

# Curso de Introdução à Programação de Computadores



Dia 5



# Tarefa 1 - Fila de entrada

- A professora vai ordenar os alunos em uma fila por ordem alfabética para entrar na sala de aula
- Escreva um programa que:
  - Leia uma linha contendo um inteiro  $N$ , o número de alunos e um inteiro  $M$ , uma posição
  - A seguir leia  $N$  linhas com um nome de aluno em cada linha. O nome não tem espaços em branco.
  - Determine qual o nome do aluno que vai ser o de número  $M$  a entrar na sala



# Strings

- Uma **String** (cadeia de caracteres) é uma **coleção** de valores de tipo **char** que podem ser acessados individualmente, com um mesmo nome de variável (como um vetor, mas há diferenças!)
- Por exemplo, uma string de nome fruta, com cinco elementos, é mostrada **esquemáticamente** abaixo

fruta      0      1      2      3      4  
          [m] [a] [n] [g] [a]

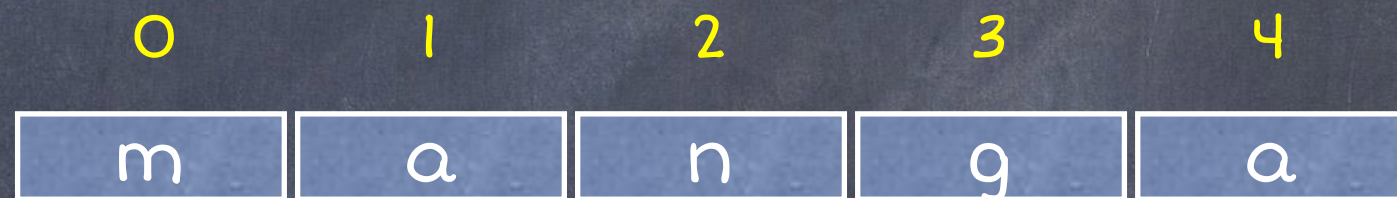


# String

nome

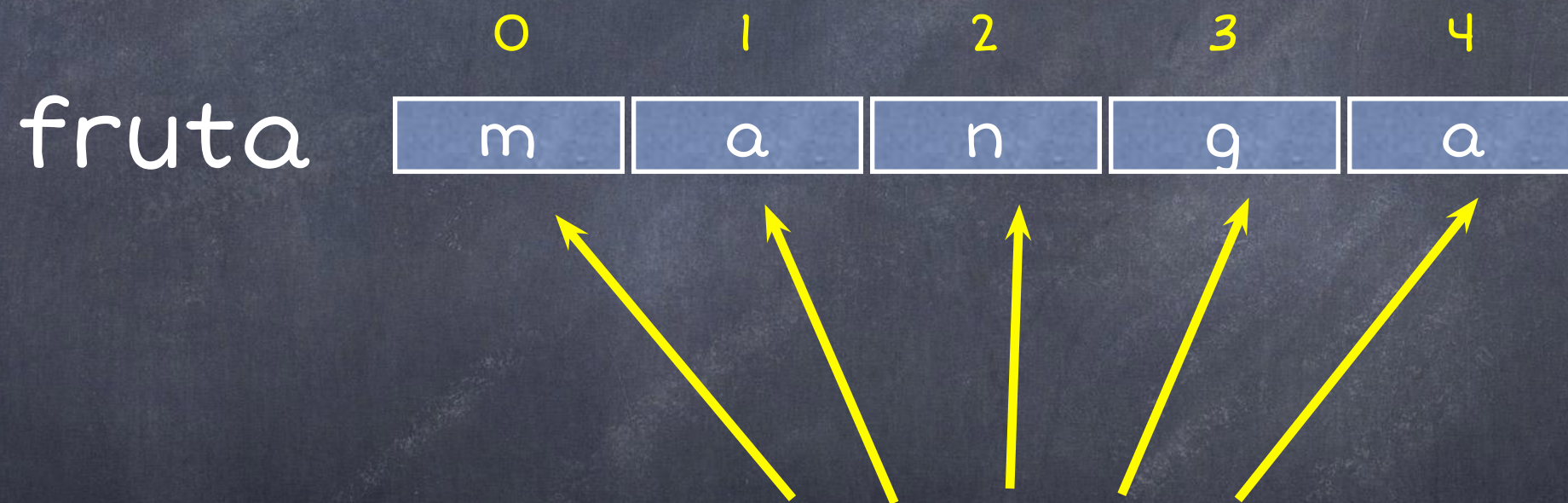


fruta





# String



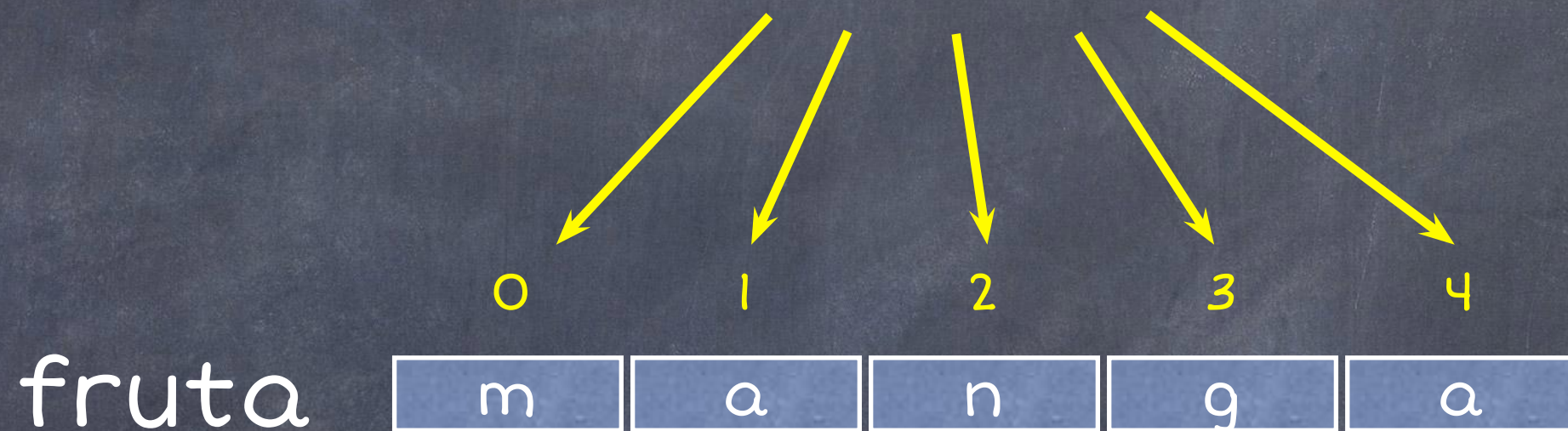
elementos da string, que  
contêm valores de tipo char



# String

índices

(o primeiro é sempre zero)





# Declaração de strings

- Como toda variável, antes de ser usada uma variável `string` deve ser declarada

```
string frutas;
```

- Para usar strings é necessário incluir mais uma linha no cabeçalho do programa:

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





# Declaração de strings

- Exemplos de declaração de strings:

```
string fruta;  
string nome = "Carlos Henrique";  
string temp = "a";
```



# Uso de strings

- Exemplos de usos de strings:

```
string pessoa;  
string nome = "Carlos Henrique";  
string sobrenome = "Santos";  
char caractere;  
  
pessoa = nome + " " + sobrenome;  
// pessoa recebe valor "Carlos Henrique Santos"  
  
caractere = nome[0];  
// caractere recebe valor 'C';
```



# Uso de strings

- Exemplos de usos de strings:

```
string pessoa;  
string nome = "Carlos Henrique";  
string sobrenome = "Santos";  
char caractere;  
  
printf("%s\n", sobrenome)  
// imprime Santos  
  
printf("%c\n", sobrenome[1])  
// imprime a
```



# Uso de strings

- Outro Exemplo de uso de strings:

```
string nome;
```

```
scanf("%s", &nome);
```

```
for (int i; i < nome.size(); i++) {
```

```
    printf("%c\n", nome[i]);
```

```
}
```



# Tarefa 1 - Fila de entrada

- A professora vai ordenar os alunos em uma fila por ordem alfabética para entrar na sala de aula
- Escreva um programa que:
  - Leia uma linha contendo um inteiro  $N$ , o número de alunos e um inteiro  $M$ , uma posição
  - A seguir leia  $N$  linhas com um nome de aluno em cada linha. O nome não tem espaços em branco.
  - Determine qual o nome do aluno que vai ser o de número  $M$  a entrar na sala



# Uso de strings

- E se os nomes contiverem espaços em branco (como "Carlos Henrique Silva")?

```
string nome;
```

```
getline(cin, nome);
```

```
// lê uma linha inteira e atribui à variável nome
```



# Tarefa 2 - Siga Mapa

- Escreva um programa que:
  - Leia uma linha contendo dois inteiros, M e N, menores ou iguais a 10.
  - A seguir leia M linhas com N inteiros cada.
  - Imprima o número de movimentos necessários para chegar até o fim do caminho.
  - Assuma que o caminho sempre começa no canto superior esquerdo do mapa.

1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	1	1	0
1	0	0	0	0	0	1	1
1	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1



# Matrices



# Matriz

- **Matriz** é um **coleção** de elementos de mesmo tipo, que podem ser acessados individualmente, organizados como um "vetor de vetores"
- Por exemplo, uma matriz de inteiros de nome notas, com quinze elementos, organizados como cinco vetores de três posições cada, é mostrada **esquemáticamente** abaixo

	0	1	2	
notas	75	80	70	0
	90	70	70	1
	87	80	75	2
	56	99	89	3
	80	70	80	4



# Matriz

nome



notas

	0	1	2	
	75	80	70	0
	90	70	70	1
	87	80	75	2
	56	99	89	3
	80	70	80	4



# Matriz

	0	1	2
notas	75	80	70
	90	70	70
	87	80	75
	56	99	89
	80	70	80

0  
1  
2  
3  
4

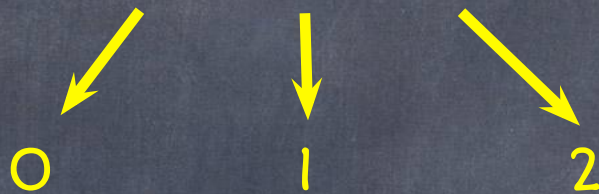
índices  
de  
"vetores"  
(primeira  
dimensão)



# Matriz

índices para cada "vetor"  
(segunda dimensão)

0      1      2



notas

75	80	70	0
90	70	70	1
87	80	75	2
56	99	89	3
80	70	80	4



# Matriz

notas[0][0]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80

A diagram illustrating a 5x3 matrix. The matrix is represented as a grid of cells. The first row is highlighted in pink, and the first cell in that row (75) is highlighted in a darker pink. Yellow arrows point from the text 'notas[0][0]' to the first row and the first column. The matrix values are: Row 0: 75, 80, 70; Row 1: 90, 70, 70; Row 2: 87, 80, 75; Row 3: 56, 99, 89; Row 4: 80, 70, 80.



# Matriz

notas[0][1]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80



# Matriz

notas[0][2]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80



# Matriz

notas[1][2]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80



# Matriz

notas[2][1]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80



# Matriz

notas[4][2]

notas

	0	1	2
0	75	80	70
1	90	70	70
2	87	80	75
3	56	99	89
4	80	70	80



# Declaração de matrizes

- Como toda variável, antes de ser usada uma matriz deve ser declarada
- A declaração de uma matriz informa o nome da matriz, o tipo dos elementos e a quantidade de elementos da matriz em cada dimensão

```
int notas [100] [20];
```



Número de elementos da primeira dimensão  
(fixo na declaração, não muda depois de declarado)



# Declaração de matrizes

- Como toda variável, antes de ser usada uma matriz deve ser declarada
- A declaração de uma matriz informa o nome da matriz, o tipo dos elementos e a quantidade de elementos da matriz em cada dimensão

```
int notas[100][20];
```



Número de elementos da segunda dimensão  
(fixo na declaração, não muda depois de declarado)



# Uso de matrizes

- Exemplos:

```
notas[0][1] = 0;
```

```
notas[0][0]++;
```

```
notas[1][2] = notas[2][1];
```

```
scanf("%d", &notas[i][j]);
```

```
if (notas[i][j] > media)  
    printf("%d\n", notas[i][j]);
```



# Uso de matrizes

- Exemplos:

```
int i, j, notas[100][3];
```

```
for (i = 0; i < 100; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &notas[i][j]);
```



# Tarefa 1 - Siga Mapa (agora sim!)

- Escreva um programa que:
  - Leia uma linha contendo dois inteiros M e N (máximo 10).
  - A seguir leia M linhas com N inteiros cada uma.
  - Imprima o número de movimentos necessários para chegar até o fim do caminho.
  - Assuma que o caminho sempre começa no canto superior esquerdo do mapa.

1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	1	1	0
1	0	0	0	0	0	1	1
1	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1



```

// Dia 5
// siga o mapa

#include <iostream>

using namespace std;

int n, m;
int mapa[12][12];

int main() {

    scanf("%d%d", &n, &m);
    // inicializar o mapa

    for (int j=0; j<n+2; j++) {
        mapa[0][j] = 0;
    }
    for (int j=0; j<n+2; j++) {
        mapa[n-1][j] = 0;
    }
    for (int i=0; i<n-2; i++) {
        mapa[i][0] = 0;
    }
    for (int i=0; i<n-2; i++) {
        mapa[i][n-1] = 0;
    }

    // ler o mapa
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=m; j++) {
            scanf("%d", &mapa[i][j]);
        }
    }

    bool ok = true;
    int i = 1, j = 1;
    int contador = 0;
    while (ok) {
        if (mapa[i+1][j] == 1) {
            mapa[i][j] = 2;
            i++;
            contador++;
        }
        else if (mapa[i-1][j] == 1) {
            mapa[i][j] = 2;
            i--;
            contador++;
        }
        else if (mapa[i][j+1] == 1) {
            mapa[i][j] = 2;
            j++;
            contador++;
        }
        else if (mapa[i][j-1] == 1) {
            mapa[i][j] = 2;
            j--;
            contador++;
        }
        else {
            ok = false;
        }
    }

    // imprime resposta
    printf("%d\n", contador);

    return 0;
}

```



# Tarefa 3 - Jogo da Velha

- Escreva um programa que:
  - Leia 3 linhas contendo três inteiros (separados por espaço) com valores 0 ou 1, representando O e X no jogo da velha.
  - Verifique se O ou X ganhou ou se deu empate.
  - Imprima uma linha contendo 'X' se o 1 ganhou, 'O' se o 0 ganhou ou 'V' se deu velha.



# Tarefa 4 - Média de Cada Aluno

- Escreva um programa que:
  - Leia uma linha contendo um inteiro N (máximo 100).
  - A seguir leia N linhas com 3 inteiros (separados por espaço), representando as notas de três provas (P1, P2 e P3) de um aluno (inteiros de 0 a 100).
  - Para cada aluno, imprima uma linha contendo a média do aluno considerando as três provas.



# Tarefa 4 - Média de Cada Aluno

- Não precisa armazenar, então não precisa usar vetor ou matriz!
- **Cuidado:** somente armazene valores se for necessário.



# Tarefa 5 - Aprovado/Reprovado

- Escreva um programa que:
  - Leia uma linha contendo um inteiro N, (máximo 100).
  - A seguir leia N linhas com 3 inteiros (separados por espaço), representando as notas de três provas (P1, P2 e P3) de um aluno (inteiros de 0 a 100).
  - Para cada aluno, imprima uma linha contendo a palavra "Aprovado" se sua Nota (média das três provas) é maior do que a Nota Média da turma (média das Notas dos alunos), ou "Reprovado" caso contrário.



# Tarefa 5 - Aprovado/Reprovado

- Também não precisa armazenar todas as notas, basta armazenar todas as médias
  - Usar um vetor de médias