INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ
CAMPUS PICOS

ESTRUTURA DE DADOS

MAÍLA DE LIMA CLARO

claromaila@gmail.com

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

INTRODUÇÃO AO C: COMANDOS DE SAÍDA

- Para a programação em C nesta disciplina estarei utilizando o site replit.com.

```c
/* Meu primeiro programa: primeiro.c
 * Programa que imprime uma mensagem na tela
 */

#include <stdio.h>

// Função principal. Inicia a execução do programa
int main()
{
	printf("Hello world!\n");
	return 0; // indica que o programa terminou com sucesso
}
```

- Saída do programa:

```
Hello world!
```

ENTENDENDO O PROGRAMA

- O programa primeiro.c, escrito em C, é composto pelos seguintes componentes:
- Comentários: São ignorados pelo compilador e servem para auxiliar o programador a descrever o programa.
  - Uma linha: // Comentário
  - Uma ou várias linhas: /* Comentário */
- Todos os comandos terminam com um ";" (ponto-e-vírgula).
- As chaves "{ e "}" indicam, respectivamente, o início e o fim de um bloco de comandos.
- Os comandos da linguagem são palavras reservadas (também chamados de palavras-chave) e são escritos em letras minúsculas.

ENTENDENDO O PROGRAMA

- A diretiva de compilação #include <stdio.h>, informa ao compilador que ele deve incluir a biblioteca stdio (Standard Input/Output) durante o processo de compilação.
- int main(): declara a função principal main() que retorna um valor do tipo int (número inteiro).
- A função printf imprime o texto na saída padrão (tela).
- O comando "return" finaliza o programa retornando o valor 0.

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

VARIÁVEIS

- Não é possível fazer programas de computador úteis sem utilizar alguma porção de memória.
- **Variáveis:**
  - são locações na memória do computador onde pode-se armazenar um valor;
  - são utilizadas para armazenar e manipular dados.
- **Declaração de variáveis**:
  - em um programa C, uma variável envolve um tipo e um identificador: tipo identificador;
    - Exemplos: int number; double real; etc.

DECLARAÇÃO DE VARIÁVEIS

- Toda variável tem **tipo**, **nome**, **endereço de memória** e **valor**.
- Uma variável deve ser declarada com um identificador e um tipo de dado antes de ser usada no programa.
- Se já existir um valor armazenado na variável e um novo valor for atribuído a esta variável, esse valor **sobrescreve** o valor anterior.
- Exemplo: int number;
  - O tipo int especifica que o valor armazenado é do tipo inteiro (valor inteiro).
  - O identificador number é o nome da variável.
- Pode-se declarar várias variáveis em uma mesma linha:
  - int number1, number2, number3, number4;

DECLARAÇÃO DE VARIÁVEIS

- **Tipos fundamentais**
  - int – armazena um número inteiro.
  - double – especifica os números reais; 3.4, -0.985, etc.
  - char – armazena um único caractere minúsculo ou maiúsculo, um dígito, ou um caractere especial ($ * @).
- Em C, os tipos fundamentais são **palavras reservadas** escritas em **letras minúsculas**.

DECLARAÇÃO DE VARIÁVEIS

- **Identificador**
  - é o nome da variável, e **não pode ser uma palavra-chave**.
  - é formado por uma combinação de letras, dígitos e " _ " sublinhado (underline), **começando sempre com uma letra** ou " _ ".
  - **case sensitive**: letras maiúsculas e minúsculas **são diferentes**.
  - para assegurar a portabilidade use no máximo **31 caracteres**.
  - escolha **nomes significativos** para facilitar a documentação e o entendimento do código.

DECLARAÇÃO DE VARIÁVEIS

Alguns erros...

```
int var1, 2var, _var3;
```

```
int var1, 2var, _var3;
```

```
int x, y, z;
int double;
```

- Porque os códigos acima geram erros?

DECLARAÇÃO DE VARIÁVEIS

- **Onde declarar?**
  - Variáveis podem ser declaradas em qualquer lugar de um programa C/C++, mas **devem aparecer antes** de serem usadas no programa.

Exemplo 1	**Exemplo 2**

int x;
x = 80;
print("%d", x);
int y;
y = 60;
print("%d", y);
	int x;
	int y;
	x = 80;
	y = 60;
	print("%d", x);
	print("%d", y);

DECLARAÇÃO DE VARIÁVEIS

- As variáveis podem ocupar tamanhos diferentes na memória, dependendo do tipo, exemplo:

Tipo	Bytes	Intervalo
char	1	0 a 255
short	2	-32.768 a 32.767
int	4	-2.147.483.648 a 2.147.483.647
long	4	-2.147.483.648 a 2.147.483.647
float	4	$1,2 \times 10^{-38}$ a $3,4 \times 10^{+38}$
double	8	$2,2 \times 10^{-308}$ a $1,8 \times 10^{+308}$

MEMÓRIA

- A memória é formada por várias células.
- Cada célula contém um endereço e um valor

(veja exemplo ao lado).

- O tamanho do endereço e do valor dependem

da arquitetura (32/64 bits).

Endereço	Valor
00010000	??
00010001	??
00010002	??
00010003	??
00010004	??
00010005	??
00010006	??
00010007	??
00010008	??
00010009	??
0001000A	??
0001000B	??

MEMÓRIA

- Exemplo:
- O caractere char i ocupa 1 byte na memória

Endereço	Valor	
00010000	??	i
00010001	??	
00010002	??	
00010003	??	
00010004	??	
00010005	??	
00010006	??	
00010007	??	
00010008	??	
00010009	??	
0001000A	??	
0001000B	??	

MEMÓRIA

- Exemplo:
- O inteiro int i ocupa 4 bytes na memória

(considerando uma arquitetura de 32 bits)

Endereço	Valor
00010000	??
00010001	
00010002	
00010003	
00010004	??
00010005	??
00010006	??
00010007	??
00010008	??
00010009	??
0001000A	??
0001000B	??

i

MEMÓRIA

- Exemplo:
- O ponto flutuante float i ocupa 4 bytes na memória

(considerando uma arquitetura de 32 bits)

Endereço	Valor
00010000	??
00010001	
00010002	
00010003	
00010004	??
00010005	??
00010006	??
00010007	??
00010008	??
00010009	??
0001000A	??
0001000B	??

i

MEMÓRIA

- Exemplo:
- double i ocupa 8 bytes na memória

(considerando uma arquitetura de 32 bits)

Endereço	Valor
00010000	??
00010001	
00010002	
00010003	
00010004	
00010005	
00010006	
00010007	
00010008	??
00010009	??
0001000A	??
0001000B	??

i

ENDEREÇOS

- Ao declararmos uma variável x, ela será associada a:
  - Um nome (exemplo: x)
  - Um endereço de memória ou referência (exemplo: 0xbfd267c4)
  - Um valor (exemplo: 9)

```
int x = 9;
```

- Para acessar o endereço de uma variável, utilizamos o operador &

OPERADOR DE ATRIBUIÇÃO

- sum = number1 + number2;
  - O símbolo '=' é um operador de atribuição.
  - Avalia-se a expressão matemática do lado direito do '=' e atribui-se o resultado à variável do lado esquerdo.
  - = e + são operadores binários; requerem dois operandos.

- **Dica**: coloque espaços em branco em ambos os lados de um operador binário para facilitar a leitura do programa.

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

OPERADORES ARITMÉTICOS

Operação	Operador Aritmético	Exemplo	Exemplo em C/C++
Adição	+	f + 7	f + 7
Subtração	-	p – c	p – c
Multiplicação	*	bm ou B x m	b * m
Divisão	/	x/y ou x ÷ y ou $\frac{x}{y}$	x / y
Módulo	%	r mod s	r % s

- Observações:
- Operador módulo %: resulta no resto da divisão inteira (somente usado com operandos inteiros)
- Exemplo: 7 % 4 é igual a 3

OPERADORES ARITMÉTICOS

- **Regras da Precedência de Operadores**
- São as mesma da álgebra:
- Operadores entre parênteses são avaliados primeiro; note que o parênteses quebra a precedência de um operador.
- A seguir, aplicam-se as operações de **multiplicação**, **divisão** e **módulo**. Se uma expressão contém vários desses operadores, as operações são aplicadas da esquerda para a direita.
- Por último aplicam-se a **adição** e a **subtração**. Se há vários + e -, a aplicação ocorre da esquerda para a direita.

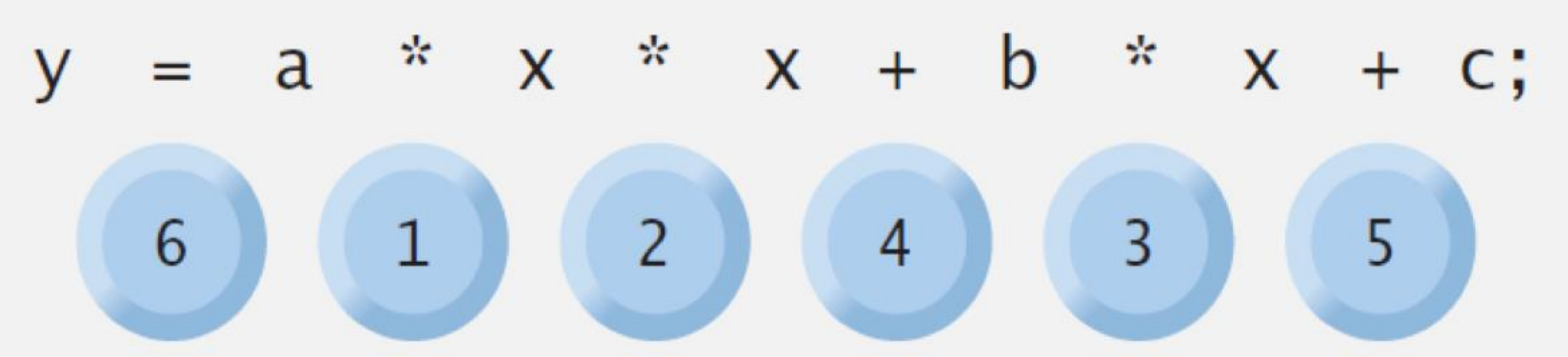
OPERADORES ARITMÉTICOS

- **Regras da Precedência de Operadores**

Operação	Operador	Ordem de avaliação
Parênteses	( )	Avaliados primeiro (pares mais internos avaliados antes)
Multiplicação Divisão Módulo	* / %	Avaliados em segundo lugar Se houver vários, avaliação da esquerda para direita
Adição Subtração	+ -	Avaliados por último. Se houver vários, avaliação da esquerda para direita.

OPERADORES ARITMÉTICOS

- **Ordem de avaliação das expressões aritméticas**



AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

A FUNÇÃO PRINT

- A função printf é parte da biblioteca <stdio.h>:
  - Utilizada para imprimir na tela.
  - Exemplo de uso:

```
printf("Olá mundo!!!\n");
```

A FUNÇÃO PRINT

- Mas.. Como imprimir um número inteiro?

Alguns erros...

```
printf(10);
```

```
int valor = 10;
printf(valor);
```

- Os códigos acima produzirão um **erro**, pois printf deve receber um texto/formato (entre aspas), não um inteiro (seja valor ou variável).

A FUNÇÃO PRINT

- Uso de printf:
  - printf(formato, valor/variável);
- Exemplo:

```
printf("%d", 10);
```

- (note que "%d" é usado para números inteiros)

A FUNÇÃO PRINT

- Alguns possíveis formatos para o comando printf:
  - "%d": int (número inteiro)
  - "%ld": long long (número inteiro)
  - "%f": float (ponto flutuante)
  - "%lf": double (ponto flutuante)
  - "%c": char (caractere)
  - "%s": string (cadeia de caracteres)

A FUNÇÃO PRINT

- Assim, para imprimir um número inteiro usamos o formato "%d" como texto e indicamos o inteiro como próximo argumento.
- Exemplos:

```
printf("%d", 100); // imprime o número inteiro 10
```

```
int number = 10;
printf("%d", number); // imprime o valor da variável number
```

```
int n1 = 10;
int n2 = 20;
int soma = n1 + n2;
printf("%d\n", soma); // imprime o valor de soma e a quebra de linha
```

A FUNÇÃO PRINT

- Note que é possível mesclar formato com texto, como por exemplo em "O resultado é %d“

```
int n1 = 10;
int n2 = 20;
int soma = n1 + n2;
printf(“A soma de %d e %d é igual a %d.\n”, n1, n2, soma);
```

- Naturalmente, o código acima produzirá a saída:

```
A soma de 10 e 20 é igual a 30.
```

A FUNÇÃO PRINT

- Outro exemplo:

```
double n1 = 10.8;
double n2 = 19.3;
double soma = n1 + n2;
printf("%lf + %lf = %lf\n", n1, n2, soma);
```

- O código acima produzirá a saída:

```
10.8 + 19.3 = 30.1
```

- Note que o caractere ponto (.) é usado para separar os decimais,

FORMATANDO A SAÍDA

- A função printf permite formatar a saída de dados. O usuário pode especificar, entre outros:
  - número de casas decimais;
  - número de caracteres ocupados pela impressão.
- Exemplos:
  - "%3d": um int usando no mínimo 3 espaços
  - "%-3d": um int usando no mínimo 3 espaços (alinhado à esquerda)
  - "%5s": uma string usando no mínimo 5 espaços
  - "%.3f": um float usando 3 casas decimais
  - "%3f": um float usando no mínimo 3 espaços
  - "%5.3f": um float usando 3 casas decimais e no mínimo 5 espaços

FORMATANDO A SAÍDA

- Exemplo:

```
printf("%-3s %8s\n","Var","Val");
printf("%-3s %8.1f\n","x",  10.222);
printf("%-3s %8.1f\n","y", 20.33);
printf("%-3s %8.1f\n","z", 30);
```

- Imprimirá na saída:

```
Var          Val
x             10.2
y             20.3
z             30.0
```

FORMATANDO A SAÍDA

- Caracteres especiais:
  - \n: quebra de linha, ou seja, passa para a linha debaixo;
  - \t: tabulação horizontal, equivalente a um **tab**;
  - \": aspas duplas;
  - \': aspas simples ou apóstrofo;
  - \\: barra invertida
  - \a: ???**beep** ;)

FORMATANDO A SAÍDA

- Exemplo de impressão de tabela:

```
printf( "Var \t Val\n");
printf("x  \t 10\n");
printf("y  \t 20\n");
printf("z  \t 30\n");
```

- Resultado:

```
Var         Val
x           10
y           20
z           30
```

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

A FUNÇÃO SCANF

- A função scanf também é parte da biblioteca <stdio.h>:
- Utilizada para ler da entrada padrão (terminal).
- O scanf tem algumas (grandes) diferenças em relação ao printf:
- A função printf imprime texto e o **valor** de variáveis.
- A função scanf **altera o conteúdo** das variáveis.
- Alterar conteúdo equivale a **modificar o que está na memória**.

A FUNÇÃO SCANF

- Uso de scanf:
  - scanf(formato, endereços de memória);
- Exemplo:

```
int x;
scanf("%d", &x);
```

- (note que "%d" é usado para números inteiros)

A FUNÇÃO SCANF

- A função scanf usa os mesmos "formatos" que printf.
- Exemplos:
  - "%d": int (número inteiro)
  - "%ld": long long (número inteiro)
  - "%f": float (ponto flutuante)
  - "%lf": double (ponto flutuante)
  - "%c": char (caractere)
  - "%s": string (cadeia de caracteres)

A FUNÇÃO SCANF

- Porquê os códigos abaixo geram **erros**?

Alguns erros...

```
int x;
scanf(x);
```

```
double valor = 10.0;
scanf(valor);
```

- scanf deve receber um texto/formato (entre aspas), não um int ou double (seja valor ou variável).
- scanf deve receber um endereço de memória, e não um valor.

A FUNÇÃO SCANF

- E os códigos a seguir? Também geram **erros**?

Alguns erros...

```
int x;
scanf("%d", x);
```

```
double valor = 10.0;
scanf("%lf", valor);
```

- Sim: scanf deve receber **endereços de memória**, não valores.

A FUNÇÃO SCANF

- Assim, para ler da entrada padrão usamos um "formato" e indicamos o endereço de memória como próximo argumento.
- Exemplos:

```
int x;
scanf("%d", &x); // lê um inteiro da entrada padrão
```

```
char c;
scanf("%c", &c); // lê um caractere da entrada padrão
```

```
int n1, n2, soma;
scanf("%d %d", &n1, &n2); // lê dois inteiros da entrada padrão
soma = n1 +n2;
printf("A soma de %d e %d eh igual a %d", n1, n2, soma);
```

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

EXEMPLOS

Exemplo 1

Elabore um programa em C que lê o valor das variáveis x, y e z do tipo int. Em seguida, calcule o resultado da expressão a seguir:

$$r = x^3 + y^2 + xyz$$

Imprima o resultado no formato do exemplo abaixo:

```
x = 10, y = 10, z = 10
r = 2100
```

EXERCÍCIOS

Exemplo 1

Elabore um programa em C que imprime o resto da divisão de 100000 por 13.

Exemplo 2

Elabore um programa em C que lê o valor das variáveis a, b, c, d, e do tipo float e, em seguida, calcula o resultado da expressão a seguir:

$$x = a^3 * (\frac{b+c}{d} + e)$$

Imprima o resultado no formato do exemplo abaixo:

```
a = 10, b = 10, c = 10, d = 10, e = 10
r = 12000
```

Exemplo 3

Elabore um programa que lê o valor de x1, x2 e x3 para calcular (e imprimir) o resultado da expressão $y = (x1 + 3)^4 + (x2 * x3)^3$

AULA DE HOJE

1. Primeiro programa
2. Variáveis
3. Operadores aritméticos
4. Comandos de saída
5. Comandos de entrada
6. Exemplos e Exercícios
7. Próxima aula

PRÓXIMA AULA

- Continuação da programação em C.

CONTATO:

Maíla de Lima Claro

(claromaila@gmail.com)