; i++)
        if(s[i]!=s[n-1-i])
            return false;
    return true;
}
```

```
string dao_nguoc(string s)
{
    int n=s.length();
    for(int i=0; i<n/2; i++){
        int c = s[i];
        s[i] = s[n-i-1];
        s[n-i-1]=c;
    }
    return s;
}
```

```
int main(){
    string s;
    cout<<"Nhập vào xâu kí tự s:";getline(cin, s);
    if(xau_doi_xung(s))
        cout<<"Xâu đối xứng!";
    else
        cout<<"Xâu không đối xứng!";
    cout<<"\nXâu đảo ngược: "<<dao_nguoc(s);
    return 0;
}
```



# Bài tập

Bài 3: Một chuỗi được gọi là chuỗi chuẩn nếu đầu và cuối chuỗi không chứa dấu cách đồng thời trong chuỗi không chứa hai dấu cách liên nhau. Lập trình đọc vào một chuỗi và đưa ra màn hình chuỗi chuẩn.

# Bài tập



Bài 3: Một xâu được gọi là xâu chuẩn nếu đầu và cuối xâu không chứa dấu cách đồng thời trong xâu không chứa hai dấu cách liên nhau. Lập trình đọc vào một xâu và đưa ra màn hình xâu chuẩn.

```
string chuan_hoa(string s)
{
    while(s[0]==' ')
        s.erase(0,1);
    while(s[s.size()-1]==' ')
        s.erase(s.size()-1,1);
    int i = 0;
    while(i<s.size())
        if(s[i]==' ' && s[i+1]==' ')
            s.erase(i,1);
        else
            i++;
    return s;
}
```

```
int main()
{
    string s;
    cout<<"Nhap xau S:"; getline(cin,s);
    cout<<"Do dai xau S ban dau: "<<s.size();
    cout<<"\nChuan hoa S:"<<chuan_hoa(s);
    cout<<"\nDo dai xau S sau khi chuan hoa: "<<chuan_hoa(s).size();
    return 0;
}
```



## Bài tập

Bài 4. Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
Nhap DSSV (<100), go Enter de thoat..  
Nhap ten sinh vien: Anh  
Nhap ten sinh vien: Ha  
Nhap ten sinh vien: Xuan  
Nhap ten sinh vien: Dung  
Nhap ten sinh vien:  
  
DSSV da sap xep la:  
  
Sinh vien[1]: Anh  
Sinh vien[2]: Dung  
Sinh vien[3]: Ha  
Sinh vien[4]: Xuan  
-----
```



## Chữa Bài tập: Sử dụng mảng và chuỗi ký tự

Bài 4. Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
void nhap_DS (string ds[], int &n)
void sap_xep (string ds[], int n)
void hien_thi_DS (const string ds[], int n)
int main()
{
    int n;
    string ds[100];
    nhap_DS(ds,n);
    sap_xep(ds,n);
    hien_thi_DS(ds,n);
    return 0;
}
```



## Chữa Bài tập: Sử dụng mảng và chuỗi ký tự

Bài 4. Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
void nhapDS(string ds[], int &n){
    string s; n = 0;
    cout<<"Nhap DSSV (<100), go Enter de thoat..\n";
    do{
        cout<<"Nhap ten sinh vien: "; getline(cin,s);
        if(s.size()==0 || n==100)
            break;
        ds[n] = s;
        n++;
    }while(1);
}

void sap_xep(string ds[], int n){
    for(int i = 0; i<n-1; i++)
        for(int j = i+1; j<n; j++)
            if(ds[i]>ds[j])
            {
                string s = ds[i];
                ds[i] = ds[j];
                ds[j] = s;
            }
}
```

```
void hien_thi_DS(const string ds[], int n)
{
    cout<<"\nDSSV da sap xep la: \n";
    for(int i = 0; i<n; i++)
        cout<<"\nSinh vien["<<i+1<<"]: "<<ds[i];
}

int main(){
    int n;
    string ds[1000];
    nhapDS(ds,n);
    sap_xep(ds,n);
    hien_thi_DS(ds,n);
    return 0;
}
```



## Bài tập

Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
Nhap DSSV (<100), go Enter de thoat..  
Nhap ten sinh vien: Anh  
Nhap ten sinh vien: Ha  
Nhap ten sinh vien: Xuan  
Nhap ten sinh vien: Dung  
Nhap ten sinh vien:  
  
DSSV da sap xep la:  
  
Sinh vien[1]: Anh  
Sinh vien[2]: Dung  
Sinh vien[3]: Ha  
Sinh vien[4]: Xuan  
-----
```



## Chữa Bài tập: Sử dụng vector và chuỗi ký tự

Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
void nhapDS(vector <string> &ds)
void sap_xep(vector <string> &ds)
void hien_thi_DS(const vector
<string> ds)

int main(){
    vector <string> ds;
    nhapDS(ds);
    sap_xep(ds);
    hien_thi_DS(ds);
    return 0;
}
```

```
Nhap DSSV, go Enter de thoat..
Nhap ten sinh vien: An
Nhap ten sinh vien: Xuan
Nhap ten sinh vien: Mai
Nhap ten sinh vien: Nhan
Nhap ten sinh vien:
DSSV da sap xep la:
Sinh vien [1]: An
Sinh vien [2]: Mai
Sinh vien [3]: Nhan
Sinh vien [4]: Xuan
```





## Chữa Bài tập: Sử dụng vector và xâu ký tự

Nhập vào danh sách tên sinh viên cho tới khi gặp tên rỗng, in ra DS sinh viên sắp xếp theo thứ tự abc.

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
void nhapDS(vector <string> &ds){
    string s;
    cout<<"Nhap DSSV, go Enter de thoat..\n";
    do{
        cout<<"Nhap ten sinh vien: "; getline(cin,s);
        if(s.size()!=0) ds.push_back(s);
    }while(s.size()!=0);
}

void sap_xep(vector <string> &ds){
    for(int i = 0; i<ds.size()-1; i++)
        for(int j = i+1; j<ds.size(); j++)
            if(ds[i]>ds[j])
            {
                string s = ds[i];
                ds[i] = ds[j];
                ds[j] = s;
            }
}
```

```
void hien_thi_DS(const vector <string> ds)
{
    cout<<"\nDSSV da sap xep la: \n";
    for(int i = 0; i<ds.size(); i++)
        cout<<"\nSinh vien ["<i+1<<"]: "<ds[i];
}

int main(){
    vector <string> ds;
    nhapDS(ds);
    sap_xep(ds);
    hien_thi_DS(ds);
    return 0;
}
```

```
Nhap DSSV, go Enter de thoat..
Nhap ten sinh vien: An
Nhap ten sinh vien: Xuan
Nhap ten sinh vien: Mai
Nhap ten sinh vien: Nhan
Nhap ten sinh vien:
DSSV da sap xep la:
Sinh vien [1]: An
Sinh vien [2]: Mai
Sinh vien [3]: Nhan
Sinh vien [4]: Xuan
```



## Bài tập áp dụng

**Quy ước:** Xâu họ và tên không chứa các dấu cách đầu và cuối xâu, trong đó họ là dãy kí tự từ đầu cho đến khi gặp dấu cách đầu tiên và tên là dãy kí tự từ sau dấu cách cuối cùng đến kí tự cuối xâu.

**Viết chương trình:**

- Nhập vào 1 danh sách họ tên của n sinh viên ( $2 < n < 50$ )
- Chuẩn hóa DS: Chữ cái đầu từ viết hoa, còn lại viết chữ thường, không thừa dấu cách đầu cuối, không có 2 dấu cách liên nhau,...
- Sắp xếp danh sách họ tên SV theo thứ tự abc
- Nhập vào 1 tên s (xâu ký tự ko chứa dấu cách). Đếm xem trong DS có bao nhiêu bạn SV tên s.



# Bài tập áp dụng

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

string chuan_hoa_ten(string s)
void nhapDS(string ds[], int &n)
void sap_xep(string ds[], int n)
void hien_thi_DS(const string ds[], int n)
string tach_ten(string hoten)
int dem_ten(string ten, const string ds[], int n)
int main()
```

# Bài tập áp dụng

```
string chuan_hoa_ten(string s)
void nhapDS(string ds[], int &n)
void sap_xep(string ds[], int n)
void hien_thi_DS(const string ds[], int n)
string tach_ten(string hoten)
int dem_ten(string ten, const string ds[], int n)
int main(){
    int n;
    string ds[100],ten;
    nhapDS(ds,n);
    sap_xep(ds,n);
    hien_thi_DS(ds,n);
    for(int i = 0; i<n; i++)
        cout<<"\nTen ["<<i+1<<"]: "<<tach_ten(ds[i]);
    cout<<"\nCho ten SV: "; cin>>ten;
    cout<<"\nSo SV co ten '"<<ten<<"' la: "<<dem_ten(ten,ds,n);
    return 0;
}
```



# Bài tập áp dụng



```
string chuan_hoa_ten(string s){
    s = " "+s;
    for(int i = 1; i<s.size();i++)
        s[i] =tolower(s[i]);
    for(int i = 1; i<s.size();i++)
        if(s[i-1]==' ' && s[i]!=' ')
            s[i] =toupper(s[i]);
    while(s[0]==' ')
        s.erase(0,1);
    while(s[s.size()-1]==' ')
        s.erase(s.size()-1,1);
    int i = 0;
    while(i<s.size())
        if(s[i]==' ' && s[i+1]!=' ')
            s.erase(i,1);
        else
            i++;
    return s;
}
```

```
string tach_ten(string hoten)
{
    string ten;
    hoten = " "+hoten;
    int n = hoten.length();
    for(int i=n-1;i>=0;i--)
    {
        if(hoten[i]==' ')
        {
            for(int j=0;j<n-1-i;j++)
                ten += hoten[i+1+j];
            break;
        }
    }
    return ten;
}
```

```
int dem_ten(string ten, const string ds[], int n){
    int dem = 0;
    for(int i = 0; i<n; i++)
        if(ten == tach_ten(ds[i]))
            dem++;
    return dem;
}
```

```
void nhapDS(string ds[], int &n)
{
    do
    {
        cout<<"Nhap so sinh vien n = "; cin>>n;
    }while(n<=2 || n>=50);
    cin.ignore();
    for(int i=0; i<n;i++)
    {
        string s;
        cout<<"Ten sinh vien ["<<i+1<<"]: "; getline(cin,s);
        ds[i]=chuan_hoa_ten(s);
    }
}
```



# Bài tập áp dụng

```

void sap_xep(string ds[], int n)
{
    for(int i = 0; i<n-1; i++)
        for(int j = i+1; j<n; j++)
            if(ds[i]>ds[j])
            {
                string s = ds[i];
                ds[i] = ds[j];
                ds[j] = s;
            }
}

void hien_thi_DS(const string ds[], int n)
{
    cout<<"\nDS sinh vien la: \n";
    for(int i = 0; i<n; i++)
        cout<<"\nTen sinh vien["<i+1<<"]: "<ds[i];
}

int main(){
    int n;
    string ds[100],ten;
    nhapDS(ds,n);
    sap_xep(ds,n);
    hien_thi_DS(ds,n);
    for(int i = 0; i<n; i++)
        cout<<"\nTen ["<i+1<<"]: "<tach_ten(ds[i]);
    cout<<"\nCho ten SV: "; cin>>ten;
    cout<<"\nSo SV co ten '"<ten<<"' la: "<dem_ten(ten,ds,n);
    return 0;
}

```

```

Nhap so sinh vien n = 4
Ten sinh vien [1]: hoang lan
Ten sinh vien [2]: mai linh
Ten sinh vien [3]: ha ANH
Ten sinh vien [4]: NGUYEN MAI

DS sinh vien la:

Ten sinh vien[1]: Ha Anh
Ten sinh vien[2]: Hoang Lan
Ten sinh vien[3]: Mai Linh
Ten sinh vien[4]: Nguyen Mai
Ten [1]: Anh
Ten [2]: Lan
Ten [3]: Linh
Ten [4]: Mai
Cho ten SV: Linh

So SV co ten 'Linh' la: 1

```



# Bài tập

Nhập chuỗi ký tự W và chuỗi S, nhập số nguyên p và q. Hãy chèn chuỗi S vào chuỗi W ở vị trí p và q, sau đó in ra chuỗi kết quả. Nếu p hoặc q lớn hơn độ dài của W thì không thực hiện việc chèn và thoát luôn khỏi chương trình, không cần in gì thêm.

Hãy xem ví dụ dưới đây để biết quy tắc in thông tin ra màn hình.

```
Nhap W = abcdef
Nhap S = +++
P = 3
Q = 1
Ket qua: a+++bc+++def
-----
Process exited after 15.47 seconds with return value 0
Press any key to continue . . .
```



# Bài tập

```
int main()
{
    string w,s;
    int p,q;
    cout<<"Nhap W = ";getline(cin, w);
    cout<<"Nhap S = ";getline(cin, s);
    cout<<"P = "; cin >>p;
    cout<<"Q = "; cin >>q;
    if ((p<= w.size())&& (q<= w.size()))
    {
        if(p>q)
        {
            w.insert(p,s);
            w.insert(q,s);
        }
        else
        {
            w.insert(q,s);
            w.insert(p,s);
        }
        cout<<"Ket qua: "<<w;
    }
    return 0;
}
```





# Bài tập

- Nhập xâu kí tự W, hãy chuẩn hóa xâu W bằng cách áp dụng tất cả các quy tắc dưới đây:
  - Xóa bỏ các kí tự trống (space) ở đầu hoặc cuối xâu
  - Nếu nhiều kí tự trống viết liền nhau thì xóa bớt chỉ để lại đúng 1 kí tự

Hãy xem ví dụ dưới đây để biết quy tắc in thông tin ra màn hình.

```
Nhap W =   DAI   HOC   THUY LOI
Chuan hoa = DAI HOC THUY LOI
-----
Process exited after 4.981 seconds with return value 0
Press any key to continue . . . _
```

# Chữa bài tập

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s;
    cout<<"Nhap W = "; getline(cin,s);
    while(s[0]==' ') s.erase(0,1);
    while(s[s.size()-1]==' ') s.erase(s.size()-1,1);
    for (int i=0; i<s.size(); i++)
        while(s[i]==' ' && s[i+1]==' ')
        {
            s.erase(i,1);
        }
    cout<<"Chuan hoa = "<<s;
    return 0;
}
```