ESCOLA DE PRIMAVERA DA MARATONA SBC DE PROGRAMAÇÃO

PROMOÇÃO:

CEFET-MG

UFMG

UFV

UNIFEI

USP Universidade de São Paulo

UFU

APOIO:

beecrowd

SBC Sociedade Brasileira de Computação

Grupo de Computação Competitiva

STRING

Por: *Henrique Campos Junqueira*

CONTEÚDOS

01 - PROBLEMA MOTIVADOR

beecrowd | 2242

Huaauhahhuahau

Por Maratona de Programação da SBC 2016 Brazil

Timelimit: 1

Em chats, é muito comum entre jovens e adolescentes utilizar sequências de letras, que parecem muitas vezes aleatórias, para representar risadas. Alguns exemplos comuns são:

huaauhahhuahau
hehehehe
ahahahaha
jaisjjkasjksjjskjakijs
huehuehue

Cláudia é uma jovem programadora que ficou intrigada pela sonoridade das "risadas digitais". Algumas delas ela nem mesmo consegue pronunciar! Mas ela percebeu que algumas delas parecem transmitir melhor o sentimento da risada que outras. A primeira coisa que ela percebeu é que as consoantes não interferem no quanto as risadas digitais influenciam na transmissão do sentimento. A segunda coisa que ela percebeu é que as risadas digitais mais engraçadas são aquelas em que as sequências de vogais são iguais quando lidas na ordem natural (da esquerda para a direita) ou na ordem inversa (da direita para a esquerda), ignorando as consoantes. Por exemplo, "hahaha" e "huaauhahhuahau" estão entre as risadas mais engraçadas, enquanto "riajkjdhhihhjak" e "huehuehue" não estão entre as mais engraçadas.

Cláudia está muito atarefada com a análise estatística das risadas digitais e pediu sua ajuda para escrever um programa que determine, para uma risada digital, se ela é das mais engraçadas ou não.

01 - PROBLEMA MOTIVADOR

Entrada

A entrada é composta por uma linha, contendo uma sequência de no máximo 50 caracteres, formada apenas por letras minúsculas sem acentuação. As vogais são as letras 'a','e','i','o','u'. A sequência contém pelo menos uma vogal.

Saída

Seu programa deve produzir uma linha contendo um caractere, "S" caso a risada seja das mais engraçadas, ou "N" caso contrário.

Exemplos de Entrada	Exemplos de Saída
hahaha	S
riajkjdhhihhjak	N
a	S
huaauhahhuahau	S

Maratona de Programação da SBC 2016

02 - DEFINIÇÃO DA ESTRUTURA

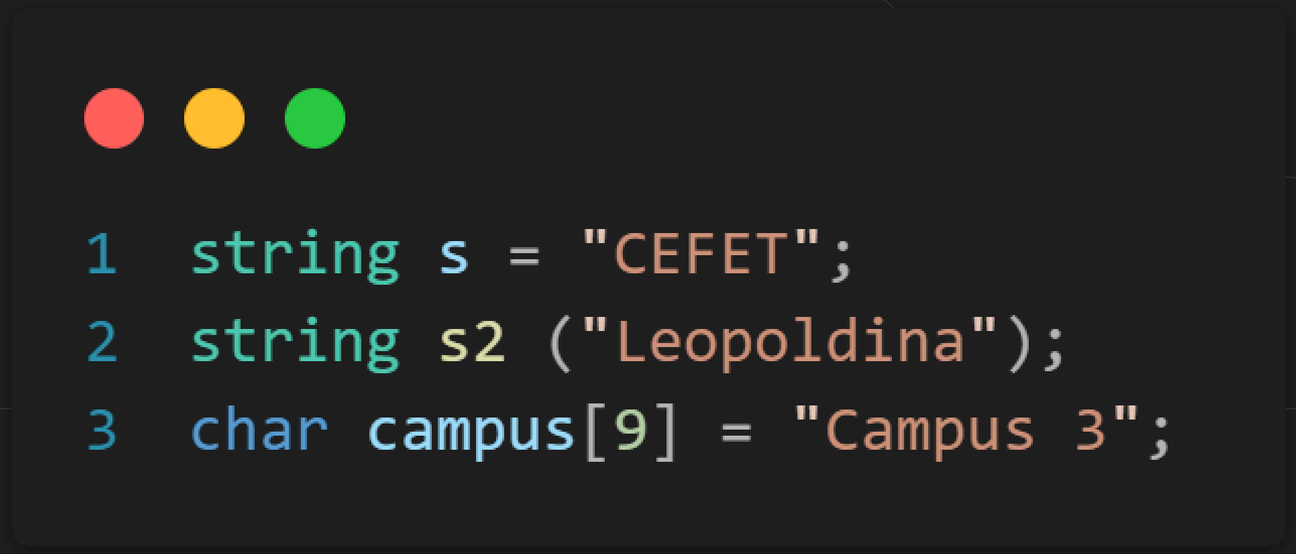
- O que é string?
  - Um dos tipos mais utilizados em programação
  - Sequência ou cadeia de caracteres
- Principal utilização
  - Manipulação de textos
- String vs. vetor de char
  - Mais fácil de usar
  - Simplifica a atribuição de textos

```
1  string s = "CEFET";
2  cout << s << endl;
```

C	E	F	E	T	\0
0	1	2	3	4	

03 - FUNCIONAMENTO DA ESTRUTURA

- Diferentes declarações
- Principais funcionalidades:
  - size(), length() e at()
  - clear() e empty()
  - front(), back()
  - append()
  - replace() e swap()
  - getline()



```
1  string s = "CEFET";
2  string s2 ("Leopoldina");
3  char campus[9] = "Campus 3";
```

04 - ESTRUTURA DE DADOS

- size(), length() e at()

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

Saída: C E F E T

```cpp
string s = "CEFET";
for (int i = 0; i < s.length(); i++) {
    cout << s.at(i) << ' ';
}
cout << endl;
```

Saída: C E F E T

04 - ESTRUTURA DE DADOS

- clear() e empty()

```cpp
string s = "CEFET";
cout << s << endl;
s.clear();
if (s.empty()) cout << "Vazia";
```

Saída: Vazia

04 - ESTRUTURA DE DADOS

- front() e back()

```
1  string s = "CEFET";
2  cout << s.front() << ' ';
3  cout << s.back() << endl;
```

Saída: C T

04 - ESTRUTURA DE DADOS

- append()

```cpp
string str;
string s = "CEFET";
string s2 = "Leopoldina";
str.append(s);
str.append(" ");
str.append(s2);
cout << str << endl;
```

```cpp
string s = "CEFET";
string s2 = "Leopoldina";
cout << s + " " + s2 << endl;
```

Saída: CEFET Leopoldina

04 - ESTRUTURA DE DADOS

- replace() e swap()

```
1  string s = "CEFET";
2  s.replace(0, 1, "c");
3  cout << s << endl;
```

Saída: cEFET

```
1  string s = "CEFET";
2  string s2 = "Leopoldina";
3
4  cout << "Antes do swap: " << s << ' ' << s2 << endl;
5  s.swap(s2);
6  cout << "Depois do swap: " << s << ' ' << s2 << endl;
```

Saída: Antes do swap: CEFET Leopoldina
Depois do swap: Leopoldina CEFET

04 - ESTRUTURA DE DADOS

- getline()

```
1  string s;
2  cin >> s;
3  cout << s << endl;
```

```
1  string s;
2  getline(cin, s);
3  cout << s << endl;
```

05 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

- Qual a ideia inicial?
- Entrada e saída
- Esboço

H	A	H	A	H	A
0	1	2	3	4	5

- Somente vogais: A A A
- Vogais ao contrário: A A A
- São iguais? (A A A = A A A)

05 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

```cpp
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string s, resp, contr;
    cin >> s;
```

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

    for (int i = resp.size() - 1; i >= 0; i--) {
        contr += resp[i];
    }

    if (resp == contr)
        cout << "S" << endl;
    else
        cout << "N" << endl;

    return 0;
}
```

05 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

```cpp
bool checkVowel(char c) {
    string vowels = "aeiou";

    for (int i = 0; i < vowels.size(); i++) {
        if (c == vowels[i])
            return true;
    }

    return false;
}
```

05 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

```cpp
#include <bits/stdc++.h>

using namespace std;

int main()
{
    vector<char> vogais = {'a', 'e', 'i', 'o', 'u'};
    vector<char> resp = {};
    vector<char> contr = {};

    string s; getline(cin, s);
```

05 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

```cpp
    for (int i = 0; i < s.size(); i++) {
        if (find(vogais.begin(), vogais.end(), s[i]) != vogais.end()) {
            resp.push_back(s[i]);
        }
    }

    for (int i = resp.size() - 1; i >= 0; i--)
        contr.push_back(resp[i]);

    cout << (resp == contr ? 'S' : 'N') << endl;

    return 0;
}
```

06 - OUTRAS APLICAÇÕES

- Existem diversos tipos de aplicação
  - Substring

C	E	F	E	T
C	E	F		
0	1	2	3	4

```
1  string s = "CEFET";
2  cout << s.substr(0, 3);
```

06 - OUTRAS APLICAÇÕES

- Existem diversos tipos de aplicação
  - Contar consoantes
    - Percorrer a string, verificar se é consante
    - Adicionar 1 em um contador

C	E	F	E	T

- Quantidade: 3

06 - OUTRAS APLICAÇÕES

- Existem diversos tipos de aplicação
  - Contar consoantes

```cpp
#include <bits/stdc++.h>

using namespace std;

bool checkVowel(char c) {
    string vowels = "aeiou";

    for (int i = 0; i < vowels.size(); i++) {
        if (c == vowels[i])
            return true;
    }

    return false;
}
```

```cpp
int main()
{
    string s;
    int consonants = 0;

    cout << "Digite uma frase: ";
    getline(cin, s);

    for (char c : s) {
        c = tolower(c);
        if (isalpha(c) && !checkVowel(c)) {
            consonants++;
        }
    }

    cout << "Numero de consoantes: " << consonants << std::endl;

    return 0;
}
```

06 - OUTRAS APLICAÇÕES

```cpp
#include <bits/stdc++.h>

using namespace std;

bool checkVowel(char c) {
    string vowels = "aeiou";

    for (int i = 0; i < vowels.size(); i++) {
        if (c == vowels[i])
            return true;
    }

    return false;
}
```

```cpp
int main()
{
    string s;
    int consonants = 0;

    cout << "Digite uma frase: ";
    getline(cin, s);

    for (char c : s) {
        c = tolower(c);
        if (isalpha(c) && !checkVowel(c)) {
            consonants++;
        }
    }

    cout << "Numero de consoantes: " << consonants << std::endl;

    return 0;
}
```

07 - PROBLEMAS

2242 - Huaauhahhuahau
1234 - Sentença Dançante
1272 - Mensagem Oculta
1253 - Cifra de César
1024 - Criptografia

OBRIGADO PELA ATENÇÃO

Grupo de Computação Competitiva

GCC.CEFETMG

gcc-cefetmg.github.io/Site-GCC

@gcc.cefetmg