

STRINGS

Dilvan Moreira

(baseado no livro Big Java e Z. Liang)

Objetivos



- Entender o que são e como usar:
 - Strings

Lembrando: Controlando Coisas

Latas de Coca p/ mês

185

92

370

485

209

128

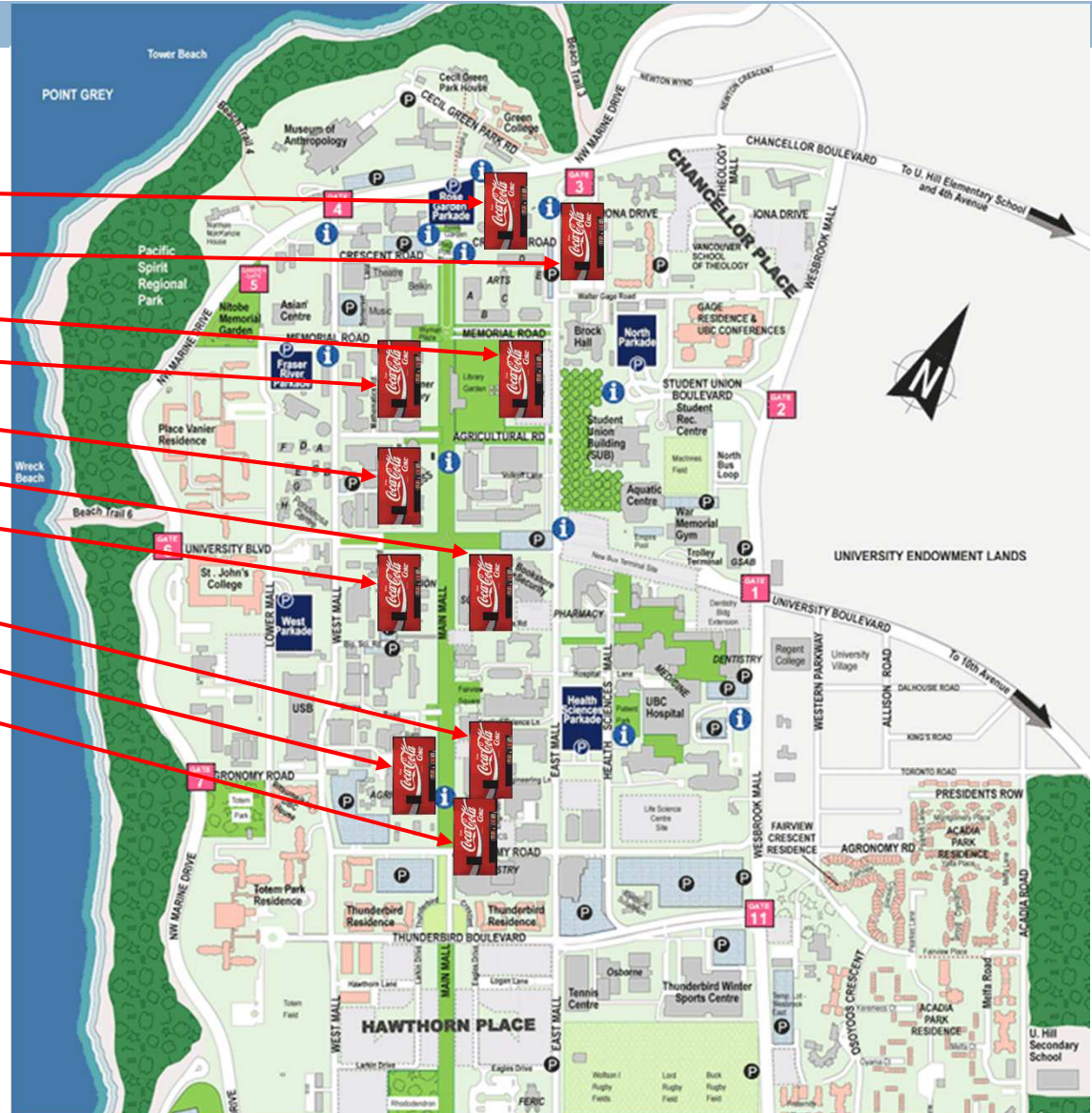
84

151

32

563

Em outras palavras:
Como posso organizar esses
dados para usá-lo no meu
computador?



Lembrando: Arrays

cansSold

185
92
370
485
209
128
84
151
32
563

- **arrays:** estrutura comum em linguagens de computador
 - ▣ agrupa dados relacionados conjuntamente
 - ▣ organização que permite que cada dado individual possa ser facilmente lido ou escrito
- coleção de variáveis
 - ▣ todas do mesmo tipo
 - ▣ compartilham um nome comum
- cada variável guarda um único valor

Lembrando: Declaração e tipo de Arrays

cansSold

0	185
1	92
2	370
3	485
4	209
5	128
6	84
7	151
8	32
9	563

- Como uma declaração comum de variável
 - ▣ declarado antes de ser usado
 - ▣ tem que ter um tipo
- Como `cansSold` contém inteiros, faça um array de inteiros:

```
int cansSold[10];
```
- Parece uma declaração de variável, exceto que:
 - ▣ Número indica o tamanho do array

Lembrando: Declaração e Tipos de Arrays

cansSold

0	185
1	92
2	370
3	485
4	209
5	128
6	84
7	151
8	32
9	563

```
int main() {
    int ARRAYSIZE = 10;
    int cansSold[ARRAYSIZE];

    cansSold[0] = 185;
    cansSold[1] = 92;
    cansSold[2] = 370;
    cansSold[3] = 485;
    cansSold[4] = 209;
    cansSold[5] = 128;
    cansSold[6] = 84;
    cansSold[7] = 151;
    cansSold[8] = 32;
    cansSold[9] = 563;

    // do useful stuff here
    printf("Element 4 is %d", cansSold[4]);
}
```

Lembrando: Usando Arrays e Loops

cansSold

0	185
1	92
2	370
3	485
4	209
5	128
6	84
7	151
8	32
9	563

□ Escreve um programa para

- criar um array
- achar número total de latas vendidas
- imprimir resultado

```
int main() {
    int totalCans = 0;
    int cansSold[] = {185, 92, 370, 485, 209,
                      128, 84, 151, 32, 563};

    for (int i = 0; i < cansSold.length; i++)
    {
        totalCans = totalCans + cansSold[i];
    }
    printf("We've sold %d cans of pop",
           totalCans);
}
```

Lembrando: Usando Diferentes Tipos de Dados

cansSold	cashIn
0 185	0 201.25
1 92	1 100.50
2 370	2 412.75
3 485	3 555.25
4 209	4 195.00
5 128	5 160.00
6 84	6 105.00
7 151	7 188.75
8 32	8 40.00
9 563	9 703.75

Mesmo tamanho -> Tipos diferentes

O que o programa imprime?

Calcule quanto cada máquina vendou e compare ao valor declarado para cada uma delas.

```
int main()
{
    double expected;
    int cansSold[] = {185, 92, 370, 485, 209,
                     128, 84, 151, 32, 563};
    double cashIn[] = {201.25, 100.50, 412.75,
                      555.25, 195.00, 160.00,
                      105.00, 188.75, 40.00,
                      703.75};

    for (int i = 0; i < cansSold.length; i++)
    {
        expected = cansSold[i] * 1.25;
        printf("Machine %d off by $ %f", (i+1),
              (expected -
               cashIn[i]));
    }
}
```


Lembrando: Array Bidimensional

	columns		
	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4
3	0	3	6

rows

```
int main() {
    int multTable[4][3];

    for (int row = 0; row < 4; row++){
        for (int col = 0; col < 3; col++) {
            multTable[row][col] = row * col;
        }
    }

    for (int row = 0; row < 4; row++){
        for (int col = 0; col < 3; col++){
            printf("%d ", multTable[row][col]);
        }
        printf("\n");
    }
}
```

Strings



- Não existe um tipo string em C
- Strings são vetores de **chars**.
- A declaração geral para uma string é:

```
char nome_da_string [tamanho];
```

- Strings têm o seu último elemento como um `'\0'`.
- O tamanho da string deve incluir o `'\0'` final.

Copiando Strings

□ Programa para copiar uma string

```
#include <stdio.h>
void main () {
    int count;
    char str1[100], str2[100];
    ... /* Le str1 que sera copiada para str2 */
    for (count = 0; str1[count]!='\0'; count++)
        str2[count] = str1[count];
    str2[count] = '\0';
    ....          /* Aqui o programa continua */
}
```

Manipulando Strings

- Como strings não são tipos primitivos em C → não podem ser manipulados diretamente
 - ▣ Não se pode igualar duas strings
`string1 = string2; //Nao funciona. Por que?`
- Para remediar essa situação a biblioteca padrão do C vem com várias funções para manipular strings

gets()

- Lê uma string do teclado
 - ▣ `gets(nome_da_string);`

```
#include <stdio.h>
void main () {
    char string[100];

    printf ("Digite o seu nome: ");
    gets (string);
    printf ("\n\n Ola %s", string);
}
```

strcpy()

□ Copia a string origem para a string destino

▣ `strcpy(string_destino, string_origem);`

```
#include <stdio.h>
```

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

```
void main () {
```

```
    char str1[100], str2[100], str3[100];
```

```
    printf ("Entre com uma string: ");
```

```
    gets (str1);
```

```
    strcpy (str2, str1); /* Copia str1 em str2 */
```

```
    strcpy (str3, "Voce digitou a string ");
```

```
        /* Copia "Voce digitou a string" em str3 */
```

```
    printf ("\n\n%s%s", str3, str2);
```

```
}
```

strcat()

- strcat - a função strcat() tem a seguinte forma geral

```
strcat (string_destino, string_origem);
```

A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

strcat()

```
#include <stdio.h>
#include <string.h>
void main ()
{
    char str1[100], str2[100];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2, "Voce digitou a string ");
    strcat (str2, str1);
    /* str2 armazenara' Voce digitou a string + o conteudo de str1 */
    printf ("\n\n%s", str2);
}
```


strlen()

- strlen - sua forma geral é:

```
strlen (string);
```

- A função `strlen()` retorna o comprimento da string fornecida. O terminador nulo não é contado. Isto quer dizer que, de fato, o comprimento do vetor da string deve ser um a mais que o inteiro retornado por `strlen()`.

strlen()

```
#include <stdio.h>
#include <string.h>
void main ()
{
    int size;
    char str[100];
    printf ("Entre com uma string: ");
    gets (str);
    size = strlen (str);
    printf ("\n\n A string que voce digitou tem tamanho %d", size);
}
```

strcmp()

- strcmp - sua forma geral é:

```
strcmp (string1, string2);
```

- A função strcmp() compara a string 1 com a string 2. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não-zero.

strcmp()

```
#include <stdio.h>
#include <string.h>
void main ()
{
    char str1[100], str2[100];
    printf ("Entre com uma string: ");
    gets (str1);
    printf ("\n\n Entre com outra string: ");
    gets (str2);
    if (strcmp(str1,str2))
        printf ("\n\n As duas strings são diferentes.");
    else
        printf ("\n\n As duas strings são iguais.");
}
```

Exercício

- Escreva um programa que lê uma string e uma substring (da primeira) e imprime a primeira posição da substring. Caso a substring não seja encontrada, imprime “Não encontrei”
 - ▣ Ex: “guarda-chuva” e “chuva” imprime “Primeira posicao eh 7”
 - ▣ Ex: “guarda-chchuva” e “chuva” imprime “Primeira posicao eh 9”

Perguntas ?

