
Projeto e Desenvolvimento de Algoritmos

Entrada e Saída de Dados

Adriano Cruz e Jonas Knopman

cederj

Índice

- **Objetivos**
- **Comandos de Entrada de Dados**
- **Comandos de Saída de Dados**

Objetivos

- Apresentar os comandos de entrada e saída de dados
- Mostrar como formatar a saída de dados para facilitar o entendimento dos resultados

Entrada e Saída de Dados

- Algoritmos úteis normalmente devem receber dados para processar (entrada de dados)
- Algoritmos devem devolver os resultados das transformações executadas nestes mesmos dados (saída de dados)

Entrada de Dados

- Em um programa os dados podem ser lidos de um teclado, de um arquivo no disco rígido, disquete, etc.
- Abaixo mostramos exemplos de periféricos de entrada que em alguns casos podem também ser de saída.
- Você é capaz de dizer qual (ou quais) destes periféricos é (são) de entrada e saída?



cederj

Saída de Dados

- Os dados de saída podem ser escritos em uma impressora, terminal de vídeo, disquete, disco rígido etc.



Entrada e Saída de Dados

- No pseudo-código que estamos usando a origem e o destino dos dados não é importante.

Fluxos de dados

- Várias linguagens consideram que os algoritmos (programas) recebem dados de um fluxo de dados de entrada e enviam os seus dados para um fluxo de dados de saída
 - Estes fluxos são organizados como seqüências de linhas de caracteres
 - Que periféricos geram e recebem estes fluxos é dependente do programador e da implementação e configuração do compilador
- cederj

Fluxos de dados cont.

- Na maior parte dos casos estes fluxos são lidos de maneira semelhante a uma fita cassete áudio sem *rewind*.



- O computador avança lendo os dados na ordem em que são fornecidos pelo usuário.
- Devemos então considerar que uma vez lido um dado o computador avança para o próximo e não existe a possibilidade de retorno.

Fluxos de dados cont.

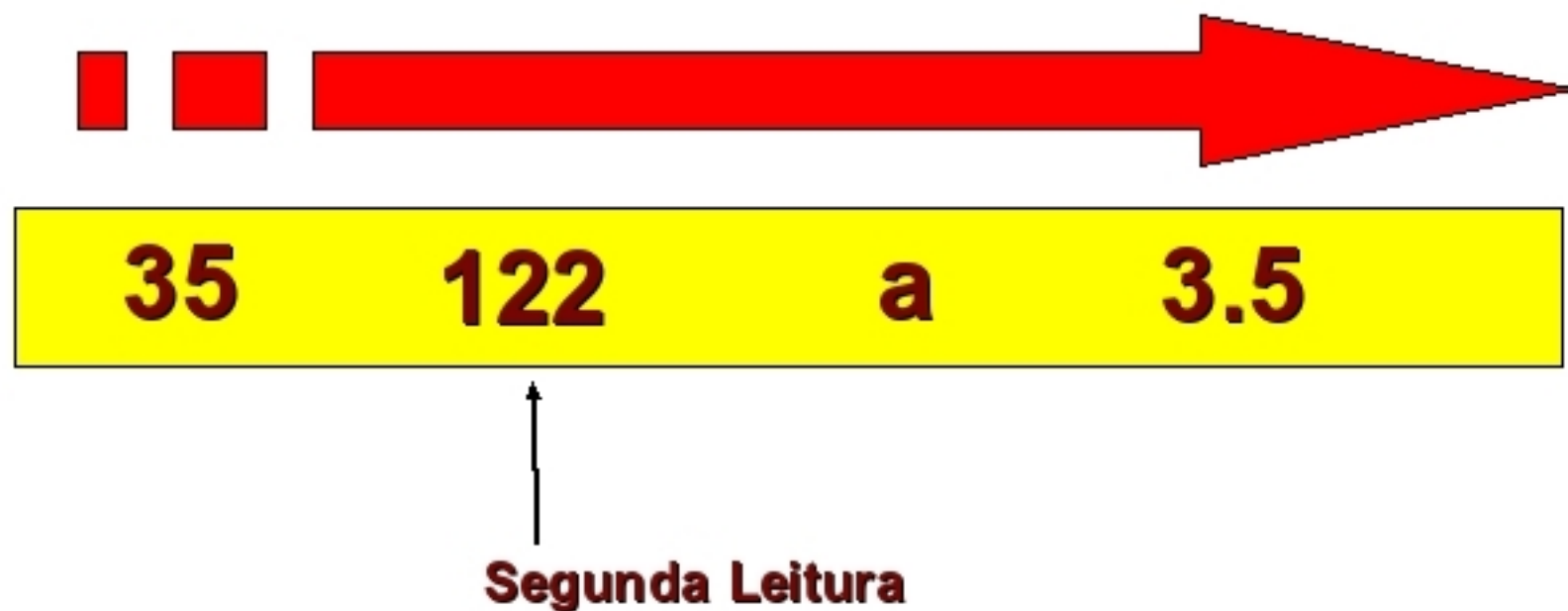
- A separação entre os dados é feita por um ou mais espaços em branco.



| | | | |
|-----------|------------|----------|------------|
| 35 | 122 | a | 3.5 |
|-----------|------------|----------|------------|

↑
Primeira Leitura

Fluxos de dados cont.



cederj

Fluxos de dados cont.



↑
Terceira Leitura

cederj

Fluxos de dados cont.



| | | | |
|----|-----|---|-----|
| 35 | 122 | a | 3.5 |
|----|-----|---|-----|

↑
Quarta Leitura

cederj

Comandos de Entrada e Saída

- Em pseudo-código iremos empregar dois comandos básicos:

- ❖ Comando de leitura de dados

- ❖ Comando de saída de dados

Forma do comando de leitura

- A forma geral do comando de leitura é a seguinte:

leia <lista de variáveis>

- **<lista de variáveis>** é um conjunto de nomes de variáveis separados por vírgulas.
- A lista de variáveis pode conter variáveis de todos os tipos válidos: inteiras, reais, caracteres

Comando de leitura

leia comprimento, largura, altura

- A ação do comando de leitura acima é obter do fluxo de dados de entrada (teclado?) três valores numéricos e atribuí-los, na ordem em que foram obtidos, às três variáveis **comprimento, largura e altura**
- Como os dados são lidos de um fluxo, as próximas leituras obterão seus dados do ponto a seguir às últimas leituras.

Exemplo de comando de leitura

Início

leia altura, largura

leia quantidade

- Caso estivéssemos lendo os dados de um teclado, o usuário teria de fornecer três valores
- Por exemplo: 5.5, 3.1 e 10
- Estes comandos de leitura são equivalentes à:

altura ← 5.5

largura ← 3.1

quantidade ← 10

cederj

Forma do comando de escrita

- A forma geral do comando de escrita é a seguinte:

imprima <lista de saída>

- **<lista de saída>** é uma lista de valores separados por vírgulas.
- Os valores podem ser variáveis e/ou constantes de todos os tipos válidos

Comando de escrita

`imprima 'A área vale', area`

- O comando de escrita acima imprime primeiro a constante cadeia de caracteres

`'A área vale'`

- Em seguida imprime o valor da variável `area` que está armazenado na memória
- Assumindo que `area` valha 4.5 teríamos na saída o seguinte resultado

`A área vale 4.5`

cederj

Uso de espaços em branco

- É importante facilitar a leitura dos resultados para os usuários.
- Procurar então colocar informações adicionais que expliquem o significado dos resultados.
- Usar espaços em branco para separar resultados.
- Por exemplo:

```
imprima 'Lado =',lado,' comprimento =', comp
```

- Na tela sairia o seguinte resultado:

```
Lado = 3.5 comprimento = 10.5
```


Usando espaços em branco

Início

Index \leftarrow 17

imprima Index, Index

imprima Index, ' ', Index

Conta \leftarrow 27.5678

imprima Conta, Conta

imprima Conta, ' ', Conta

letra \leftarrow 'Z'

imprima letra, letra

imprima letra, ' ', letra

fim

cederj

Usando espaços em branco

- No próximo slide os resultados dos comandos em **vermelho** estão em **vermelho**.
- Observe como os espaços em branco facilitou a leitura dos resultados.

Espaços em branco (a saída)

1717

17 17

27.567827.5678

27.5678 27.5678

ZZ

Z Z

cederj