FORMAÇÃO C

☐ **Fundamentos da linguagem**

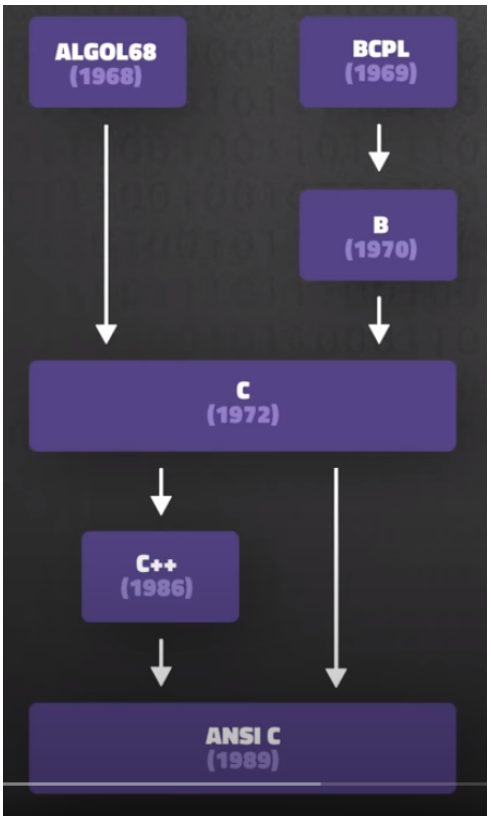
Artigo: Começando a programar com C

- Compilar: *gcc nome_arquivoC.c* | *gcc nome_arquivoC -o nome_arquivo.out* (.exe para windows)
- Executar: *./a.out* (em linux) | *a.exe* (em windows)

Vídeo: C (A Linguagem de Programação que é uma MÃE) // Dicionário do Programador

A linguagem C foi criada em **1972** por **Dennis Ritchie** (cientista da computação americano da Bell Labs).

- Direcionada para criação de compiladores e SOs
  - Foi a principal linguagem usada na criação do Unix (SO criado por Ritchie)
- Derivação das linguagens Algal68 e B.



- É de alto nível, mas provê recursos de baixo nível (permite incorporação de códigos Assembly)

- **Características**: Procedural, Portável, Compartimentalizada, de Uso geral e de Tipagem fraca (possível "desligar" o sistema de tipagem manipulando diretamente os bytes da memória)

- **Tipos de Dados**: primitivos e derivados
  int - 4 bytes | char - 1 byte | float - 4 bytes (número real com precisão de 6 casa decimais)| double - 8 bytes (número real com precisão de 15 casa decimais)

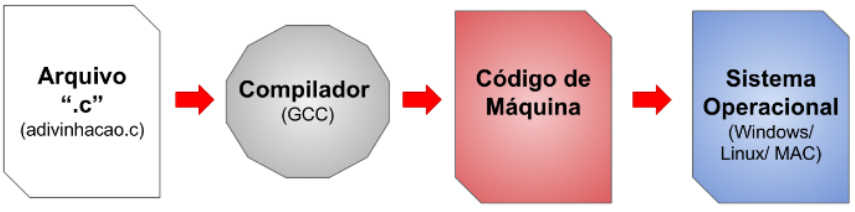
  *array - tamanho do tipo de arrray * número de posições*
  struct - soma dos tamanho de cada tipo definido na estrutura
  pointer - variável que armazena o endereço de outra variável (ponteiro)

- **Modificadores**:
  - short : reduz o tamanho
  - long : estende o tamanho
  - signed : valores positivos e negativos
  - unsigned - só valores positivos

- APIs: stdio, libc

Curso de C: conhecendo a Linguagem das Linguagens

- Compiladores: O GCC é um compilador de linguagem C. O compilador é o responsável por transformar todo o código que escrevemos em linguagem C para código de máquina (afinal, o computador não entende a linguagem C diretamente).



- Escopo de uma variável: são "os lugares onde podemos usá-la". Sempre que declaramos uma variável, podemos usá-la em qualquer lugar dentro das "chaves" que ela está, e nas "chaves" de dentro (por exemplo, se temos um if).
- Variáveis Numéricas:

Inteiros:

- `short` = 2 bytes
- `int` = 4 bytes = 32 bits ≅ 2^{32}
- `long` = 8 bytes = 64 bits ≅2^{64}

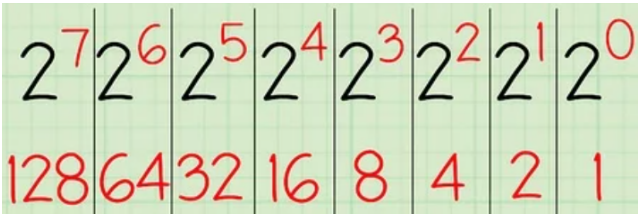
Ponto Flutuante:

- `float` = 4 bytes
- `double` = 8 bytes = 64 bits

- ascii art

Curso de C: avançando na linguagem

- Arrays: uma maneira de guardarmos "vários" de um só tipo. Esses elementos ficam armazenados na memória, um ao lado do outro. Um array/vetor é simplesmente um ponteiro que aponta para uma posição da memória que tem espaços vazios para guardar as N posições declaradas.
    - Memória de uma máquina: consegue apenas diferenciar apenas se algo está imantado para o valor 1 ou 0.
    - Base binária:



Como Converter Números Binários em Decimais
Como Converter de Decimal para Binário

    - O conjunto de 8 bits juntos é conhecido por 1 byte, e o maior número que poderia representar 2 elevado a 8 números, que é igual a 256.

- Um padrão famoso por ter substituído o ASCII como mais utilizado na internet em 2008, o UTF-8.
- Base Hexadecimal: números que usam até 16 algarismos para representar qualquer número (0 1 2 3 4 5 6 7 8 9 A B C D E F).
- Como representar letras (chars) em números binários? Basta fazermos uma tabela, dando um número para cada letra. Pronto, o computador é capaz de representar todas as letras existentes no nosso alfabeto usando somente 8 dígitos, 0 ou 1. Essa tabela do nosso alfabeto, incluindo diversos outros caracteres, é a tabela ASCII, usada por muito tempo como o principal padrão de tradução de números e caracteres por um computador.
- Como representar uma imagem? Mapeamos uma imagem para um conjunto de números inteiros. E já sabemos que conseguimos representar números inteiros de forma binária. Imagine então uma foto. Ela é composta por diversos pontos, e cada um desses pontos tem uma cor diferente. Imagine que a foto tem 1000 por 1000 pontos. O que podemos fazer é primeiro enumerar todas as cores existentes. Mas vamos limitá-las a 256 diferentes e não podemos usar o nome da cor (precisamos dar um número para ela).
  Com essa tabela em mãos, podemos escrever uma linha para cada ponto na foto. Com isso podemos representar qualquer imagem com tamanho finito (que caiba na memória) e que o número de cores esteja limitado a 256. Esse é o raciocínio atrás de um mapa de bits de cores, um bitmap, o formato BMP, muito utilizado antigamente no mundo da computação. Sua adoção não é mais tão grande devido ao tamanho que o mesmo ocupa: uma imagem com grande qualidade exige uma tabela de cores enorme, e um número de pontos maior ainda. Algoritmos de compressão como o JPEG dominam esse mercado.
  Assim como pensamos na alternativa de representar um terço como uma fração (um dividido por três), poderíamos representar uma imagem como uma sequência de traços, círculos etc. E essa também é uma ideia boa para diversos tipos de imagem, e é o raciocínio por trás das imagens vetoriais, como o SVG.
  O mesmo pode ser aplicado ao mundo da música, com o WAV e MP3, aos filmes etc. No fim, representamos desde um número inteiro até uma imagem de tomografia com apenas zeros ou uns.

- Ponteiros
- Funções

☐ **Recursos Avançados da linguagem**

- Matrizes: é um ponteiro que aponta para uma lista de arrays. Como arrays são ponteiros, então uma matriz é um ponteiro que aponta para uma lista de ponteiros.
- Ponteiros de ponteiros: é um ponteiro que aponta para outros ponteiros de inteiro, a declaração é int**, ou seja, duas estrelas para representar "ponteiro de ponteiro".
- Variáveis declaradas como globais são globais para todo o programa. Mas, para que um arquivo enxergue uma variável global definida em outro arquivo, precisamos fazer uso da palavra ***extern***.
- Funções de manipulação:
    - ***strcpy()*** para copiar um array para o outro
    - ***memcpy()*** para copiar o que está na memória, além de passar o destino e a origem, precisamos passar também a quantidade de bytes que queremos copiar.
    - ***sizeof()*** para calcular a quantidade de bytes
    - ***memset()*** para inicializar structs ou arrays com algum valor padrão (geralmente, "nulo").
- Ifdefs e Ifndefs: são ifs diferentes: eles acontecem em tempo de compilação.