



Caracteres e Strings

Neilor Tonin

Introdução

O mais importante: diferenciar os tipos de variáveis:

- Char;
- Vetor de char;
- String;
- Vetor de string e
- Utilização de string e char em funções (reforço)

Introdução

- Strings
 - Série de caracteres tratada como uma única unidade
 - Podem incluir letras, dígitos e caracteres especiais (*, /, \$)
 - Literal string (constante string) – escrita entre aspas
 - "Hello"
 - Strings são arrays de caracteres
 - String é um apontador para o primeiro caractere
 - Valor de um string é o endereço do primeiro caractere

Char

Variáveis tipo caracteres:

- São utilizadas quando se deseja ler, mostrar, atribuir ou utilizar apenas um caractere.
 - Exemplos:

```
char x;  
x = 's';  
cout << x+1 << endl;
```

t

Char

Variáveis tipo caracteres:

- Segundo exemplo:

```
char x,y;  
cin >> x;  //s  
cin >> y;  //a  
int z = x-y;  
cout << z << endl;
```

```
s  
a  
18
```

Char

Variáveis tipo caracteres:

- Terceiro exemplo:

```
char x,y;  
for (char i='a'; i <= 'm'; i++){  
    cout << i << ' ';  
}
```

```
a b c d e f g h i j k l m
```

Char

Variáveis tipo caracteres:

- Quarto exemplo (comparação com char):

```
char x;  
cin >> x;  
if (x<='Z' && x >= 'A') {  
    cout << "Maiusculo" << endl;  
} else {  
    cout << "Minusculo" << endl;  
}
```

s
Minusculo

Exercício 1: Faça uma função que receba uma string com espaços e converta toda ela para maiúscula.

Vetor de Char

- Declarações de vetor de caracteres
 - Declare como um array de caracteres ou uma variável de tipo **char ***

```
char cor[] = "azul";  
char *corPtr = "azul";  
char linha[6] = { 'm', 'a', 'n', 'g', 'o' };
```
 - Lembre-se que strings representados como arrays de caracteres terminam com '**\0**'
 - **cor** tem 5 elementos
 - Pode-se ler utilizando usando gets
 - `#include <cstdio>`
 - ...
 - `gets(linha);` `/// getline só pode ser usado com string`

Vetor de Char

- Declarações de vetor de caracteres
 - Declare como um array de caracteres ou uma variável de tipo **char ***

```
char cor[] = "azul";  
char *corPtr = "azul";  
char linha[6] = { 'm', 'a', 'n', 'g', 'o' };
```
 - Lembre-se que strings representados como arrays de caracteres terminam com '`\0`'
 - **cor** tem 5 elementos

Biblioteca de Manipulação de Caracteres (ctype)

| Protótipo | Descrição |
|------------------------------------|---|
| <code>int isdigit(int c)</code> | Retorna verdadeiro se c é um dígito e falso caso contrário. |
| <code>int isalpha(int c)</code> | Retorna verdadeiro se c é um letra e falso caso contrário. |
| <code>int isalnum(int c)</code> | Retorna verdadeiro se c é um dígito or uma letra e falso caso contrário. |
| <code>int isxdigit(int c)</code> | Retorna verdadeiro se c é um caractere dígito hexadecimal e falso caso contrário. |
| <code>int islower(int c)</code> | Retorna verdadeiro se c é uma letra minúscula e falso caso contrário. |
| <code>int isupper(int c)</code> | Retorna verdadeiro se c é uma letra maiúscula; falso caso contrário. |
| <code>int tolower(int c)</code> | Se c é auma letra maiúscula, tolower retorna c como uma letra minúscula. Caso contrário, tolower retorna o argumento sem modificação. |
| <code>int toupper(int c)</code> | Se c é uma letra minúscula, toupper retorna c como uma letra maiúscula. Caso contrário, toupper retorna o argumento sem modificação. |

- Exercício 2: Cada linha de estudantes constrói um programa para executar uma destas funções.

Vetor de Char

Variáveis tipo **VETOR** de caracteres:

- É necessário quando queremos armazenar várias letras para posterior manipulação ou acesso.
- É manipulada como se fosse um vetor de outro tipo qualquer. Exemplo:

```
char x[26];
```

```
for (int i=0; i<26; i++){  
    x[i] = 'a' + i;  
}
```

```
for (int i=10; i>=0; i-- ){  
    cout << x[i] << " ";  
}
```

```
k j i h g f e d c b a
```

String

Variáveis tipo **string** são um pouco diferentes:

- Embora possamos utilizá-las como um vetor de caracteres, é um tipo específico do C++ (e não do C)
- A classe string possui várias funções membro (métodos) que podem ser utilizadas, tais como size, substr, c_str, clear, compare, etc;
- Deve-se utilizar o `#include <string>`
- A definição é feita da seguinte forma:
 - `string x;`
 - `String y = "isto eh um texto";`

String

Primeiro exemplo de string:

- Atribuir de uma string para uma char, apresentando na tela utilizando o método size() :

```
string x= "um teste";  
char letra;  
for (int i = x.size()-1; i >= 0; i--) {  
    letra = x[i];  
    cout << letra << endl;  
}
```

e
t
s
e
t

m
u

String

Segundo exemplo de string:

- Alguns outros problemas são resolvidos com utilização do método size. Saber as posições válidas de uma string para poder percorrê-la de forma correta é muito importante:

```
string x= "teste";  
string y= "melancia";  
int tam1= x.size(), tam2= y.size();  
if (tam1 > tam2 ) {  
    cout << x << " tem mais letras" << endl;  
} else {  
    cout << y << " tem mais letras" << endl;  
}
```

```
Melancia tem mais letras
```


String

Terceiro exemplo de string:

- Muito importante: se o método `size()` retorna 5, as posições válidas são de 0 até 4 [`size()`-1]

```
string x = "Menta";  
int tam = x.size();  
for (int i = 0; i < tam; i++ ) {  
    cout << i << ": " << x[i] << endl;  
}
```

```
0: M  
1: e  
2: n  
3: t  
4: a
```

String

Quarto exemplo de string:

- Utilizando o exemplo anterior, posso deslocar uma letra para esquerda ou direita em cada posição:

```
string x = "Menta";  
int tam = x.size();  
for (int i = 0; i < tam; i++) {  
    cout << i << ": " << (char) (x[i] +1) << endl;  
}
```

```
0: N  
1: f  
2: o  
3: u  
4: b
```

- É necessário a conversão de cast (char) porque se não for utilizado, o compilador mostra o valor ascii dos caracteres (valor numérico) pois tinha uma soma (+)

String

Quinto exemplo de string:

- Utilizando o método substr() pode-se pegar uma parte qualquer de uma string.

```
           012345678  
string x = "Mangojata";  
string y = x.substr(4,3);  
cout << y << endl;  
  
Cout << endl;  
  
cout << 'S' + y << endl;
```

oja

Soja

- Tanto faz somar “S” ou ‘S’ à string, funciona de qualquer forma.

String

Sexto exemplo de string:

- Método compare, muito útil para comparar lexicograficamente 2 strings

```
string nome1 = "Francisco";  
string nome2 = "Franciele";  
string nome3 = "Heitor";
```

```
cout << nome1.compare(nome2) << endl;  
cout << nome1.compare(nome3) << endl;  
cout << nome3.compare("Heitor Pedo") << endl;  
cout << nome1.compare("Francisco") << endl;
```

```
1  
-1  
-5  
0
```

String

Sétimo exemplo de string:

- Métodos find (encontrar) e erase (apagar/remove).

```
string texto = "AGDFUZZT";  
              // 0123456  
int pos = texto.find('F');  
int pos2 = texto.find('M');  
cout << pos << " " << pos2 << endl;  
  
texto.erase(pos, 1);  
cout << texto << endl;
```

```
3 -1  
AGDUZZT
```


String

Oitavo exemplo de string:

- Quando você precisar utilizar uma função da biblioteca sobre um vetor de char mas sua variável é uma string do C++.
- A função `atoi()`, por exemplo, transforma um vetor de char para um valor inteiro:

```
string valor = "12375";  
int x = atoi( valor.c_str() );  
cout << x/100.0 << endl;
```

123.75

- Obs: a função `atoi` está na biblioteca **`cstdlib`**:
`#include <cstdlib>`

String

Nono exemplo de string:


- Outro exemplo de uso do método `c_str()`:

```
        64  
       128 16  
        ↓ ↓ ↓  
string valor = "10010101";  
              32 8 4 2 1  
cout << strtol (valor.c_str(), NULL, 2);  
  
128 + 16 + 4 + 1 = 128 + 21 = 149
```

149

- **Obs:** a função `strtol` é do `cstdlib` e espera como primeiro parâmetro um vetor de caracteres.

Plano de Dieta

Por TopCoder*  EUA

Tempo Limite: 1

O doutor deu a você a sua dieta, na qual cada caractere corresponde a algum alimento que você deveria comer. Você também sabe o que você tem comido no café da manhã e no almoço, nos quais cada caractere corresponde a um tipo de alimento que você deveria ter comido aquele dia. Você decidiu que irá comer todo o restante de sua dieta durante o jantar, e você quer imprimi-la como uma String (ordenada em ordem alfabética). Se você trapaceou de algum modo (ou por comer muito de tipo de alimento, ou por comer algum alimento que não está no plano de dieta), você deveria imprimir a cadeia "CHEATER" (significa trapaceiro), sem as aspas.

Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que indica a quantidade de casos de teste. Cada caso de teste é composto por três linhas, cada uma delas com uma string com até 26 caracteres de 'A'-'Z' ou vazia, representando respectivamente os alimentos da dieta, do café da manhã e do almoço.

Saída

Para cada caso de teste imprima uma string que representa os alimentos você deve consumir no jantar, ou "CHEATER" caso você tenha trapaceado na sua dieta.

Exemplo de Entrada

4

ABCD

AB

C

ABEDCS

EDSMB

MSD

A

Exemplo de Saída

D

ABCDES

CHEATER

DEIN

Vetor de String

Vetor de string:

- Supondo que você queira guardar o nome de 5 pessoas em um vetor de strings:

```
string pessoas[3];  
string A = "Neilor Tonin";  
pessoas[0] = A;  
pessoas[1] = "Maria Araper";  
pessoas[2] = "Mangojata Mancuda";  
pessoas[1] += " Seguida";  
  
for (int i=0; i<3; i++) {  
    cout << pessoas[i] << endl;  
}
```

```
Neilor Tonin  
Maria Araper Seguida  
Mangojata Mancuda
```


Utilização de char e string em funções

Utilização de char em funções: faça uma função que recebe dois caracteres e mostra do menor até o maior:

```
void mostra_letras(char a, char b) {  
    if (a>b) {  
        swap(a,b);  
    }  
    for (char i=a; i<=b; i++) {  
        cout << i << endl;  
    }  
}  
  
int main () {  
    char a='m', b= 'd';  
    mostra_letras(a,b);  
    return 0;  
}
```

```
d  
e  
f  
g  
h  
i  
j  
k  
l  
m
```

Utilização de char e string em funções

Utilização de string em funções: faça uma função que recebe duas string e diz qual é a maior (em tamanho) no programa principal:

```
string maior(string a, string b) {  
    if (a.size() > b.size()) {  
        return a;  
    }  
    return b;  
}
```

```
int main () {  
    string a="Meu", b="cachorro";  
    cout << maior(a,b);  
    return 0;  
}
```

cachorro

Utilização de vetor de char em funções

Em uma função, receba um vetor letras[10] com as primeiras 10 letras do alfabeto. Desloque cada letra 2 posições para a direita e mostre o resultado no programa principal:

```
void transforma(char letras[]) {  
    for (int i=0; i<7; i++) {  
        letras[i] = letras[i] + 2;  
    }  
}  
  
int main () {  
    char letras[7]={'a','b','c','d','e','f','g'};  
    transforma(letras);  
    for (int i=0; i<7; i++) {  
        cout << letras[i] << endl;  
    }  
    return 0;  
}
```

c
d
e
f
g
h
i