

Linguagem de Programação I

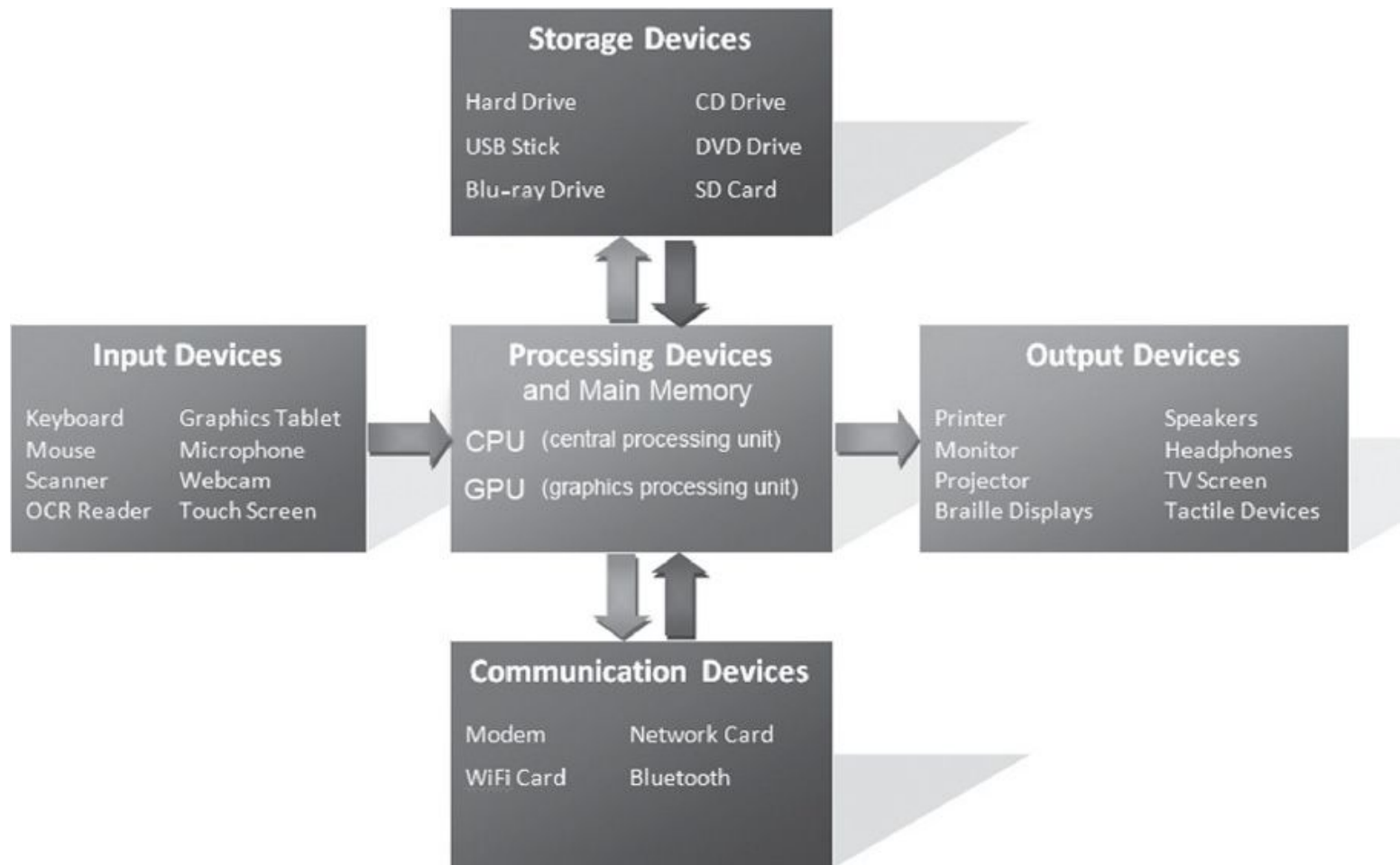
História e Conceitos

A Máquina Universal

- Um computador moderno pode ser definido como:
 - “uma máquina que armazena e manipula informação sob controle de um programa mutável”.
- O computador é uma máquina que aceita dados (entrada) e os processa em informação útil (saída), seguindo a sequência de instruções de um **software ou programa de computador**.
- Dois elementos chaves:
 - Computadores e periféricos para manipular informação.
 - Computadores operam sob controle de um programa mutável.

Máquina Universal

- Componentes Fundamentais



Hardware & Software

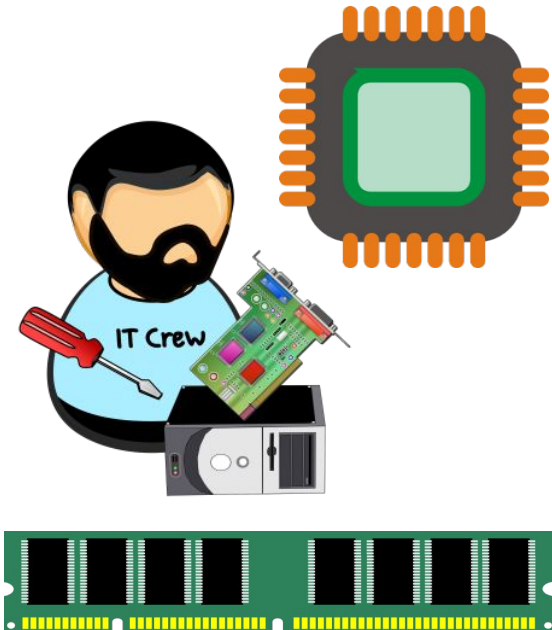
-ware

1. Artigos do mesmo tipo geral, feitos de um determinado material ou usados em um aplicativo específico: **glassware; silverware; tableware**

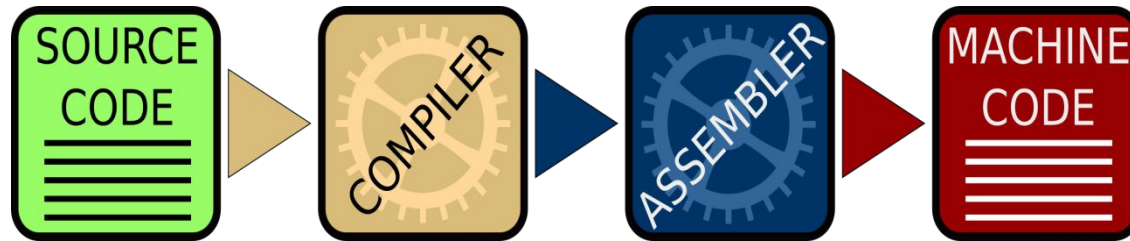
Hardware

1. Ferramentas, máquinas e outros equipamentos **duráveis**.

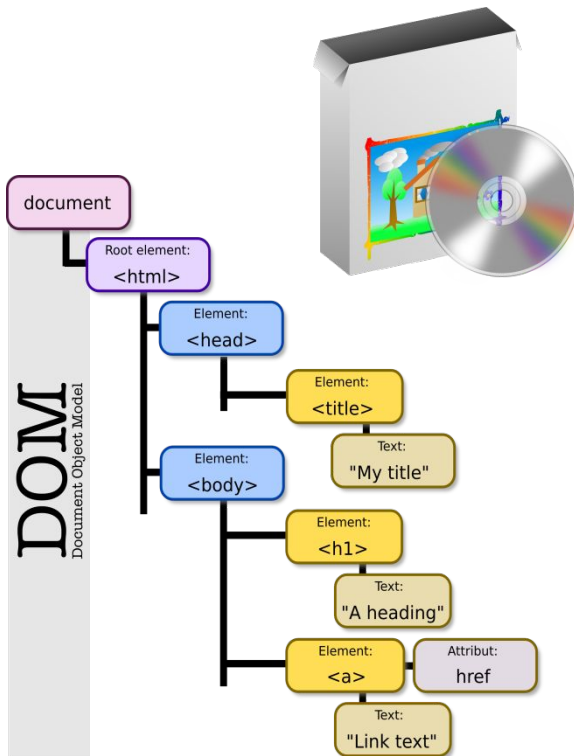
2. Máquinas, fiação e outros **componentes físicos** de um computador ou outro sistema eletrônico.



Hardware & Software



Software



1. Programas, rotinas e linguagens simbólicas que controlam o funcionamento do hardware e direcionam seu funcionamento.

2. Programas para organiza e executar a operação de um computador ou processar dados

Programa

Programa

1. Uma **sequência de instruções codificadas** alimentadas em um computador, permitindo que ele execute operações lógicas e aritméticas especificadas em dados

- Para que serve um programa?
 - Um programa serve para traduzirmos o que desejamos em **linguagem de máquina**
- O programa “**instrui**” ao computador o que ele deve fazer
- É sua “receita de bolo”

Programa



Ada Lovelace

Você sabia?

Os primeiros programas de computador foram escritos para um computador mecânico projetado por Charles Babbage em meados dos anos 1800.

A pessoa que escreveu esses programas era uma mulher, **Ada Lovelace**, que era uma talentosa matemática. Assim, ela é referida como “o primeira programadora de computador”.

A linguagem de programação ADA é assim denominada em sua homenagem.

Algoritmo

Algoritmo

- É um número finito de etapas inequívocas “exequíveis”, claramente descritas, que podem ser seguidas sistematicamente para produzir um resultado desejado para uma entrada dada em uma quantidade limitada de tempo.
- Algoritmos, portanto, são **métodos computacionais** gerais usados para resolver instâncias de problemas particulares.
- O algoritmo simplesmente diz o que deve ser feito. Para resolver de fato um problema, **devemos definir como executar os passos do algoritmo.**

Algoritmo



Al-Khwarizmi
(Século IX d.C.)

Você sabia?

A palavra “**algoritmo**” é derivada do nome do matemático árabe do século IX (d.C.), **Al-Khwarizmi**, que trabalhou em “processos escritos para alcançar algum objetivo”. O termo “álgebra” também deriva do termo “al –jabr”, apresentado por ele.

Algoritmo, Código e Programa

- **Algoritmo:** é a lógica a ser implementada, é basicamente a ideia de como resolver um problema.
- **Código:** é o texto escrito em uma dada linguagem de programação que será traduzido para a linguagem do computador (binário).
- **Programa:** é o executável que comunica-se com o computador e executa a ordem.

Linguagem de Programação: Definições

مرحبا العالم! Hallo Welt!
 Hej Värld! Hello World!
 Ciao Mondo
 ハローワールド!
 ¡Holá mundo! 世界您好!
 Salut le Monde!

O que é uma
linguagem de programação???



Linguagem de Programação: Definições

- **Linguagem Natural (LN)**: Maneira de comunicação (escrita, fala) baseada em um código linguístico que foi desenvolvido naturalmente ao longo do tempo.
Exemplos: Português, Inglês, Japonês chinês.
- **Linguagem de Artificial (LA)**: Linguagem criada artificialmente baseada em regras bem definidas, linguagem construída.

https://en.wikipedia.org/wiki/Natural_language

https://en.wikipedia.org/wiki/Constructed_language

Linguagem de Programação: Aspectos

- **Linguagem Natural:**
 - Aceita:
 - Verdadeiro
 - Falso
 - Ambiguidade
- **Linguagem de Programação (LA):**
 - Aceita:
 - Verdadeiro
 - Falso

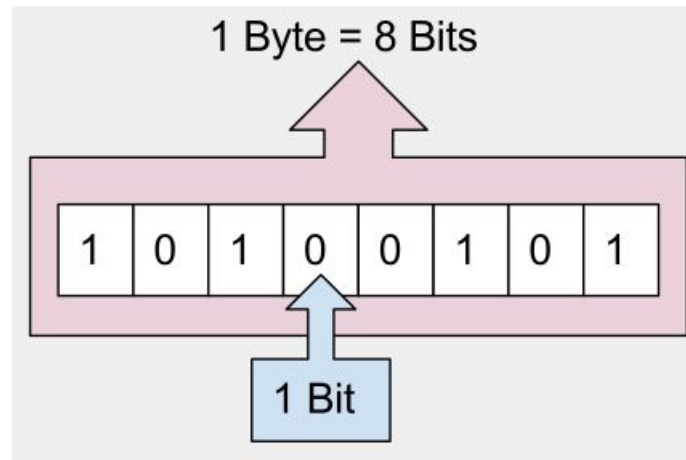
Linguagem de Programação: Comunicação

Comunicação com o computador:

- ↓ Algoritmo;
- ↓ Linguagem de programação (código);
- ↓ Tradução para linguagem de computador (interpretador/compilador);
- ↓ Linguagem de computador (binário):
Verdadeiro (1) ou Falso (0);
→ Resultados.

Linguagem de Programação: Byte

- **Bit (binary digit)**: é a menor unidade de informação que pode ser armazenada ou transmitida em um computador:
0 ou 1
- **Byte (binary term)**: conjunto de 8 bits, é como os dados são armazenados:



<https://ro.wikipedia.org/wiki/Bit>

Sintaxe e Semântica

- **Sintaxe**: o termo "sintaxe" também é usado para referir o estudo das regras que regem o comportamento de sistemas matemáticos, como a lógica, e as linguagens de programação de computadores.
- **Semântica**: o estudo do significado.
[1] Incide sobre a relação entre significantes, tais como palavras, frases, sinais e símbolos, e o que eles representam, a sua denotação.

<https://pt.wikipedia.org/wiki/Sintaxe>

<https://pt.wikipedia.org/wiki/Sem%C3%A2ntica>

Sintaxe e Semântica

- **Sintaxe:**
 - Significado das palavras e símbolos individualmente
 - Basicamente: O que são as partículas.

- **Semântica:**
 - Significado sobre contexto, como se organiza o texto (em LN) ou código (em LP)
 - Basicamente: Como as partículas interagem entre si.

<https://pt.wikipedia.org/wiki/Sintaxe>

<https://pt.wikipedia.org/wiki/Sem%C3%A2ntica>

Sintaxe e Semântica

- A sintaxe de uma linguagem é um conjunto de caracteres e os arranjos aceitáveis (ou sequências) desses caracteres.
 - O português inclui as letras do alfabeto, pontuação, palavras corretamente escritas e frases adequadamente estruturadas.

“ontem o pão comprar ir”

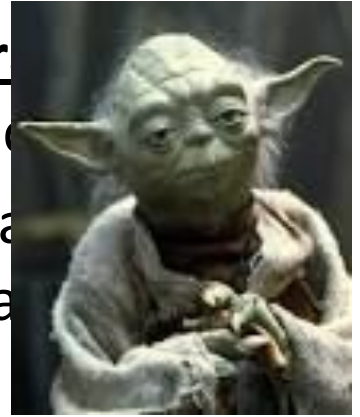
Sintaticamente incorreta

“fui comprar o pão ontem.”

Sintaticamente Correta

Sintaxe e Semântica

- A sintaxe de uma linguagem é um conjunto de regras que define os arranjos aceitáveis (ou sequências) desses caracteres.
 - O português inclui as letras do alfabeto, pontuação, palavras corretamente escritas e frases adequadamente estruturadas.



“ontem o pão comprar ir” - Yoda, 1004 BBY

Sintaticamente incorreta

“fui comprar o pão ontem.”

Sintaticamente Correta

Sintaxe e Semântica: importância

- A semântica de uma linguagem é o significado associado a cada sequência de caracteres sintaticamente correta.

“Num calor girassol com alegria” - Gilberto Gil, 1968

Sintaticamente correta
Semanticamente incorreta

“No calor, o girassol me dá alegria”

Sintática e semanticamente correta

Exemplos de Linguagens de Programação

- Problema: Exibir uma tabuada de 1 a 10 de um número inserido pelo usuário.

Pseudolinguagem :

leia num

para n de 1 até 10 passo 1 faça

 tab <- num * n

 imprime tab

fim faça

Exemplos de Linguagens de Programação

Linguagem C:

```
scanf (&num) ;
for (n = 1; n <= 10; n++) {
    tab = num * n;
    printf ("\n %d", tab) ;
} ;
```

Exemplos de Linguagens de Programação

Basic:

```

10 input num
20 for n = 1 to 10 step 1
30 let tab = num * n
40 print chr$ (tab)
50 next n

```

Exemplos de Linguagens de Programação

Fortran:

```
read (num) ;  
do 1 n = 1:10  
    tab = num * n  
    write(tab)  
10 continue
```


Exemplos de Linguagens de Programação

Assembly para INTEL 8088:

```

MOV CX, 0
IN  AX, PORTA
MOV DX, AX
LABEL:
INC CX
MOV AX, DX
MUL CX
OUT AX,  PORTA
CMP CX, 10
JNE LABEL
    
```

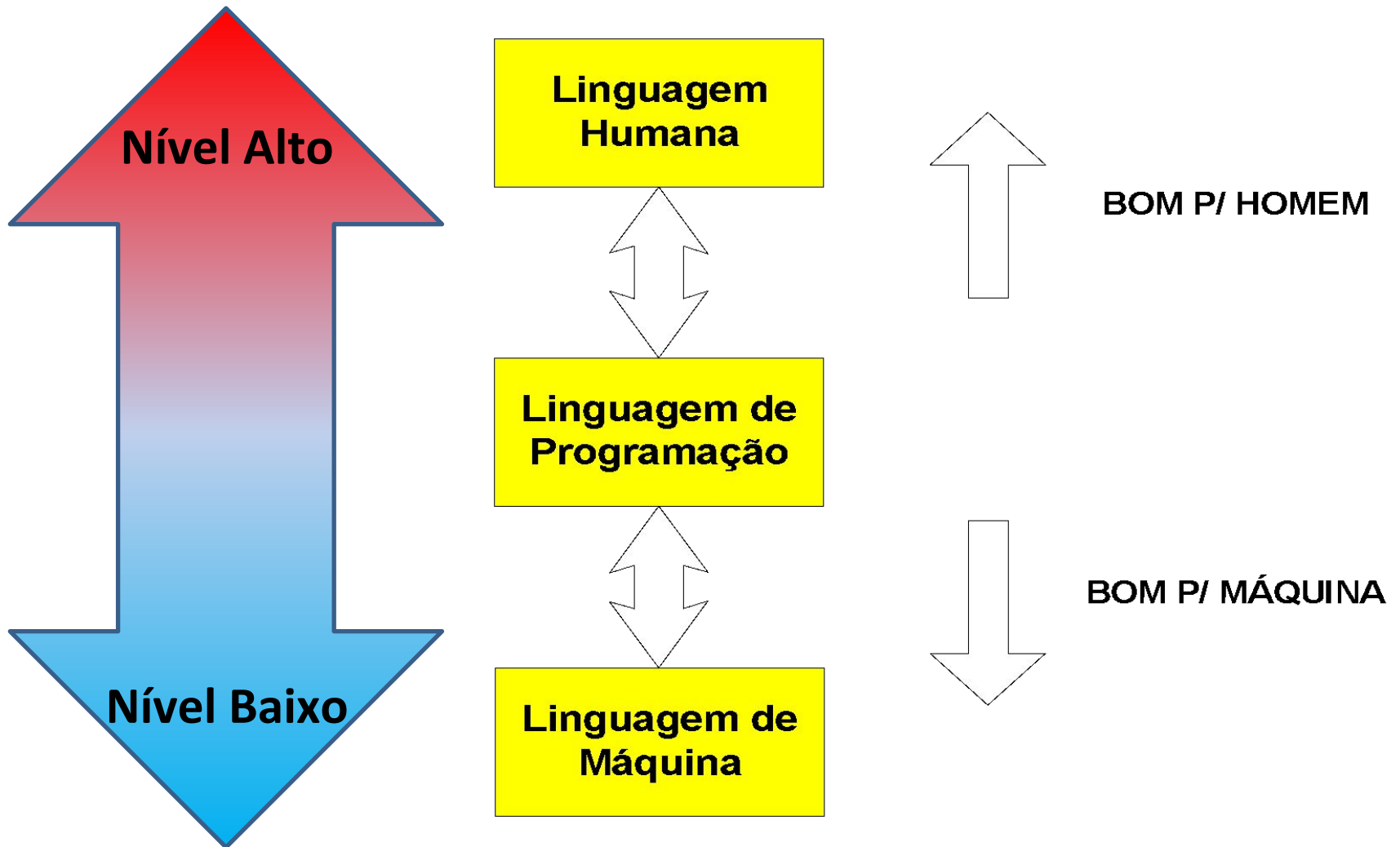
Exemplos de Linguagens de Programação

Python:

```
num = int(input())  
for i in range(1,11):  
    tab = num * i  
    print(tab)
```

Classificação

Em relação ao Nível



Classificação das Linguagens de Programação

- **Baixo Nível**
 - Linguagens mais próxima de linguagem da máquina.
 - Escritas utilizando as instruções do próprio microprocessador

Classificação das Linguagens de Programação

- **Baixo Nível**
 - Vantagens:
 - Os programas são executados com maior velocidade de processamento
 - Os programas ocupam menos espaço na memória
 - Desvantagens:
 - Pouca (ou nenhuma) portabilidade;
 - Programação pouco amigável a seres humanos;
 - Muitas linhas de código para execução;
 - Microgerenciamento dos recursos da máquina.
 - Ex.: **Assembly**

Classificação das Linguagens de Programação

- **Médio Nível**
 - Linguagens mais amigáveis a seres humanos e máquinas;
 - São uma combinação entre as linguagens de Alto Nível e as de Baixo Nível:
 - Pode-se acessar registros do sistema e trabalhar com endereços de memória (características de linguagens de baixo nível) e ao mesmo tempo realizar operações de alto nível (`if...else; while; for`)
 - Contêm comandos simples e complicados

Classificação das Linguagens de Programação

- **Médio Nível**
 - Vantagens:
 - Geralmente são linguagens poderosas, permitindo a criação de diversos softwares, desde jogos a programas de alta performance
 - Desvantagens:
 - Alguns comandos têm uma sintaxe um pouco difícil de compreender;
 - Menos instruções que as de baixo nível;
 - Ainda possui microgerenciamento de memória no caso da linguagem C e similares.
 - Ex.: **Linguagem C**

Classificação das Linguagens de Programação

- **Alto Nível**

- Linguagens voltadas para o ser humano. Em geral utilizam sintaxe mais estruturada, tornando o seu código mais fácil de entender
- São linguagens independentes de arquitetura.
 - Um programa escrito em uma linguagem de alto nível, pode ser migrado de uma máquina a outra sem nenhum tipo de problema (portabilidade)
- Permitem ao programador “**esquecer**” completamente do funcionamento interno da máquina
 - Sendo necessário um tradutor que entenda o código fonte e as características da máquina

Classificação das Linguagens de Programação

- **Alto Nível**
 - Vantagens:
 - Por serem compiladas ou interpretadas, têm maior portabilidade, podendo ser executados em várias plataformas com pouquíssimas modificações
 - Em geral, a programação é mais fácil
 - Desvantagens:
 - Em geral, as rotinas geradas (em linguagem de máquina) são mais genéricas e, portanto, mais complexas e por isso são mais lentas e ocupam mais memória
 - Ex.: Java, C#, C++, **Python**

Classificação das Linguagens de Programação

Em relação ao Paradigma

O que é um “paradigma”?

Paradigma é um **modelo interpretativo** (ou conceptualização) de uma realidade.

Pode dizer-se que um paradigma é um **ponto de vista**: um ponto de vista que determina como uma realidade é entendida e como se atua sobre ela.

Classificação das Linguagens de Programação

- **Imperativo/Estruturado:**
 - Orientadas a ações, onde a computação é vista como uma sequência de instruções que manipulam valores de variáveis (leitura e atribuição)
 - Também denominado de procedural, por incluir sub-rotinas ou procedimentos como mecanismo de estruturação.
 - Baseia-se na arquitetura de computadores Von Neumann
 - Ex.: Basic, Pascal, C, Ada, Fortran, Cobol, Assembly...

Classificação das Linguagens de Programação

- **Funcional:**
 - Trata a computação como um processo de avaliação de funções matemáticas, evitando o uso de estados ou dados mutáveis
 - Consistem em pensar qual função deve ser aplicada para transformar uma entrada qualquer na saída desejada
 - Ex.: Lisp, Haskell, Erlang, Scheme...

Classificação das Linguagens de Programação

- **Lógico:**
 - Baseado em lógica formal
 - Equivalente à descrição do problema expressa de maneira formal, similar à maneira que o ser humano raciocinar sobre ele
 - Requer um estilo mais descritivo
 - Programas descrevem um conjunto de regras que disparam ações quando suas premissas são satisfeitas
 - Ex.: Prolog

Classificação das Linguagens de Programação

- **Orientado a objetos:**
 - Tratam os elementos e conceitos associados ao problema como objetos
 - Objetos são entidades abstratas que embutem dentro de suas fronteiras, as características e operações relacionadas com a entidade real;
 - Sugere a diminuição da distância entre a modelagem computacional e o mundo real
 - Uma aplicação é estruturada em módulos (classes) que agrupam um estado (atributos) e operações (métodos) sobre este
 - A classe é o modelo ou molde de construção de objetos. Ela define as características e comportamentos que os objetos irão possuir (herança).
 - Ex. C++, Java, C#, Eiffel, Smalltalk, **Python**

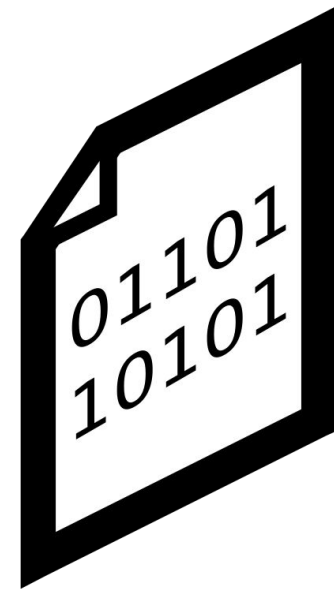
Classificação das Linguagens de Programação

Em relação a Estrutura de Tipos

- **Forte** / Fraca:
 - **Fracamente tipada:** o tipo da variável não importa, permitindo operações sem conversão explícita.
 - **Fortemente tipada:** toda variável tem um tipo e ele é importante durante a codificação.
- **Dinâmica** / Estática:
 - **Dinamicamente tipada:** o tipo da variável pode ser alterado durante a execução do programa.
 - **Estaticamente tipada:** uma vez definido, o tipo da variável não pode ser alterado em tempo de execução.
- **Inferência de tipo:**
 - **Sem inferência de tipo:** o tipo da variável deve ser explicitamente declarado.
 - **Com inferência de tipo:** o tipo da variável é automaticamente reconhecido pelo interpretador/compilador.

Implementação das Linguagens de Programação

Como fazer uma computador entender uma série de comandos escritos em texto?



Implementação das Linguagens de Programação

O computador precisa converter o código da linguagem de alto nível para algo que ele entenda:

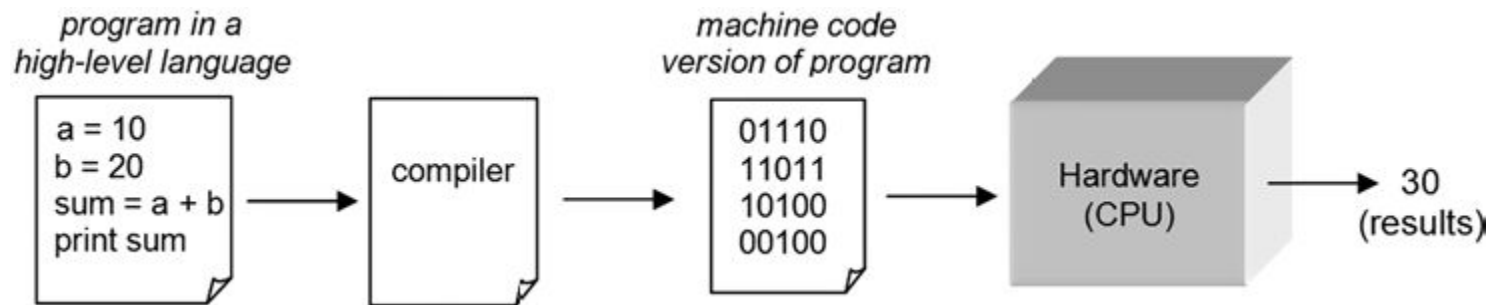
- ↓ Um programa (compilador) faz a conversão das instruções do código em alto nível (texto) para instruções binárias;
- ↓ Efetivamente o computador executa a instrução;
- + Resultado.

Ao programa original, em linguagem de alto nível, dá-se o nome de **Código/Programa Fonte** (Source Code) e ao resultado, em linguagem de máquina, de **Código/Programa Objeto**.

Implementação das Linguagens de Programação

- **Compilador:**

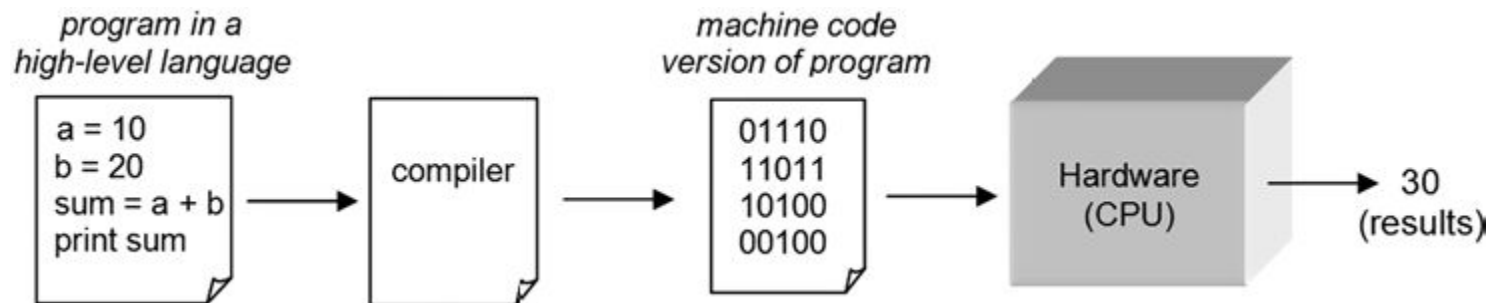
O programa **conversor** recebe a primeira instrução do programa fonte, confere-a para ver se está escrita corretamente, converte-a **para linguagem de máquina** em caso afirmativo e passa para a próxima instrução, repetindo o processo sucessivamente até a última instrução do programa fonte.



Implementação das Linguagens de Programação

- **Compilador:**

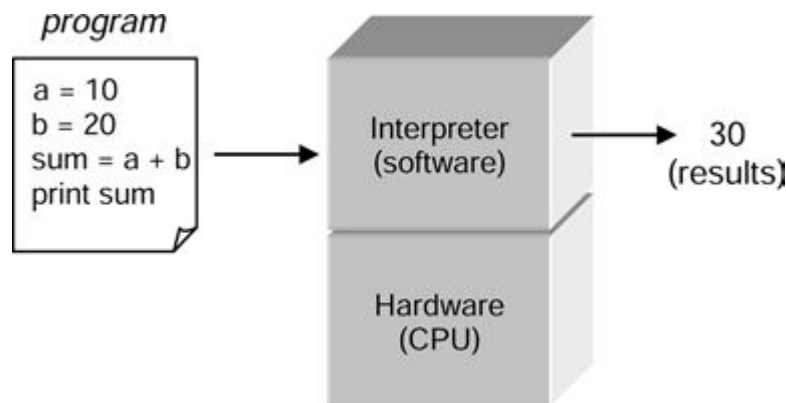
Se este programa for executado uma segunda vez, não haverá necessidade de uma nova tradução, uma vez que todos os comandos em linguagem binária foram memorizados em um novo programa completo.



Implementação das Linguagens de Programação

- **Interpretador:**

O programa conversor recebe a primeira instrução do programa fonte, confere para ver se está escrita corretamente, converte-a em linguagem de máquina e então ordena ao computador que execute esta instrução. Depois repete o processo para a segunda instrução, e assim sucessivamente, até a última instrução do programa fonte. Quando a segunda instrução é trabalhada, a primeira é perdida, isto é, **apenas uma instrução fica na memória em cada instante.**



Implementação das Linguagens de Programação

- **Compilação vs. Interpretação:**
 - Uma vez que um programa tenha sido compilado, ele pode ser executado várias e várias vezes sem o seu código-fonte ou compilador.
 - Se o programa é interpretado, tanto o código-fonte quanto o interpretador são necessários cada vez que o programa é executado.
 - Programas compilados geralmente executam mais rápido desde que a tradução do código-fonte já ocorreu anteriormente durante o processo de compilação.

Implementação das Linguagens de Programação

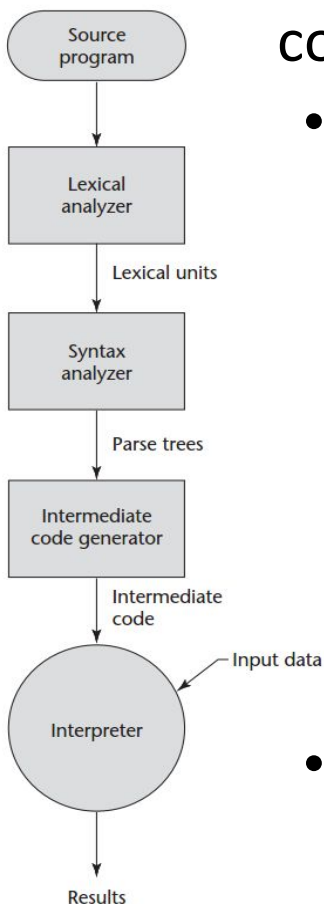
- **Compilação vs. Interpretação:**
 - Linguagens interpretadas são parte de ambientes de programação mais flexíveis, pois podem ser desenvolvidas e executadas interativamente.
 - Programas interpretados são mais portáteis.
 - Isso significa que o código executável produzido por um compilador para um processador **AMD** não executa num processador **Intel** sem que o código-fonte seja recompilado (isso se forem de arquiteturas diferentes).
 - Se um interpretador existir, então o código-fonte pode ser executado sem nenhuma alteração.

Implementação das Linguagens de Programação

- Compilação vs. Interpretação:

- Existe ainda uma terceira classe de linguagens que são compiladas e interpretadas

- Neste método, o computador ao invés de armazenar as instruções do programa fonte tal como lhe são fornecidas, ele transforma-as em **códigos intermediários**, que não exigem tanto espaço de memória como as instruções originais. Estas instruções intermediárias são em seguida transformadas em linguagem de máquina e executadas, funcionando daí por diante como no **INTERPRETADOR**.



- Método conhecido como “Tradutor”

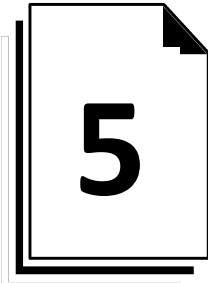
Implementação das Linguagens de Programação

- Linguagens Compiladas:
 - C, Pascal
- Linguagens Interpretadas
 - JavaScript, BASIC, **Python**, Perl
- Linguagens Compiladas e Interpretadas
 - Java, C#

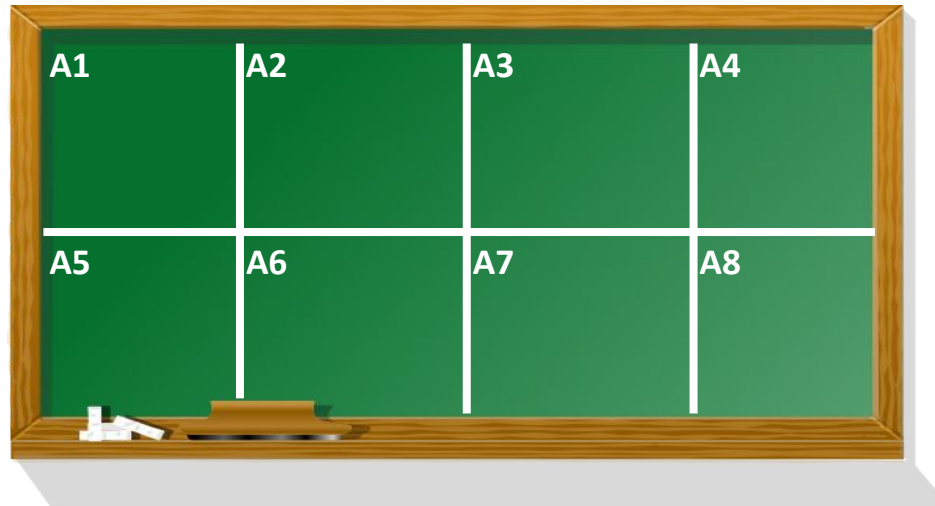
Computador Simplificado

Computador Simplificado

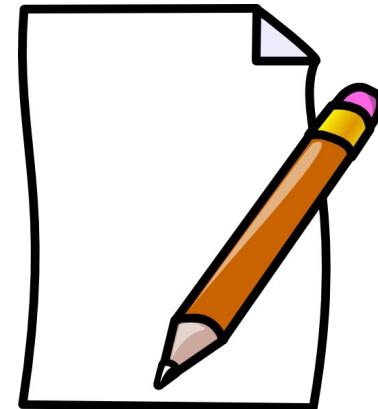
Cartões
Disponíveis



Cartões
Utilizados



Saída

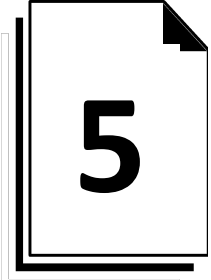


Lista de Instruções

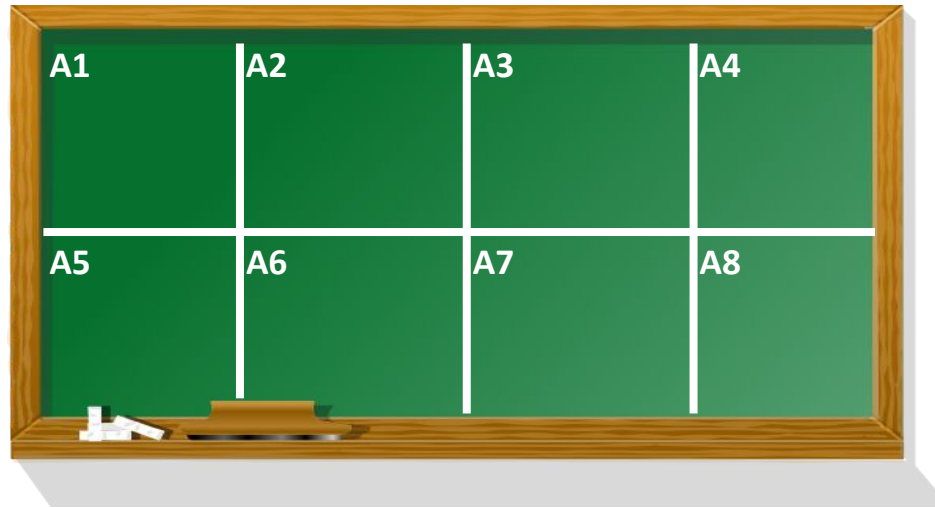
Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

Computador Simplificado

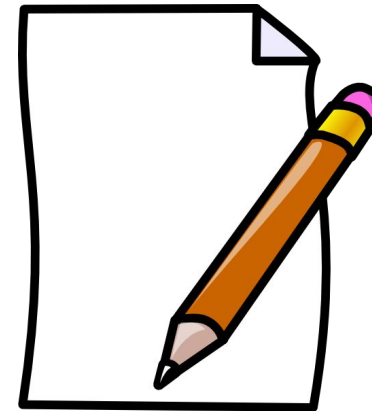
Cartões
Disponíveis



Cartões
Utilizados



Saída



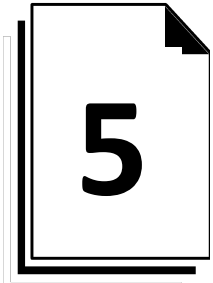
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

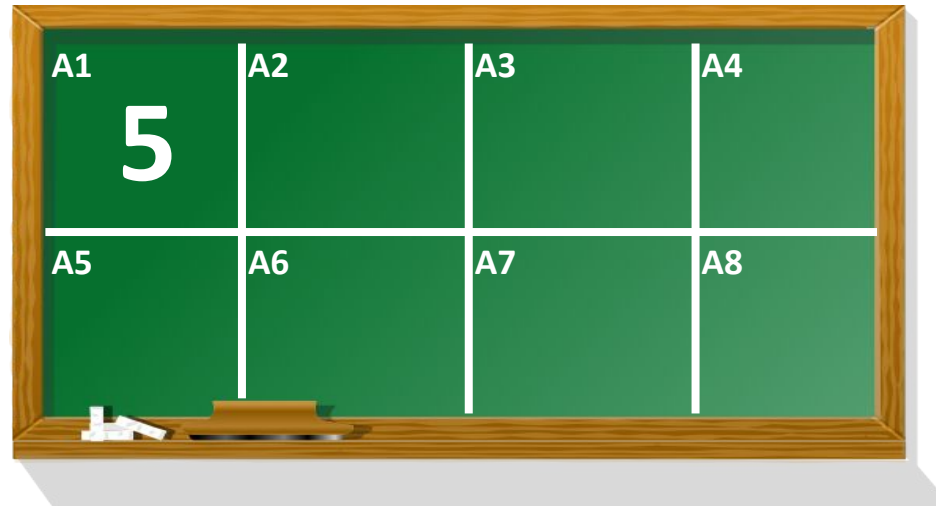


Computador Simplificado

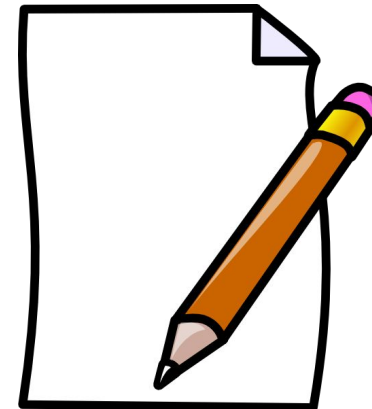
Cartões
Disponíveis



Cartões
Utilizados



Saída



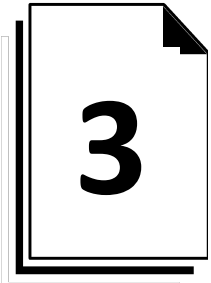
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

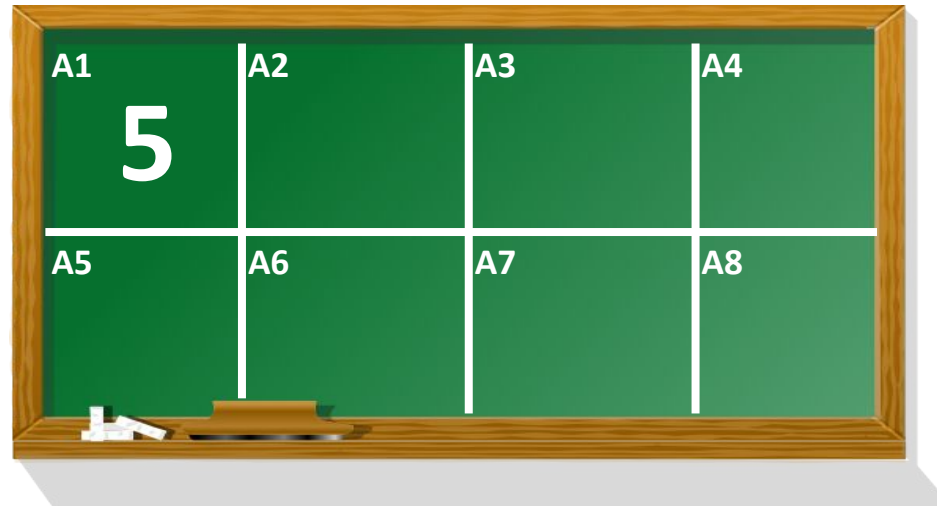


Computador Simplificado

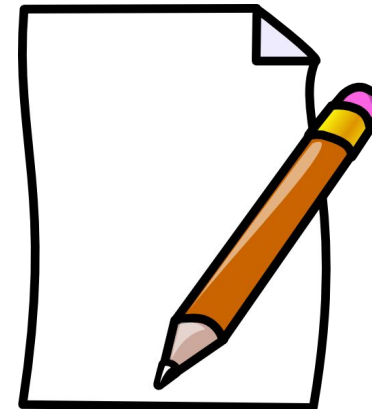
Cartões
Disponíveis



Cartões
Utilizados



Saída



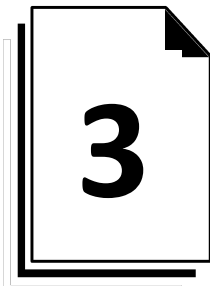
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

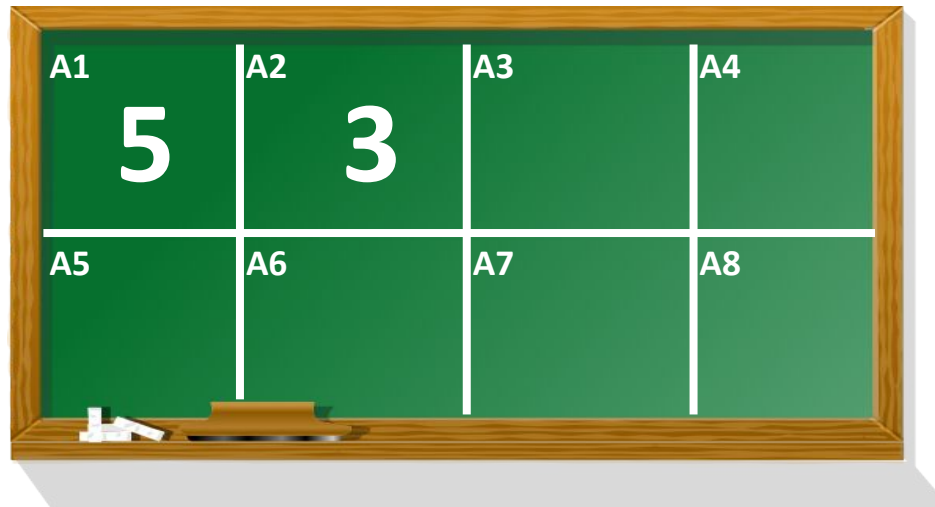


Computador Simplificado

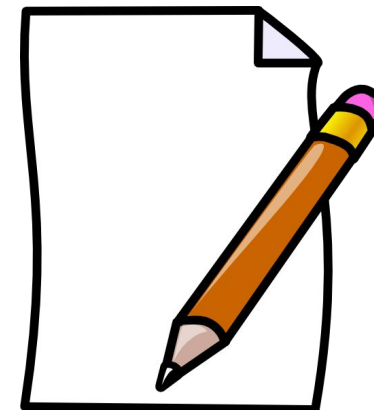
Cartões
Disponíveis



Cartões
Utilizados



Saída



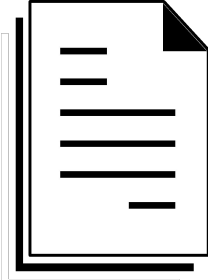
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

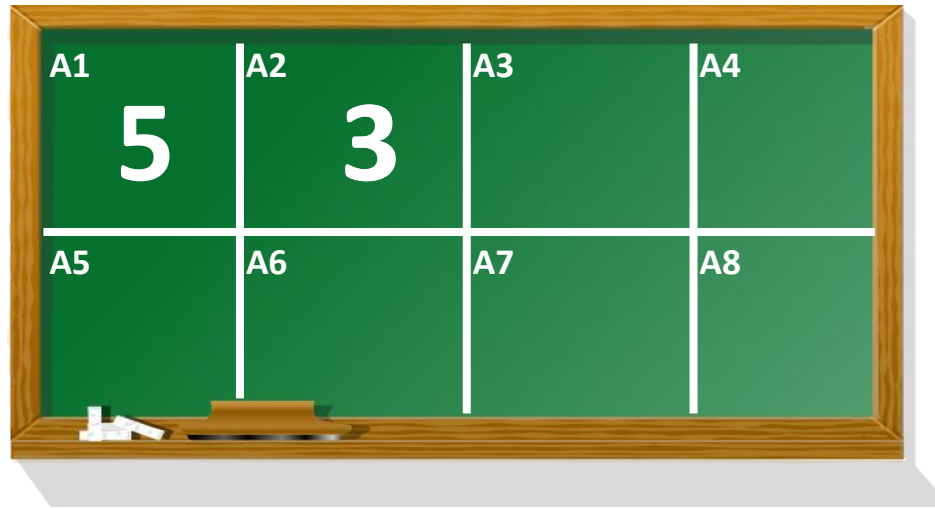


Computador Simplificado

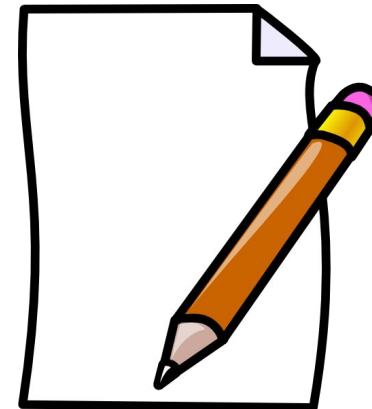
Cartões
Disponíveis



Cartões
Utilizados



Saída



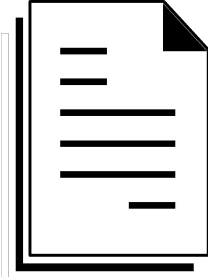
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

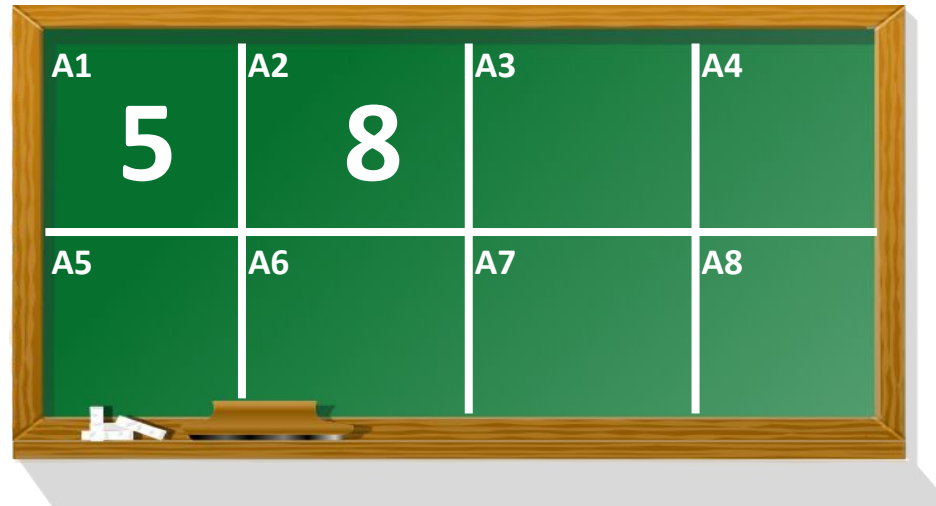


Computador Simplificado

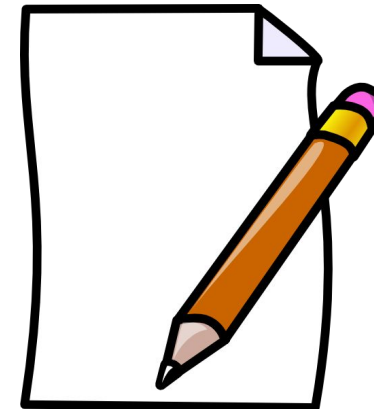
Cartões
Disponíveis



Cartões
Utilizados



Saída



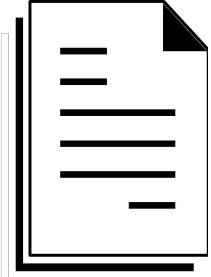
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

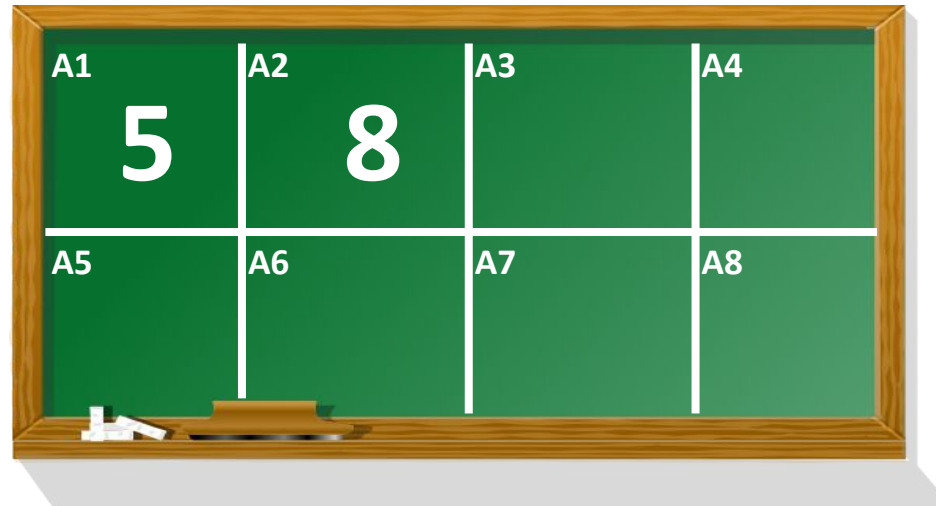


Computador Simplificado

Cartões
Disponíveis



Cartões
Utilizados



Saída



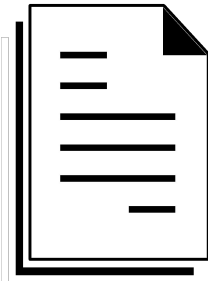
Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2

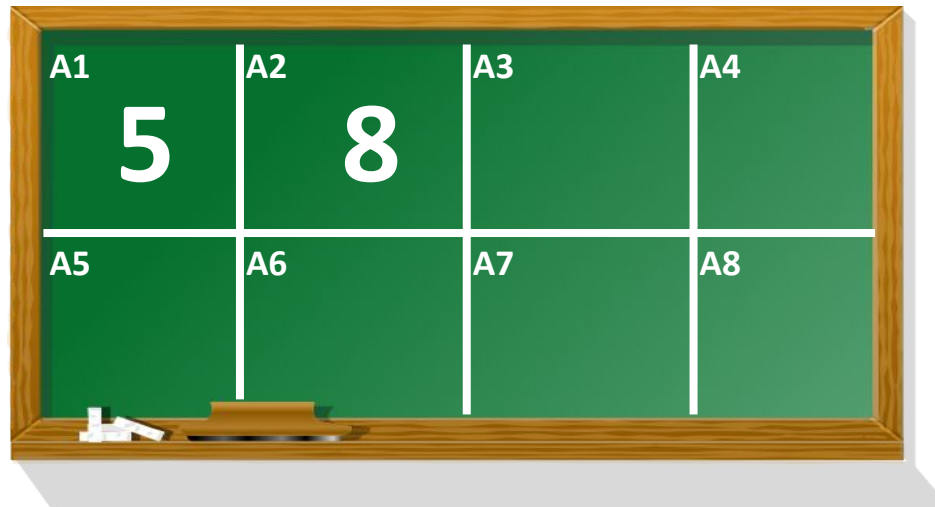


Computador Simplificado

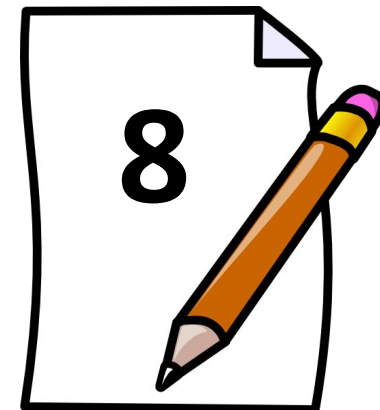
Cartões
Disponíveis



Cartões
Utilizados



Saída



Lista de Instruções

Pos.	Instrução
P1	Leia um cartão e copie seu valor em A1
P2	Leia um cartão e copie seu valor em A2
P3	Some o conteúdo de A1 com o de A2 e escreva o resultado em A2
P4	Imprima o conteúdo de A2



Computador Simplificado

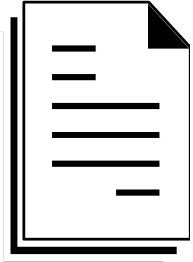
- Exercícios -

Escreva as instruções para resolver os problemas a seguir

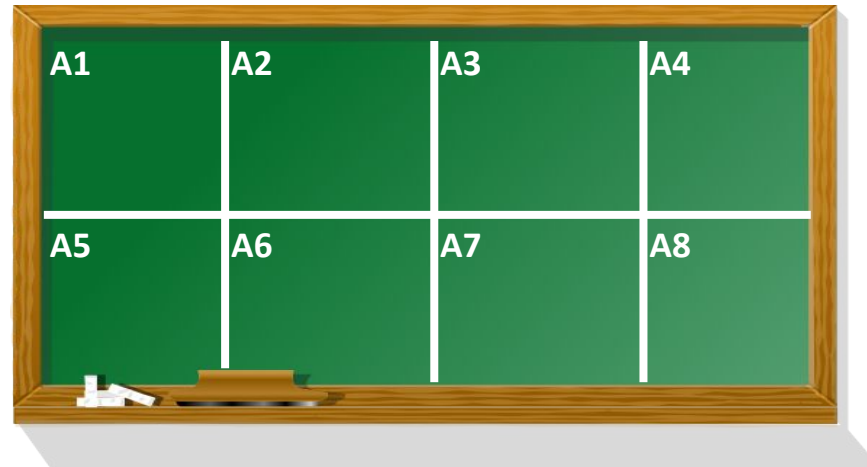
Ler 3 valores, somar os dois primeiros, subtrair desse resultado o terceiro valor e exibir o resultado final:



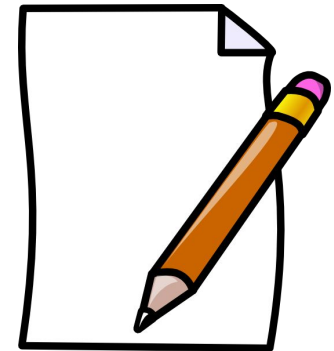
Cartões
Disponíveis



Cartões
Utilizados



Saída



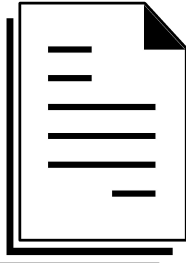
Lista de Instruções

Instrução	
P1	Leia um cartão e copie seu valor em A1
P2	...
P3	...
.	...
.	...
.	...

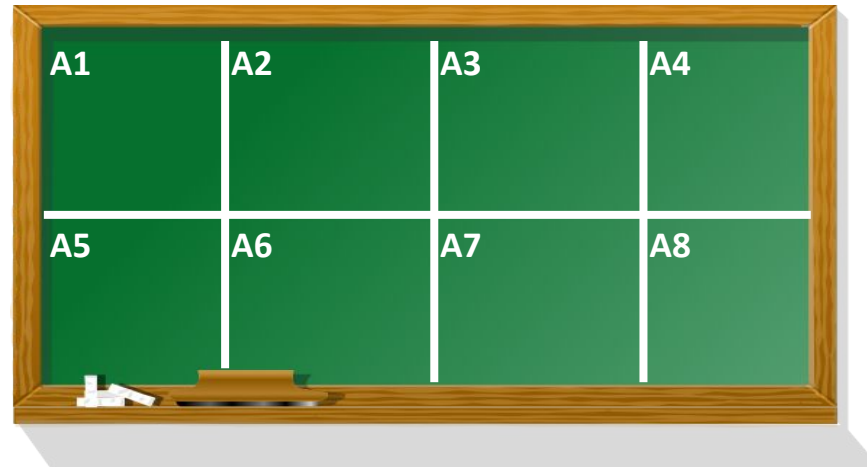
Ler 2 valores, representando os catetos de um triângulo retângulo e calcular e exibir o tamanho da hipotenusa:



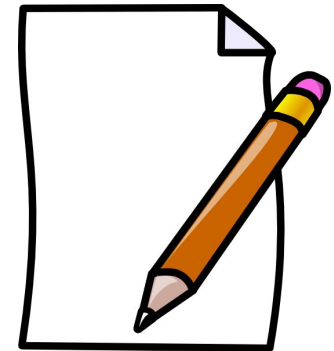
Cartões
Disponíveis



Cartões
Utilizados



Saída



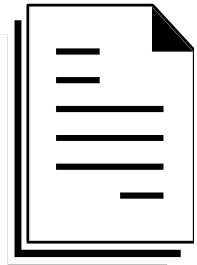
Lista de Instruções

Instrução	
P1	Leia um cartão e copie seu valor em A1
P2	...
P3	...
.	...
.	...
.	...

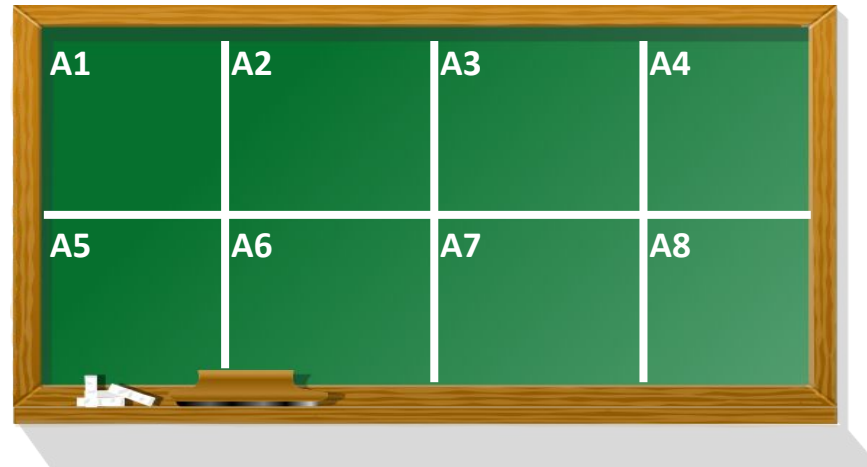
Ler 1 valor, representando uma temperatura em Fahrenheit, calcular e exibir o valor da temperatura convertido para Celsius:



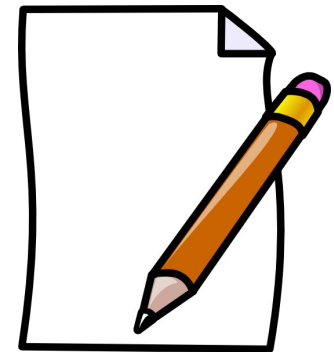
Cartões
Disponíveis



Cartões
Utilizados



Saída



Lista de Instruções

Instrução	
P1	Leia um cartão e copie seu valor em A1
P2	...
P3	...
.	...
.	...
.	...

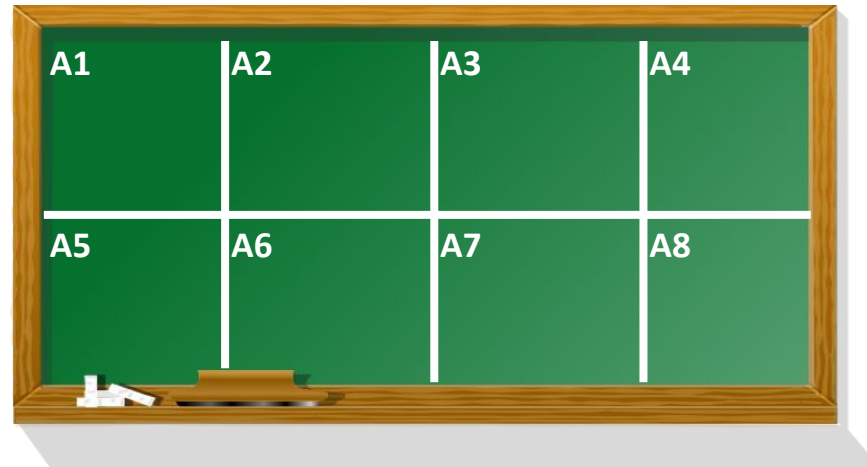
Ler 1 valor, representando a quantidade de chuva em polegadas, calcular e exibir o valor convertido para milímetros:



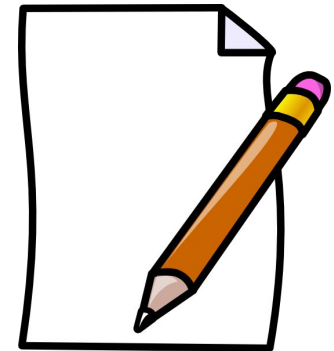
Cartões
Disponíveis



Cartões
Utilizados



Saída



Lista de Instruções

Instrução	
P1	Leia um cartão e copie seu valor em A1
P2	...
P3	...
.	...
.	...
.	...