

Các hàm string xử lý xâu C++ (thường sẽ được cài đặt trong thư viện string, nếu như không chạy được thì chuyển sang sử dụng thêm thư viện string)

1. Hàm push_back và pop_back

Hàm **push_back()** sẽ thêm 1 ký tự vào cuối string và **pop_back()** sẽ xóa đi ký tự cuối cùng trong string.

Lưu ý là hàm này sẽ thêm 1 ký tự chứ không phải là thêm 1 chuỗi ký tự, nếu bạn cố tính push_back 1 chuỗi ký tự sẽ gây lỗi.

VD:

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

int main(){
    string s = "Huy 123";
    s.push_back('@');
    cout << s << endl;
    s.push_back('#');
    cout << s << endl;
    s.pop_back();
    cout << s << endl;
}
```

Question: Nhập vào một chuỗi s và 1 số nguyên n từ bàn phím, chèn vào cuối chuỗi ký tự có giá trị bằng ký tự có chỉ số n trong chuỗi (kiểm tra vị trí của n)

VD: python 3 -> pythonh

2. Hàm append: Cho phép nối chuỗi ký tự vào chuỗi ký tự (giống hàm ở phần 1 nhưng cho phép lồng cả 1 chuỗi ký tự)

VD:

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "Huy 12345678";
    //Nối toàn bộ xâu
    s.append("blog");
```

```

cout << s << endl;
string t = "0123456789";
//Nối 5 kí tự tính từ chỉ số 1 của xâu t vào s
s.append(t, 1, 5);
cout << s << endl;
}

```

- Có thể sử dụng toán tử += để thay thế 2 phương thức trên

3. Hàm insert(): chèn 1 xâu khác vào vị trí bất kỳ trong xâu hiện tại thông qua chỉ số chèn

- Cú pháp: `s.insert(index, string t);`

VD:

```

#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "Huyhuyhuyhuyhuy";
    //Chen vao chi so 2 cua xau s
    s.insert(2, "#####");
    cout << s << endl;
}

```

```
//Chen vao dau xau s
s.insert(0, "blog ");
cout << s << endl;
}
```

4. Hàm erase: Xóa đi 1 hoặc 1 chuỗi ký tự của xâu hiện tại thông qua chỉ số

Cú pháp: `s.erase(a,[b])`

+ a là vị trí bắt đầu xóa

+ b là số lượng ký tự cần xóa. Nếu không truyền vào thì mặc định là xóa đến hết chuỗi

VD:

```
#include <iostream>
#include <string>

using namespace std;

int main(){
    string s = "Huy";
    //Xoa tu chi so 1 toi het xau
    s.erase(1);
    cout << s << endl;
```

```
string t = "28tech.com.vn";  
//Xoa 4 ki tu tinh tu chi so 2  
t.erase(2, 4);  
cout << t << endl;  
}
```

Question: Cho 1 xâu s và 1 ký tự x được nhập vào từ bàn phím.

Xóa ký tự x đầu tiên và thứ 2 trong chuỗi.

5. Hàm substr: Tách xâu con của xâu hiện tại, trả về xâu con tách được.

Cú pháp: `s.substr(a, [b]);`

Định nghĩa tương tự phần erase.

VD:

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

```

int main(){
    string s = "abcdefxyz";
    //Xau con tu chi so 2 toi het xau
    string sub1 = s.substr(2);
    cout << sub1 << endl;
    string t = "adobe photoshop";
    //Xau con tu chi so 0 co do dai 6
    string sub2 = t.substr(0, 6);
    cout << sub2 << endl;
}

```

6. So sánh 2 string: sử dụng hàm compare()

VD:

```

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

int main(){
    string s = "Huy", t = "hUy", p = "huY";
    cout << s.compare(p) << endl;
    cout << s.compare(t) << endl;
    cout << t.compare(p) << endl;
    return 0;
}

```

Sẽ trả về 0 nếu cả 2 bằng nhau

Trả về -1 nếu $a < b$ (hoặc 1 số âm)

Trả về 1 nếu $a > b$ (hoặc 1 số dương)

Question: Nhập vào 2 chuỗi A và B. Xác định xem chuỗi này có phải chuỗi con của chuỗi kia không ? (có phân biệt in thường và in hoa)

VD:

Huy

Huy dep trai -> co

Bui Xuan Huan

Kha Binh -> khong

7. Hàm find()

- Tìm kiếm xâu con trong xâu hiện tại
- Nếu có sẽ trả về chỉ số đầu tiên của xâu con, ngược lại sẽ trả về `string::npos` (phương thức kiểm tra: `s.find("....." == string::npos`)
- VD:

```
- #include <iostream>  
- #include <string>  
-
```

```

- using namespace std;
-
- int main(){
-     string s = "Lap trinh C";
-     if(s.find("Lap trinh") == string::npos){
-         cout << "NOT FOUND\n";
-     }
-     else{
-         cout << "FOUND\n";
-         cout << s.find("Lap trinh");
-     }
- }

```

Question: Cũng giống phần 6 nhưng dùng hàm find()

8. Chuyển đổi xâu thành số

Công thức xác định hàm: sto + “ký tự viết tắt của tên kiểu:

VD: int -> i

Double -> d

Unsigned long -> ul

VD:

```
#include <iostream>
```



```
#include <string>

using namespace std;

int main(){
    string s1 = "87546";
    string s2 = "-125";
    string s3 = "12365double2T";
    int n1 = stoi(s1);
    int n2 = stoi(s2);
    int n3 = stoi(s3);
    cout << n1 << endl;
    cout << n2 << endl;
    cout << n3 << endl;
    return 0;
}
```

- Nếu chuyển đổi mà kết quả chuyển đổi bị tràn: vượt quá giới hạn dữ liệu -> trả về giá trị sai
- Nếu xâu có thêm chữ cái hoặc ký tự đặc biệt không phải số -> sẽ chuyển đổi từ đầu đến khi gặp ký tự không thuộc dạng đó

9. Chuyển đổi số thành chuỗi: sử dụng hàm `to_string(<số cần chuyển thành chuỗi>)`

VD:

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

int main(){
    int n = 12567;
    string s = to_string(n);
    cout << s << endl;
    return 0;
}
```

Question: Nhập 1 số nguyên n và 1 chữ số nguyên x từ bàn phím.
Hãy xuất ra màn hình số tự nhiên lớn nhất chỉ chứa chữ số x và nhỏ hơn n.

VD:

123

3 -> xuất ra màn hình 33

8974

7 -> xuất ra màn hình 7777

10. Chuyển đổi hoa thường: sử dụng tolower() và toupper()

- Lưu ý: Chỉ sử dụng với 1 ký tự duy nhất. Muốn chuẩn hóa cả xâu thì phải duyệt từng ký tự

VD:

```
#include <iostream>
#include <string>
#include <ctype.h>

using namespace std;

void to_lower(string &s){
    for(int i = 0; i < s.size(); i++){
        s[i] = tolower(s[i]);
    }
}

void to_upper(string &s){
    for(int i = 0; i < s.size(); i++){
        s[i] = toupper(s[i]);
    }
}
```

```

}

int main(){
    string s = "adobe photoshop";
    to_lower(s);
    cout << s << endl;
    to_upper(s);
    cout << s << endl;
}

```

11. Stringstream

stringstream nằm trong thư viện **sstream**, nó có chức năng biến chuỗi string thành luồng (tương tự như luồng vào từ bàn phím là cin). Từ đó bạn có thể đọc từng từ trong luồng stringstream ra và xử lý.

Ứng dụng: bài toán tách từ trong chuỗi

```

#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "python  java  C++ olympic";

```

```

//Khai báo stringstream ss và gán cho nó nội dung của s
stringstream ss(s);
string word;
int dem = 0;
while(ss >> word){
    ++dem;
    cout << "Tu thu " << dem << " tach duoc : " <<
word << endl;
}
return 0;
}

```

- Trường hợp muốn phân hoạch theo 1 ký tự khác: ta sử dụng getline()

VD:

```

#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "Dai-hoc-cong-nghiep-ha-noi";
    stringstream ss(s);
    string word;
}

```

```

int dem = 0;
while(getline(ss, word, '-')){
    ++dem;
    cout << "Tu thu " << dem << " tach duoc : " <<
word << endl;
}
return 0;
}

```

Trong trường hợp muốn phân hoạch cùng 1 lúc theo nhiều ký tự đặc biệt khác nhau (@ + - #), ta nên chuyển hết tất cả về 1 ký tự chung, rồi phân hoạch

Question:

1. Chuẩn hóa xâu ở bên trái mũi tên sao cho khi xuất ra màn hình ta sẽ nhận được xâu bên phải mũi tên

nGuyen mAnh HUNg -> Nguyen Manh Hung

2. Đếm số lượng từ trong câu bằng stringstream, xuất kết quả ra màn hình

VD: Hoc Lap Trinh C -> 4

Bài tập tổng hợp

Câu 1: Nhập vào bàn phím 1 chuỗi ký tự s , xác định số ký tự in hoa trong chuỗi đó

Câu 2: Nhập vào bàn phím 1 chuỗi ký tự s và 1 số nguyên dương n . Xóa tất cả các ký tự có chỉ số bằng bội của n trong chuỗi

Câu 3: Nhập 2 chuỗi ký tự a và b . Xác định xem chuỗi này có phải con của chuỗi kia không ? (Tính cả in hoa in thường)

Câu 4: Viết chương đếm số lượng kí tự in thường trong chuỗi cho trước, chuỗi nhập vào có độ dài không vượt quá 1000 và có thể bao gồm dấu cách.

Câu 5: Viết chương trình chuyển toàn bộ kí tự trong chuỗi cho trước về dạng in thường.

Câu 6: Viết chương trình chuyển toàn bộ kí tự trong chuỗi cho trước về dạng in hoa.

Câu 7: Viết chương trình kiểm tra xem kí tự nhập vào có phải là chữ cái. Nếu đúng trả về 1, ngược lại trả về 0.

Câu 8: Viết chương trình kiểm tra xem kí tự nhập vào có phải chữ cái hoặc chữ số. Nếu đúng trả về 1, ngược lại trả về 0.

Câu 9: Nhập 1 chuỗi ký tự s xác định chuỗi đó có phải là chuỗi đối xứng không (Cách xác định chuỗi đối xứng giống như cách xác định mảng đối xứng)

Câu 10: Nhập từ bàn phím 2 xâu ký tự a và b. Xác định xem 2 chuỗi này có phải là 2 chuỗi anagram không ? Biết chuỗi anagram là từ các ký tự của chuỗi này có thể sắp xếp lại thành chuỗi kia.

VD:

abcxyz

zybcax -> là 2 chuỗi anagram

kha banh

bankh ah -> là 2 chuỗi anagram

(Thực hành theo cả 2 cách, sử dụng append và sử dụng kết hợp compare và substr)

Câu 11: Chuẩn hóa xâu bên trái sao cho khi xuất ra màn hình, ta được xâu bên phải

a) ngUyen manH huNg -> Nguyen Manh Hung

b) ho@@ngOc@Ha -> Ho Ngoc Ha

c) ThiEn@@-DUOnG bao--@#Son -> Thien Duong Bao
Son

Câu 12: Nhập vào từ bàn phím 1 chuỗi ký tự s (có dấu cách). In chuỗi theo thứ tự ngược lại của các từ

VD: bo an co -> co an bo

thien duong bao son -> son bao duong thien