

# SCC0221 – Introdução à Ciência de Computação I

Universidade de São Paulo  
Instituto de Ciências Matemáticas e de Computação  
Departamento de Ciências de Computação

Rodrigo Fernandes de Mello  
<http://www.icmc.usp.br/~mello>  
[mello@icmc.usp.br](mailto:mello@icmc.usp.br)

- Objetivo:
  - Apresentar os conceitos básicos para o desenvolvimento de programas, utilizando uma linguagem de programação como apoio

- Ementa:
  - Conceitos básicos sobre computadores: hardware, software, sistema operacional, compiladores, representação interna de dados e linguagem de programação
  - Resolução de problemas e desenvolvimento de algoritmos:
    - Análise e solução de problemas
    - Representação e documentação
    - Estruturas de programas: decisão e repetição.
    - Tipos de dados simples
    - Modularização de programas: procedimentos, funções e passagem de parâmetros.
    - Tipos de dados compostos: vetores, matrizes, cadeias de caracteres, registros, conjuntos e estruturas dinâmicas (ponteiros)
    - Arquivos
    - Depuração de programas
    - Programação em linguagem estruturada

- **Livro texto:**

- ASCENCIO, A. F. G.; CAMPOS, E. A. V. – Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++, Prentice Hall, 2003. (2002, 2005, 2008).

- **Bibliografia Complementar:**

- SCHILDT, HERBERT, C Completo e Total, Pearson, 2006. (2008).
- KELLEY, A. A book on C, Addison-Wesley, 2005 (1998).
- FORBELLONE, A. L. V.; EBERSPACHER, H. F. – Lógica de Programação, 2a. Edição, Makron Books, 2000. (2006)
- MANZANO, J.A. Estudo dirigido de algoritmos. Ed. Érica, 2008
- KERNIGHAM,B.; RITCHIE,D. The C Programming Language, Prentice-Hall, 1988.
- KERNIGHAM,B. A Prática de Programação, 2000.
- **VICTORINE VIVIANE MIZRAHI, Treinamento em Linguagem C – Módulo 1 e Módulo 2, Makron Books, 1990.**
- **VICTORINE VIVIANE MIZRAHI, Treinamento em Linguagem C, 2ª Ed., Makron Books, 2008.**

- **Outros:**

- Lopes, A.; Garcia, G.. Introdução à Programação: 500 Algoritmos Resolvidos, Editora Campus.

- Avaliação:
  - 3 Provas
  - Trabalhos
  - Utilização de média harmônica com ponderação

$$M = \frac{15 + k}{\frac{3}{P_1+1} + \frac{5}{P_2+1} + \frac{7}{P_3+1} + \sum_{i=1}^k \frac{1}{T_i+1}} - 1$$

- Material será disponibilizado no Tidia  
**<https://ae4.tidia-ae.usp.br/portal>**

- Com Professor:
  - Quintas-feiras
    - 15:30 hs às 17:30 hs
    - Sala 3-242
    - mello@icmc.usp.br

- Submissão eletrônica de trabalhos
  - Utilizando o sistema Run Codes:
    - <http://run.codes>
- Presenças:
  - Mínimo de 70%
  - Sobre Atestados
- Evitar atrasos
- Evitar conversas em excesso
- **Perguntem MUITO!**



Perguntas?

# Arquitetura de von Neumann

# Arquitetura de von Neumann

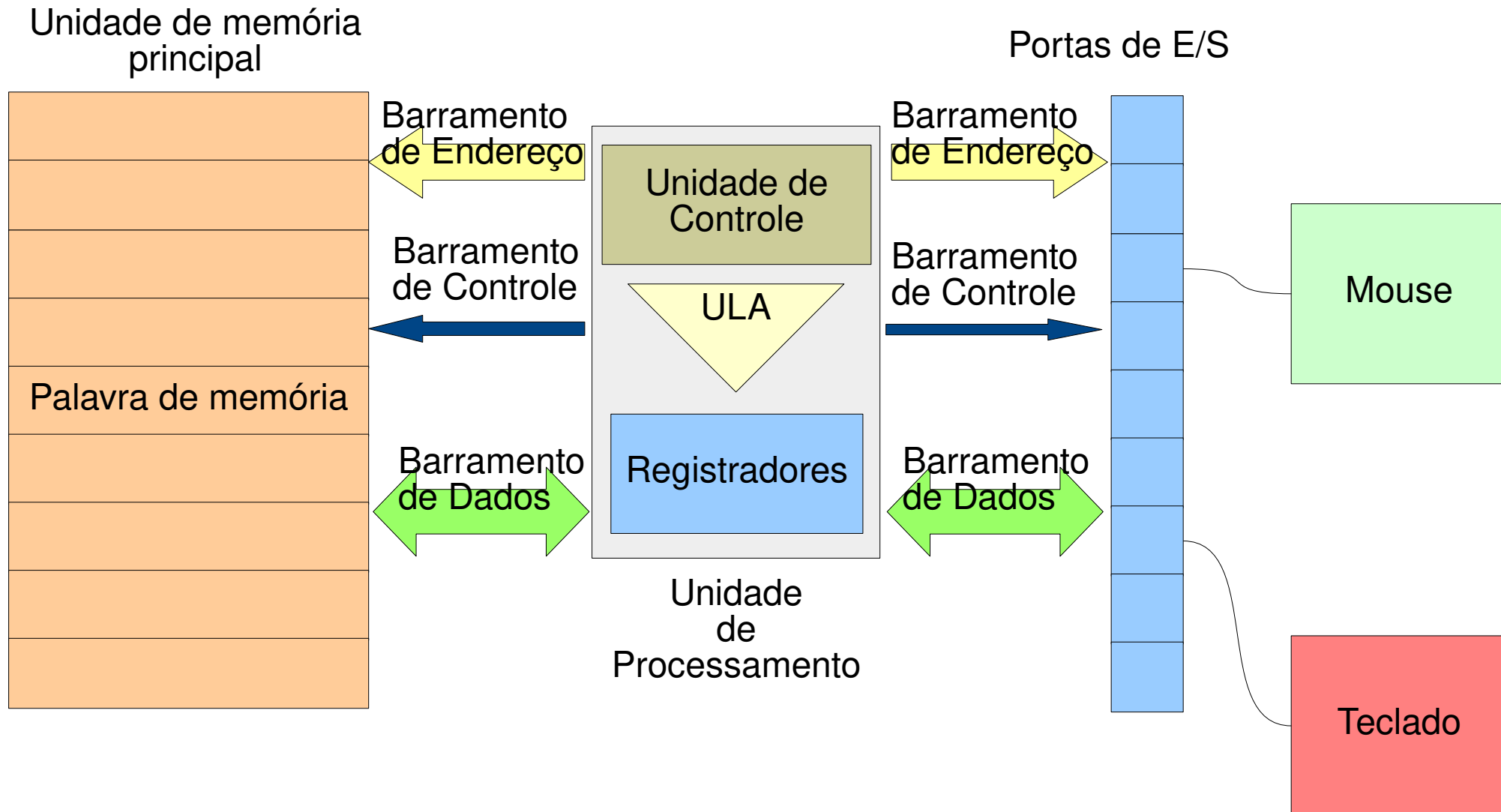
- Projetada por **John von Neumann**
  - Matemático Húngaro, naturalizado Americano
  - Contribuiu na teoria dos conjuntos, análise funcional, teoria ergódica, mecânica quântica, ciência da computação, economia, teoria dos jogos, análise numérica, hidrodinâmica das explosões, estatística e muitas outras as áreas da matemática.
  - Considerado um dos mais importantes matemáticos do século XX
- Arquitetura caracterizada pela possibilidade da máquina armazenar seus programas no mesmo espaço de memória em que armazena os dados

# Arquitetura de von Neumann

- Arquitetura de von Neumann contém:
  - Unidade de memória principal
  - Unidade de processamento
    - Unidade de controle
    - Unidade Lógica-Aritmética
    - Registradores
  - Portas de Entrada e Saída
  - Conexão com dispositivos

# Arquitetura de von Neumann

- Arquitetura de von Neumann



# Arquitetura de von Neumann

- Alguns conceitos necessários para estudar esta arquitetura:
  - O que é bit?
  - O que é byte?
  - Por que computadores utilizam unidade binária e não decimal ou octal?
  - Como converter decimal em binário?

# Arquitetura de von Neumann

- Computadores/Processadores são comumente caracterizados pelos barramentos de endereço e de controle:

<b>Processador</b>	<b>B. End.</b>	<b>B. Dados</b>	<b>Registradores</b>
8086	20 bits	8 bits	8 bits
80286	24 bits	16 bits	16 bits
80386SX	24 bits	16 bits	32 bits
80486	32 bits	32 bits	32 bits
Intel Core i7	46 bits	64 bits	64 bits

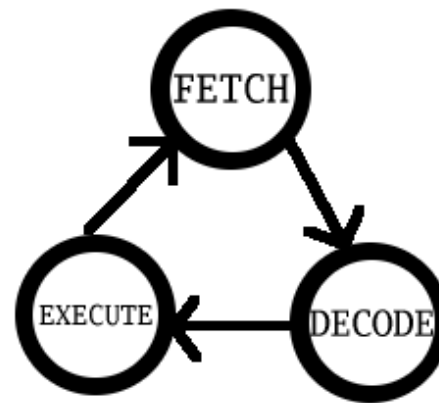
# Arquitetura de von Neumann

- Podemos:
  - **Em função do número de bits do barramento de endereço** → encontrar o número de palavras de memória que pode ser endereçadas
  - **Em função do número de bits do barramento de dados** → saber quantos bits podem trafegar do processador para a memória e vice-versa
  - **Em função do número de bits dos registradores** → saber qual o tamanho dos números que podemos formar
  - **Em função do número de bits do barramento de controle** → saber quantas operações podem ser definidas
    - Operações são definidas por conjuntos de bits
    - Em mais alto nível na forma de linguagem de montagem (Assembly)
    - Em mais alto nível ainda, usando uma linguagem como C



# Arquitetura de von Neumann

- Ciclo de von Neumann
  - Necessário pois operações ocorrem sempre dentro da CPU
  - Visa:
    - Buscar operações e ou dados
    - Decodificar operações
    - Executar operações



# Iniciando a Programação de Computadores

# Iniciando a Programação de Computadores

- Há diversas linguagens de programação que podem ser adotadas em um curso como este
  - Adotaremos a linguagem C
- Como começamos a programar em linguagem C?
  - Precisamos de um editor de texto e um compilador C
- **Começamos pela função main**
  - Define o início do programa
  - Vamos começar a programar...
    - Imprimir algo na tela para começarmos

# Iniciando a Programação de Computadores

- Como compilar/linkar um programa em linguagem C?
  - Usar gcc
- O que foi gerado pela compilação/linkagem?
  - Vejamos os arquivos binários
- Utilizando o comando nm para estudar o código binário gerado
- Podemos linkar de forma estática e dinâmica
  - O que isso significa?
  - Qual o impacto no tamanho dos arquivos e uso da memória?

# Iniciando a Programação de Computadores

- O que é compilação?
- O que é linkagem?
- Então a linkagem que é estática ou dinâmica? Sim.

## Tipos primitivos:

Variable Type	Keyword	Bytes Required	Range
Character	char	1	-128 to 127
Unsigned character	unsigned char	1	0 to 255
Integer	int	2	-32768 to 32767
Short Integer	short int	2	-32768 to 32767
Long Integer	long int	4	-2,147,483,648 to 2,147,438,647
Unsigned Integer	unsigned int	2	0 to 65535
Unsigned Short integer	unsigned short int	2	0 to 65535
Unsigned Long Integer	unsigned long int	4	0 to 4,294,967,295
Float	float	4	1.2E-38 to
Double	double	8	2.2E-308 to
Long Double	long double	10	3.4E-4932 to 1.1E+4932

- Vamos verificar seus tamanhos com sizeof e realizar operações simples de entrada e saída

## Funções básicas para entrada e saída:

- printf
- scanf
  - Oportunidade para discutir endereços da memória principal

Uso básico de ponteiros:

- Criar ponteiro para dois inteiros
- Operar sobre valores e sobre seus endereços
- Criar um inteiro e operar sobre ele com um ponteiro do tipo char



## Estruturas condicionais:

- IF

## Aplicações iniciais:

- Faça um programa que leia dois números e imprima o maior
- Faça um programa que receba um inteiro e verifique se ele é par ou ímpar
- Faça um programa que leia 3 números e os imprima em ordem crescente

## Mais Aplicações:

- Dada a hora inicial e final, calcule o tempo total de uma partida de futebol
- Faça um programa que receba um salário e calcule uma bonificação de  $x\%$
- Receba 3 reais que representam os comprimentos das faces de um triângulo. Calcule se eles formam de fato um triângulo e identifique seu tipo: equilátero, isósceles ou escaleno

**22.** Faça um programa que receba o salário base e o tempo de serviço de um funcionário. Calcule e mostre:

- O imposto, conforme a tabela a seguir.

SALÁRIO BASE	% SOBRE O SALÁRIO BASE
< R\$ 200,00	isento
Entre R\$ 200,00 (inclusive) e R\$ 450,00 (inclusive)	3%
Entre R\$ 450,00 e R\$ 700,00	8%
≥ R\$ 700,00	12%

- A gratificação, de acordo com a tabela a seguir.

SALÁRIO BASE	TEMPO DE SERVIÇO	GRATIFICAÇÃO
Superior a R\$ 500,00	Até 3 anos	20
	Mais de 3 anos	30
Até R\$ 500,00	Até 3 anos	23
	Entre 3 e 6 anos	35
	De 6 anos para cima	33

- O salário líquido, ou seja, salário base menos imposto mais gratificação.
- A categoria, que está na tabela a seguir.

SALÁRIO LÍQUIDO	CLASSIFICAÇÃO
Até R\$ 350,00	A
Entre R\$ 350,00 e R\$ 600,00	B
De R\$ 600,00 para cima	C

**24.** Faça um programa que receba o preço, o tipo (A — alimentação; L — limpeza; e V — vestuário) e a refrigeração (S — produto que necessita de refrigeração; e N — produto que não necessita de refrigeração) de um produto. Suponha que haverá apenas a digitação de dados válidos e, quando houver digitação de letras, utilize maiúsculas. Calcule e mostre:

- O valor adicional, de acordo com a tabela a seguir:

REFRIGERAÇÃO	TIPO	PREÇO	VALOR ADICIONAL
N	A	< R\$ 15,00	R\$ 2,00
		≥ R\$ 15,00	R\$ 5,00
	L	< R\$ 10,00	R\$ 1,50
		≥ R\$ 10,00	R\$ 2,50
	V	< R\$ 30,00	R\$ 3,00
		≥ R\$ 30,00	R\$ 2,50
S	A		R\$ 8,00
	L		R\$ 0,00
	V		R\$ 0,00

O valor do imposto, de acordo com a regra a seguir.

PREÇO	PERCENTUAL SOBRE O PREÇO
< R\$ 25,00	5%
≥ R\$ 25,00	8%

- O preço de custo, ou seja, preço mais imposto.
- O desconto, de acordo com a regra a seguir.

O produto que **não** preencher nenhum dos requisitos a seguir terá desconto de 3%, caso contrário, 0 (zero).

Os requisitos são:

**Tipo:** A

**Refrigeração:** S

- O novo preço, ou seja, preço de custo mais adicional menos desconto.
- A classificação, de acordo com a regra a seguir.

NOVO PREÇO	CLASSIFICAÇÃO
≤ R\$ 50,00	Barato
Entre R\$ 50,00 e R\$ 100,00	Normal
≥ R\$ 100,00	Caro

## Laços de repetição:

- FOR
- WHILE
- DO-WHILE

## Aplicações:

- Resolva o fatorial de um número
- Resolva a sequência de Fibonacci
- Dado um conjunto de  $n$  reais, calcule a média

- 
- 2.** Faça um programa que leia um valor N inteiro e positivo. Calcule e mostre o valor de E, conforme a fórmula a seguir:

$$E = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

- 
- 3.** Faça um programa que leia um número N que indica quantos valores inteiros e positivos devem ser lidos a seguir. Para cada número lido, mostre uma tabela contendo o valor lido e o fatorial desse valor.

- 
- 4.** Foi feita uma estatística em cinco cidades brasileiras para coletar dados sobre acidentes de trânsito. Foram obtidos os seguintes dados:

- a) código da cidade;
- b) número de veículos de passeio;
- c) número de acidentes de trânsito com vítimas.

Deseja-se saber:

- a) qual é o maior e qual é o menor índice de acidentes de trânsito e a que cidades pertencem;
- b) qual é a média de veículos nas cinco cidades juntas;
- c) qual é a média de acidentes de trânsito nas cidades com menos de 2.000 veículos de passeio.



**5.** Faça um programa que leia o número de termos e um valor positivo para X. Calcule e mostre o valor da série a seguir:

$$S = \frac{-X^2}{1!} + \frac{+X^3}{2!} - \frac{X^4}{3!} + \frac{+X^5}{4!} - \frac{X^6}{3!} + \frac{+X^7}{2!} - \frac{X^8}{1!} + \frac{+X^9}{2!} - \frac{X^{10}}{3!} + \frac{+X^{11}}{4!} - \dots$$

- 9.** Faça um programa que receba duas notas de seis alunos. Calcule e mostre:
- a média aritmética das duas notas de cada aluno; e
  - a mensagem que está na tabela a seguir:

Capít

MÉDIA ARITMÉTICA	MENSAGEM
Até 3	Reprovado
Entre 3 e 7	Exame
De 7 para cima	Aprovado

- o total de alunos aprovados;
- o total de alunos de exame;
- o total de alunos reprovados;
- a média da classe.

**13.** Foi feita uma pesquisa para determinar o índice de mortalidade infantil em certo período. Faça um programa que:

- leia o número de crianças nascidas no período;
- identifique o sexo (M ou F) e o tempo de vida de cada criança nascida.

O programa deve calcular e mostrar:

- a percentagem de crianças do sexo feminino mortas no período;
- a percentagem de crianças do sexo masculino mortas no período;
- a percentagem de crianças que viveram 24 meses ou menos no período.

**16.** Faça um programa para calcular a área de um triângulo e que não permita a entrada de dados inválidos, ou seja, medidas menores ou iguais a 0.

**21.** Faça um programa que receba vários números, calcule e mostre:

- a soma dos números digitados;
- a quantidade de números digitados;
- a média dos números digitados;
- o maior número digitado;
- o menor número digitado;
- a média dos números pares;
- a porcentagem dos números ímpares entre todos os números digitados.

Finalize a entrada de dados com a digitação do número 30.000.

## Arrays/Vetores na memória Stack:

- Vetores de inteiros e outros tipos
- Vetores de caracteres (Strings)

# Exercícios

- 1.** Faça um programa que preencha um vetor com nove números inteiros, calcule e mostre os números primos e suas respectivas posições.
- 
- 2.** Uma pequena loja de artesanato possui apenas um vendedor e comercializa dez tipos de objetos. O vendedor recebe, mensalmente, salário de R\$ 545,00, acrescido de 5% do valor total de suas vendas. O valor unitário dos objetos deve ser informado e armazenado em um vetor; a quantidade vendida de cada peça deve ficar em outro vetor, mas na mesma posição. Crie um programa que receba os preços e as quantidades vendidas, armazenando-os em seus respectivos vetores (ambos com tamanho dez). Depois, determine e mostre:
    - um relatório contendo: quantidade vendida, valor unitário e valor total de cada objeto. Ao final, deverão ser mostrados o valor geral das vendas e o valor da comissão que será paga ao vendedor; e
    - o valor do objeto mais vendido e sua posição no vetor (não se preocupe com empates).

# Exercícios

- 3.** Faça um programa que preencha dois vetores de dez elementos numéricos cada um e mostre o vetor resultante da intercalação deles.

Vetor 1	3	5	4	2	2	5	3	2	5	9
	1	2	4	4	5	6	7	8	9	10

Vetor 2	7	15	20	0	18	4	55	23	8	6
	1	2	4	4	5	6	7	8	9	10

Vetor resultante da intercalação

3	7	5	15	4	20	2	0	2	18	5	4	3	55	2	23	5	8	9	6
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

- 4.** Faça um programa que preencha um vetor com oito números inteiros, calcule e mostre dois vetores resultantes. O primeiro vetor resultante deve conter os números positivos e o segundo, os números negativos. Cada vetor resultante vai ter, no máximo, oito posições, que não poderão ser completamente utilizadas.

# Exercícios

- 6.** Faça um programa que preencha um vetor com dez números inteiros, calcule e mostre o vetor resultante de uma ordenação decrescente.

x	3	5	4	2	1	6	8	7	11	9
	1	2	3	4	5	6	7	8	9	10

Ordenado	11	9	8	7	6	5	4	3	2	1
	1	2	3	4	5	6	7	8	9	10

- 7.** Faça um programa que, no momento de preencher um vetor com oito números inteiros, já os armazene de forma crescente.

Mais sobre as memórias Stack e Heap

# Iniciando a Programação de Computadores

- Como um programa em linguagem C acessa a memória principal?
  - Memória Stack
  - Memória Heap
- Criando dois programas simples para estudar essas memórias:
  - Verificar o conteúdo do `/proc/<número>/maps`
- **Atenção:** não confundir linkagem ou ligação estática e dinâmica com alocação de memória estática e dinâmica. São assuntos diferentes!



# Iniciando a Programação de Computadores

- Como podemos usar a memória Stack e a Heap?
  - A linguagem C define tipos de dados para isso
- Tipos mais comuns:

Variable Type	Keyword	Bytes Required	Range
Character	char	1	-128 to 127
Unsigned character	unsigned char	1	0 to 255
Integer	int	2	-32768 to 32767
Short Integer	short int	2	-32768 to 32767
Long Integer	long int	4	-2,147,483,648 to 2,147,438,647
Unsigned Integer	unsigned int	2	0 to 65535
Unsigned Short integer	unsigned short int	2	0 to 65535
Unsigned Long Integer	unsigned long int	4	0 to 4,294,967,295
Float	float	4	1.2E-38 to
Double	double	8	2.2E-308 to
Long Double	long double	10	3.4E-4932 to 1.1E+4932

- Há ainda um tipo chamado de ponteiro cujo tamanho depende da arquitetura do computador

# Iniciando a Programação de Computadores

- Estudemos alguns tipos utilizando programas em linguagem C
  - Para isso iremos criar variáveis
    - Regiões da memória principal em que armazenamos dados
  - Como um inteiro (int) é armazenado na memória de um computador de arquitetura Intel?
    - Seria ele armazenado de forma distinta em outra arquitetura?
    - O que é uma arquitetura?
  - Podemos utilizar a linguagem C para verificar como esse inteiro foi armazenado na memória?
    - Fazemos um programa para isso

# Iniciando a Programação de Computadores

- Estudemos alguns tipos utilizando programas em linguagem C
  - O que ocorre se criamos várias variáveis na memória utilizando diferentes tipos?
    - Podemos ver em qual local da memória cada uma dessas variáveis está? Como fazemos isso?
  - O que é um ponteiro?
    - Vejamos seu tamanho
    - Se podemos verificar seu tamanho, podemos verificar os tamanhos dos demais tipos?
    - Vejamos como um ponteiro aponta para uma região da memória

# Iniciando a Programação de Computadores

- Quais as operações mais comuns que podemos fazer sobre:
  - Inteiros: int, short, long
  - Pontos flutuantes: float, double
- Como operar sobre inteiros usando bitwise operators
  - `>>`, `<<`, `&` e `|`
- Resumir máscaras utilizadas para imprimir tipos com `printf`

# Iniciando a Programação de Computadores

- Exercício
  - Como um float e um double são armazenados na memória de um computador Intel?
  - O que ocorreria se você percorresse esse float utilizando um ponteiro para char?

## Arrays/Vetores na memória Heap:

- Refaçamos alguns dos exercícios anteriores

## Matrizes na memória Stack e na Heap

- Matrizes de inteiros e reais
- Matrizes de caracteres

# Exercícios

- 1.** Faça um programa que preencha uma matriz  $M$  ( $2 \times 2$ ), calcule e mostre a matriz  $R$ , resultante da multiplicação dos elementos de  $M$  pelo seu maior elemento.
- 2.** Faça um programa que preencha uma matriz  $10 \times 3$  com as notas de dez alunos em três provas. O programa deverá mostrar um relatório com o número dos alunos (número da linha) e a prova em que cada aluno obteve menor nota. Ao final do relatório, deverá mostrar quantos alunos tiveram menor nota em cada uma das provas: na prova 1, na prova 2 e na prova 3.

**ALGORITMO** Solução:

- 4.** Crie um programa que preencha uma matriz  $10 \times 20$  com números inteiros e some cada uma das linhas, armazenando o resultado das somas em um vetor. A seguir, o programa deverá multiplicar cada elemento da matriz pela soma da linha correspondente e mostrar a matriz resultante.

**ALGORITMO** Solução:

- 6.** Crie um programa que preencha uma primeira matriz de ordem  $4 \times 5$  e uma segunda matriz  $5 \times 2$ . O programa deverá, também, calcular e mostrar a matriz resultante do produto matricial das duas matrizes anteriores, armazenando-o em uma terceira matriz de ordem  $4 \times 2$ .

**ALGORITMO** Solução:



# Exercícios

**24.** Uma escola deseja controlar as médias das disciplinas que seus alunos cursam. Sabe-se que nessa escola existem três turmas, com oito alunos cada uma, e cada aluno cursa quatro disciplinas. Crie um programa que armazene essas médias em uma matriz  $3 \times 8 \times 4$ . Depois da leitura, ele deverá calcular e mostrar:

- a média geral de cada aluno;
- a média de cada turma.

**ALGORITMO** Solução:

# Procedimentos e Funções

- Incluir passagem de parâmetros por valor e por referência

**1.** Faça um programa contendo uma sub-rotina que retorne 1 se o número digitado for positivo ou 0 se for negativo.

**2.** Faça um programa contendo uma sub-rotina que receba dois números positivos por parâmetro e retorne a soma dos  $N$  números inteiros existentes entre eles.

**3.** Faça um programa contendo uma sub-rotina que receba três números inteiros  $a$ ,  $b$  e  $c$ , sendo  $a$  maior que 1. A sub-rotina deverá somar todos os inteiros entre  $b$  e  $c$  que sejam divisíveis por  $a$  (inclusive  $b$  e  $c$ ) e retornar o resultado para ser impresso.

**9.** Faça uma sub-rotina que leia cinco valores inteiros, determine e mostre o maior e o menor deles.

**10.** Crie uma sub-rotina que receba como parâmetro um valor inteiro e positivo  $N$  e retorne o valor de  $S$ , obtido pelo seguinte cálculo:

$$S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

# Procedimentos e Funções

- Incluir passagem de parâmetros por valor e por referência

---

**13.** Faça uma sub-rotina que receba um vetor A de dez elementos inteiros como parâmetro. Ao final dessa função, deverá ter sido gerado um vetor B contendo o fatorial de cada elemento de A. O vetor B deverá ser mostrado no programa principal.

---

- Implementar uma função de ordenação de números inteiros e outra para busca sequencial e binária

## Manipulação de cadeias de caracteres

- Nas memórias Stack e Heap
- Inicializar cadeias de caracteres
- Copiar caracteres
- Concatenar cadeias de caracteres
- Comparação de cadeias
- Tamanho de uma cadeia de caracteres
- Descobrir a posição de uma cadeia dentro de outra

## Manipulação de cadeias de caracteres

- Nas memórias Stack e Heap
- Alterar caracteres
- Obter o código ASCII de um caracter
- Conversão para maiúsculas e minúsculas
- Converter um inteiro para caracter e vice-versa

# Exercícios

- 
- 1.** Faça um programa que receba uma frase, calcule e mostre a quantidade de vogais da frase digitada. O programa deverá contar vogais maiúsculas e minúsculas.

SOLUÇÃO:

- Digitar uma frase.
- Pegar o tamanho da frase.
- Percorrer a frase, pegando caractere por caractere.
- Comparar cada caractere com as vogais (maiúsculas e minúsculas).
- Quando encontrar uma vogal, acrescentar um na quantidade.

- 
- 2.** Faça um programa que receba uma frase, calcule e mostre a quantidade de consoantes da frase digitada. O programa deverá contar consoantes maiúsculas e minúsculas.

SOLUÇÃO:

- 
- 3.** Faça um programa que receba uma frase, calcule e mostre a quantidade de palavras da frase digitada.

SOLUÇÃO:

- Digitar uma frase.
- Pegar o tamanho da frase.
- Percorrer a frase, pegando caractere por caractere.
- Comparar cada caractere com espaço em branco.
- Quando encontrar um espaço, acrescentar um na quantidade.
- Como após a última palavra não tem espaço, acrescentar um na quantidade.

# Exercícios

- 5.** Faça um programa para criptografar uma frase dada pelo usuário (a criptografia troca as vogais da frase por \*).

Exemplo:

Frase: EU ESTOU NA ESCOLA

Saída: \*\* \* ST\*\* N\* \*SC\*L\*

- 7.** Faça um programa que receba uma frase e coloque as palavras da frase em ordem alfabética.

Exemplo:

Entrada: A informática está em constante evolução

Saída: A constante em está evolução informática

SOLUÇÃO:

- Digitar uma frase.
- Pegar o tamanho da frase digitada.
- Percorrer a frase, pegando caractere por caractere.
- Cada vez que encontrar um espaço em branco ou o fim da frase, foi obtida uma palavra completa.
- Se a nova frase (onde as palavras ficarão em ordem alfabética) estiver vazia, copiar nela a palavra obtida.
- Caso contrário, percorrer a nova frase até encontrar a posição adequada para colocar a palavra extraída da frase digitada.
- Caso a palavra extraída da frase digitada seja alfabeticamente maior que todas as palavras que já estão armazenadas na nova frase, concatenar essa palavra ao final da nova frase.

# Exercícios

- 9.** Faça um programa que receba uma frase e uma palavra, calcule e mostre a quantidade de vezes que a palavra digitada aparece na frase.

Exemplo:

Frase: EU ESTOU NA ESCOLA E A ESCOLA É LEGAL.

Palavra: ESCOLA

Resposta: A palavra ESCOLA apareceu 2 vezes na frase.

SOLUÇÃO:

- Digitar uma frase.
- Pegar o tamanho da frase digitada.
- Percorrer a frase, pegando caractere por caractere.
- Cada vez que encontrar um espaço em branco ou o fim da frase, foi obtida uma palavra completa.
- Se a palavra completa for igual à palavra digitada, acrescentar um no contador.



# Exercícios

**16.** Faça um programa que receba uma palavra e verifique se ela constitui um palíndromo, ou seja, se a palavra escrita do fim para o começo fica igual à palavra escrita do começo para o fim.

Exemplos:

ARARA

ANA

MIRIM

SOLUÇÃO:

- Digitar uma palavra.
- Gerar uma nova palavra, que será a palavra digitada invertida.
- Comparar a palavra digitada com a palavra invertida.
- Se as palavras forem iguais, trata-se de palíndromo; caso contrário, não se trata de palíndromo.

# Exercícios

- 23.** Faça um programa para criptografar uma frase em que cada caractere deverá ser substituído pelo caractere que está três posições à sua frente na tabela ASCII. Os três últimos caracteres da tabela ASCII deverão ser substituídos pelos três primeiros.

Exemplo:

BONECO ZABUMBA  
ERQHFR CDEXPED

SOLUÇÃO:

- Digitar uma frase.
- Pegar o tamanho da frase.
- Percorrer toda a frase, convertendo cada caractere para ASCII, acrescentando três e convertendo novamente para caractere.
- Copiar o novo caractere para a frase criptografada.

## Registros ou Structs

- Declaração
- Acesso aos campos

# Exercícios:

---

- 1.** Faça um programa que realize o cadastro de contas bancárias com as seguintes informações: número da conta, nome do cliente e saldo. O banco permitirá o cadastramento de apenas 15 contas e não poderá haver mais que uma conta com o mesmo número. Crie o menu de opções a seguir.

Menu de opções:

1. Cadastrar contas.
2. Visualizar todas as contas de determinado cliente.
3. Excluir a conta com menor saldo (supondo a não existência de saldos iguais).
4. Sair.

# Exercícios:

**3.** Faça um programa que utilize os registros a seguir.

CLIENTES	DOCUMENTOS
cod_cli	num_doc
nome	cod_cli
fone	data_venc
endereco	data_pag
	valor
	juros

Sabe-se que um documento só pode ser cadastrado para um cliente que já exista. Considere que podem existir, no máximo, 15 clientes e 30 documentos. Crie um vetor para clientes e outro para documentos. Crie um menu para a realização de cada uma das operações especificadas a seguir.

- Cadastrar clientes* — não pode existir mais que um cliente com o mesmo código.
- Cadastrar documentos* — ao cadastrar um documento, se a data de pagamento for maior que a data de vencimento, calcular o campo 'juros' do registro documentos (5% sobre o valor original do documento).
- Excluir clientes* — um cliente só poderá ser excluído se não existir nenhum documento associado a ele.
- Excluir documentos individuais* — por meio de seu número. Caso o documento não exista, o programa deverá mostrar a mensagem *Documento não encontrado*.
- Excluir documentos por cliente* — o programa deverá informar o código do cliente e excluir todos os seus documentos. Caso o cliente não exista, deverá mostrar a mensagem *Cliente não encontrado*.
- Excluir documentos por período* — o programa deverá informar a data inicial e a data final e excluir todos os documentos que possuam data de vencimento nesse período.
- Alterar as informações sobre os clientes* — só não pode ser alterado o código do cliente.
- Mostrar o total de documentos de determinado cliente*.
- Sair.

Quando forem excluídos clientes ou documentos, os vetores deverão ser reorganizados, ou seja, todas as posições não preenchidas dos vetores deverão ficar no final. Exemplo: se for necessário excluir o número 8 do vetor a seguir, tanto o 9 quanto o 1 deverão ser movidos uma casa para a esquerda e a última posição deverá ficar livre para uma nova inclusão.

**Vetor inicial**

12	5	8	9	1
----	---	---	---	---

**Vetor modificado com uma  
posição livre ao final**

12	5	9	1	
----	---	---	---	--

- 4.** Faça um programa que efetue reserva de passagens aéreas de determinada companhia. O programa deverá ler os números dos aviões e o número de lugares disponíveis em cada um. Utilize um vetor de quatro posições, no qual cada posição representa um avião, e outro vetor também de quatro posições para armazenar os lugares disponíveis.

O programa deverá mostrar o seguinte menu de opções:

1. Cadastrar o número dos aviões.
2. Cadastrar o número de lugares disponíveis em cada avião.
3. Reservar passagem.
4. Consultar por avião.
5. Consultar por passageiro.
6. Finalizar.

Imagine que poderão ser registradas até 60 reservas e que cada uma deverá possuir o número do avião e o nome do passageiro.

**Para realizar a opção 1,** deverá ser solicitado ao usuário o número dos quatro aviões disponíveis.

**Para realizar a opção 2,** deverá ser solicitado ao usuário o número de lugares disponíveis em cada avião cadastrado na opção 1.

**Para realizar a opção 3,** deverá ser verificado se o número do avião digitado é válido. Posteriormente, checar se, no avião escolhido, ainda existe lugar disponível. Caso exista, o programa deverá diminuir o total de vagas e mostrar a mensagem *Reserva confirmada*. Caso contrário, deverá mostrar a mensagem *Voo lotado*. Observe que não podem ser feitas mais de 60 reservas.

**Para realizar a opção 4,** deverá ser solicitado o número do avião desejado e, posteriormente, deverão ser exibidas todas as suas reservas.

**Para realizar a opção 5,** deverá ser solicitado o nome do passageiro e, posteriormente, deverão ser exibidas todas as reservas feitas em seu nome.

**A opção 6** encerra o programa.

# Exercícios:

- 7.** Crie um programa para ler o código, o sexo (M — masculino; F — feminino) e o número de horas-aula ministradas pelos professores de uma escola durante um mês. Sabe-se que um professor ganha R\$ 60,50 hora-aula e que a escola possui dez professores. Após a leitura, o programa deverá mostrar:
- a) Uma listagem contendo o código, o salário bruto, o desconto e o salário líquido de todos o professores.
  - b) A média aritmética dos salários brutos dos professores do sexo masculino.
  - c) A média aritmética dos salários brutos dos professores do sexo feminino.
- Os descontos devem ser assim calculados:

Sexo	Até 70 horas/aula ao mês	Mais que 70 horas/aula ao mês
Masculino	10%	8%
Feminino	7%	5%

## Arquivos:

- Abrir
  - Apagar se existente
  - Ler se existente
  - Ler e escrever se existente
- Funções de leitura e escrita
- Movimentar o ponteiro de posicionamento dentro de um arquivo
- Teste de fim de arquivo



# Exercícios:

- 1.** Faça um programa para criar um arquivo chamado `ALUNOS.DAT`, no qual cada registro será composto pelos seguintes campos: `numero`, `nome`, `curso`, `nota1`, `nota2`.

**ALGORITMO** Solução:

- 3.** Faça um programa para alterar as notas dos alunos do arquivo criado no Exercício 1.

**ALGORITMO** Solução:

- 5.** Faça um programa para excluir os alunos do arquivo criado no Exercício 1.

- Ler um arquivo binário de áudio raw e imprimir seus valores
- Criptografar arquivos de entrada

Algumas outras ideias de programas

# Iniciando a Programação de Computadores

- **Faça um programa que leia  $N$  inteiros, faça a soma de todos e calcule a média desses termos**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Faça um programa que leia um conjunto de caracteres com vírgulas e extraia os tokens (subconjunto de caracteres) separados por vírgulas**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- Vejamos mais sobre as principais funções da linguagem C e sobre como consultar o manual dessas funções:
  - strlen
  - strcat
  - strcpy
  - strcmp
  - memcpy
  - strtok
  - malloc, free e realloc

# Iniciando a Programação de Computadores

- **Faça um programa que realize o escalonamento de uma matriz a fim de resolver um sistema linear**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Utilizando seu programa que realiza o escalonamento de uma matriz, implemente um sistema de criptografia de dados**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Considere um arquivo de entrada que define uma imagem. Construa um sistema que aplica um filtro de convolução sobre essa imagem de entrada**
  - **Testar para diferentes máscaras**
  - **Qual a utilidade?**
- **Compilar:**
  - Com ligação dinâmica
  - Com ligação estática
  - Comparar o tamanho dos binários gerados
    - Relembrar o que ocorre com o uso da memória
- Poderíamos programar este mesmo programa de outras formas?



# Iniciando a Programação de Computadores

- **Crie um programa que receba um vetor de inteiros e implemente o algoritmo K-médias (<http://pt.wikipedia.org/wiki/K-means>)**
  - **Em quais cenários pode ser utilizado?**
- **Compilar:**
  - Com ligação dinâmica
  - Com ligação estática
  - Comparar o tamanho dos binários gerados
    - Relembrar o que ocorre com o uso da memória
- Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Crie um programa que leia um arquivo binário de entrada que corresponde a um arquivo de áudio. Em seguida processe esse arquivo usando a Transformada de Fourier para encontrar seus coeficientes complexos. Utilize isso para eliminar algumas frequências e compactar o áudio.**
  - **Em quais cenários pode ser utilizado?**
- **Compilar:**
  - Com ligação dinâmica
  - Com ligação estática
  - Comparar o tamanho dos binários gerados
    - Relembrar o que ocorre com o uso da memória
- Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- Sobre funções:
  - Criar exemplos de programas com passagem de parâmetro por valor e por referência
  - Vejamos mais sobre como a Stack recebe parâmetros que são passados
  - Vejamos os endereços dos parâmetros na Stack
  - Podemos fazer uma função que retorne qualquer tipo de dado?

# Iniciando a Programação de Computadores

- **Crie uma função para ler o conteúdo de um arquivo texto e retorná-lo**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Crie uma função para criptografar o texto lido no exercício anterior. Crie uma função que restaure os dados. E, finalmente, crie outra função que grave os resultados em um arquivo de saída textual**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- Outros assuntos:
  - union
  - typedef
  - enum
  - structs

# Iniciando a Programação de Computadores

- **Crie uma função que leia um registro (struct) de um arquivo binário. Crie outra para gravar um registro em arquivo binário**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- Mais funções sobre arquivos:
  - fopen
  - fclose
  - fwrite e fread
  - fprintf e fscanf
  - remove



# Iniciando a Programação de Computadores

- **Implemente um programa que calcule o fatorial de um inteiro positivo usando o paradigma iterativo e, em seguida, o recursivo**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?

# Iniciando a Programação de Computadores

- **Estude o problema da Torre de Hanói e o implemente usando recursão**
  - Compilar:
    - Com ligação dinâmica
    - Com ligação estática
    - Comparar o tamanho dos binários gerados
      - Relembrar o que ocorre com o uso da memória
  - Poderíamos programar este mesmo programa de outras formas?