

TIN ĐẠY CƯỜNG

BÀI 8: LÀM VIỆC VỚI STRING

Nội dung buổi trước

- Bổ sung những khái niệm liên quan đến biến và kiểu dữ liệu: hằng số, tham chiếu, các kiểu dữ liệu cơ sở, phạm vi sử dụng biến, vòng đời của biến,...
- Các khai báo struct để tạo kiểu dữ liệu mới
- Kiểu dữ liệu chuỗi kí tự (string):
 - Khái niệm, cách khai báo và khởi tạo
 - Sử dụng chỉ mục với string
 - Một số hàm thành phần của string
 - Các loại bài tập xử lý string thông dụng

Nội dung

1. Ôn tập về cách định nghĩa kiểu dữ liệu mới với struct
2. Chi tiết về cách làm việc của các hàm thành phần của kiểu string
3. Chữa bài tập kiểu string buổi trước
4. Các bài tập bổ sung

Phần 1

Ôn tập về cách định nghĩa kiểu dữ liệu mới với struct

Khai báo và sử dụng struct

- Cú pháp sử dụng struct:

```
struct <tên kiểu> {  
    <các dữ liệu thành phần>  
};
```

- Ví dụ:

```
struct ThoiGian { // kiểu dữ liệu ThoiGian  
    int ngay, thang, nam; // các thành phần con  
};
```

```
ThoiGian homnay; // biến kiểu ThoiGian  
homnay.ngay = 21; // thành phần ngày = 21  
homnay.thang = 10; // thành phần tháng = 10  
homnay.nam = 2016; // thành phần năm = 2016
```

Hãy tự tạo vài kiểu dữ liệu mới

1. Kiểu dữ liệu **Point** mô tả một điểm trên mặt phẳng tọa độ (gồm tọa độ trục X và trục Y)
2. Kiểu dữ liệu **Line** mô tả một đoạn thẳng trên mặt phẳng tọa độ (gồm 2 điểm đầu cuối)
3. Kiểu dữ liệu **GiaoVien**, lưu trữ thông tin về các giáo viên trong trường, gồm có: họ tên, địa chỉ cư trú, số điện thoại, năm vào trường
4. Kiểu dữ liệu **SinhVien**, lưu trữ thông tin về các sinh viên trong trường, gồm: họ tên, khóa học, lớp quản lý, địa chỉ, điện thoại

Phần 2

Chi tiết về cách làm việc của các hàm thành phần của kiểu string

Hàm find

Hàm “string `find(<s>, <p>)`”: tìm vị trí xuất hiện đầu tiên của chuỗi `<s>` trong chuỗi hiện tại, bắt đầu tìm từ vị trí `<p>`

- Nếu không tìm thấy thì trả về -1
- Nếu không có tham số `<p>` thì tìm từ đầu chuỗi

```
string str = "This is text";
```

```
// tìm chuỗi "is" xuất hiện ở vị trí nào, tìm từ đầu
```

```
int x = str.find("is");
```

```
// tìm chuỗi "is" xuất hiện ở vị trí nào, tìm từ ô 10
```

```
int x = str.find("is", 4);
```


Hàm substr

Hàm “string `substr`(`<đầu>`, `<độ dài>`)” : tạo chuỗi mới là chuỗi con của chuỗi hiện tại

- Lấy từ vị trí `<đầu>`
- Lấy `<độ dài>` kí tự
- Nếu không có tham số `<độ dài>` thì lấy đến hết chuỗi

```
string str = "Đại học Thuy Loi";  
string str2 = str.substr(4, 3); // "hoc"  
string str3 = str.substr(9); // "Thuy Loi"
```

Hàm replace

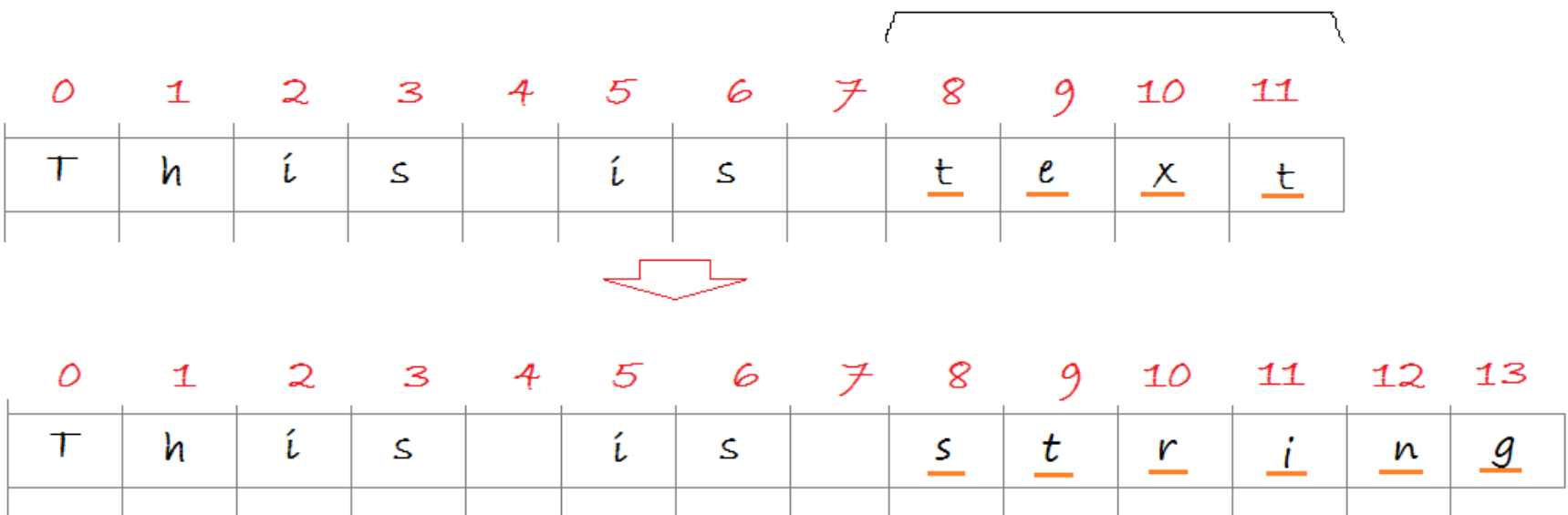
Hàm “string `replace`(<v>, <k>, <s>)” : thay đoạn con từ vị trí <v>, độ dài <k> bằng chuỗi <s>

```
string str = "This is text";  
// thay thế đoạn 4 kí tự, từ vị trí số 8 (“text”)  
// bằng chuỗi “string”, các nội dung khác giữ nguyên  
string s = str.replace(8, 4, "string");  
// in ra “This is string”  
cout << s;
```

Hàm replace

```
string str = "This is text";
```

```
str.replace(8, 4, "string");
```



Một số hàm khác (nhắc lại)

- Hàm `length()` hoặc `size()`: trả về chiều dài của string (số kí tự có trong string)
- Hàm `append(x)`: thêm x vào cuối string hiện tại (có thể dùng phép toán `+=` thuận tiện hơn)
- Hàm `push_back(c)`: thêm kí tự c vào cuối string (chuyên dùng cho kiểu kí tự, chạy nhanh hơn hàm `append`)
- Hàm `insert(v, str)`: chèn chuỗi str vào vị trí v trong string hiện tại
- Hàm `erase(v, k)`: xóa k kí tự bắt đầu từ vị trí v

Phần 3

Chữa bài tập kiểu string buổi trước

Bài tập về kiểu string

- 1. Nhập vào chuỗi S, in ra màn hình chuỗi vừa nhập và thông tin về chuỗi đó.*
- 2. Nhập vào chuỗi S, kiểm tra xem S có chứa toàn các chữ số hay không?*
- 3. Nhập và đếm số từ trong chuỗi S (một từ là dãy các kí tự liên tiếp không chứa dấu cách).*
- 4. Xóa mọi kí tự A trong chuỗi W nhập từ bàn phím*
- 5. Đếm xem chuỗi W nhập từ bàn phím chứa bao nhiêu dấu mở hoặc đóng ngoặc.*

Phần 4

Các bài tập bổ sung

Các bài tập bổ sung (1)

1. Nhập chuỗi ký tự W và ký tự C , hãy xóa tất cả những ký tự C khỏi chuỗi W sau đó in ra W .
2. Nhập chuỗi ký tự W và ký tự C . Hãy chèn ký tự C vào giữa tất cả những ký tự trong W (chuỗi “ABCD” sau khi chèn ‘M’ sẽ được “AMBMCMD”). In ra W .
3. Nhập chuỗi ký tự W và số nguyên n . Nếu độ dài của W nhỏ hơn n , hãy thêm vào đầu chuỗi W n ký tự ‘A’, nếu độ dài của W lớn hơn hoặc bằng n , hãy xóa đi $n/2$ ký tự đầu và $n/2$ ký tự cuối của W . In W ra màn hình sau khi xử lý.

Các bài tập bổ sung (2)

4. Nhập xâu kí tự W , hãy xóa đi tất cả những kí tự là chữ số trong W (xóa đi những kí tự '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' khỏi W). Sau đó in ra W .
5. Nhập xâu kí tự W và xâu S , nhập số nguyên p và q . Hãy chèn xâu S vào W ở vị trí p và q , nếu p hoặc q lớn hơn độ dài của W thì không thực hiện việc chèn. In ra xâu kết quả.
6. Nhập xâu kí tự W , xâu S và xâu C . Tìm xem xâu S có xuất hiện trong W hay không? Nếu có thì hãy chèn xâu C vào W ở vị trí xuất hiện S đầu tiên.

Các bài tập bổ sung (3)

7. Nhập xâu kí tự W và xâu S . Kiểm tra xem xâu S có xuất hiện trong W hay không? Nếu không thì in ra thông báo “ S không xuất hiện trong W ”, nếu có thì hãy chỉ ra vị trí xuất hiện cuối cùng của S trong W .
8. Nhập xâu kí tự W , gọi độ dài của W là n . Tạo ra xâu S bằng cách ghép liên tiếp 3 xâu W . Hãy chia S thành n chuỗi con độ dài 3 kí tự. Hãy in ra các xâu con đó
 - Ví dụ: $W = \text{“ABCD”}$ thì $S = \text{“ABCDABCDABCD”}$ và 4 chuỗi con sẽ là: “ABC”, “DAB”, “CDA”, “BCD”