

LINGUAGEM C – UMA INTRODUÇÃO

AULA 5 – Expressões e Operadores

1 – Precedência dos operadores aritméticos

Na Aula 3 do nosso curso foram apresentados os Operadores Aritméticos (binários, unários, de incremento e decremento e operadores de atribuições). A primeira parte dessa nossa Aula 5 é apresentar maneiras de combinarmos esses operadores para formarmos expressões matemáticas ou expressões lógicas.

Apenas como revisão, segue uma lista com os operadores já estudados até esse ponto de nosso curso:

Operador binário	Descrição
=	Atribuição
+	Soma
-	Subtração
*	Multiplificação
/	Divisão
%	Resto da divisão

Operador unário	Descrição
-	Sinal negativo
+	Sinal positivo

Operador	Descrição
++	Incrementa 1
--	Decrementa 1

Instrução normal	Instrução reduzida
var = var + expr	var += expr
var = var - expr	var -= expr
var = var * expr	var *= expr
var = var / expr	var /= expr

Até este ponto, você deve ser capaz de entender expressões do tipo:

```
int nota1 = 7;
int nota2 = 8;
int soma = nota1 + nota2;
//O valor de soma é 15
```

```
float nota1 = 7.3;
float nota2 = 8.4;
float soma;
float media;
soma = nota1 + nota2;
media = soma/2;
```

```
//O valor de média é 7.85
```

```
int contador=1;  
contador++;  
//O valor de contador é 2
```

```
int contador=1;  
contador += 5;  
contador = -contador;  
//O valor de contador é -6
```

Agora começaremos a analisar expressões um pouco mais complicadas, mais próximas de problemas reais implementados computacionalmente. Vejamos alguns exemplos:

```
float nota1 = 7.3;  
float nota2 = 8.4;  
float media;  
media = nota1 + nota2 / 2; //Expressão 1
```

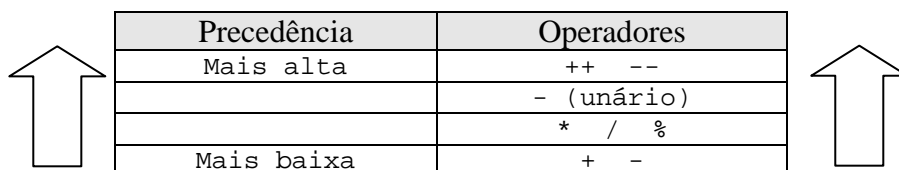
Pergunta 1: Qual o valor da variável *media*?

```
float nota1 = 7.3;  
float nota2 = 8.4;  
float media;  
media = ++nota1 + nota2 / 2; //Expressão 2
```

Pergunta 2: Qual o valor da variável *media*?

1.2 – Tabela de precedência

A seguinte tabela resume a ordem de precedência dos operadores em Linguagem C (e em praticamente todas as linguagens de programação):



Precedência	Operadores
Mais alta	++ --
	- (unário)
	* / %
Mais baixa	+ -

Com base na tabela acima, vamos analisar as duas expressões propostas anteriormente:

Expressão 1: `media = nota1 + nota2 / 2;`

O operador divisão (/) possui uma precedência mais alta que o operador adição (+), portanto a primeira operação que ocorre é a divisão de *nota2* por 2. O resultado dessa divisão é somado ao valor de *nota1* e finalmente o valor dessa soma é atribuído à variável *media* através do operador de atribuição (=).

Logo, a resposta da “Pergunta 1” é: $7.3 + 4.2 = 11.5$

Expressão 2: `media = ++nota1 + nota2 / 2;`

O operador de incremento (++) possui precedência sobre os operadores de divisão e soma, então a primeira operação realizada é o incremento da variável *nota1* (ela passa a valer 8.3). Agora o operador divisão (/) possui uma precedência mais alta

que o operador adição (+), portanto a próxima operação que ocorre é a divisão de *nota2* por 2. O resultado dessa divisão é somado ao valor incrementado de *nota1* e finalmente o valor dessa soma é atribuído à variável *media* através do operador de atribuição (=).

Logo, a resposta da “Pergunta 4” é: $8.3 + 4.2 = 12.5$

1.3 – Parênteses

Os parênteses são utilizados para alterarmos a ordem de precedência dos operadores. Assim, o cálculo correto da média poderia ser feito da seguinte forma:

```
float nota1 = 7.3;
float nota2 = 8.4;
float media;
media = ( nota1 + nota2 ) / 2; //Expressão 3
```

Pergunta 3: Qual o valor da variável *media*?

Na expressão 3, primeiro será analisado o que está entre parênteses, ou seja, primeiro *nota1* será somado à *nota2*. Essa soma então será dividida por 2 e o resultado da divisão será atribuído à variável *média*.

Logo, a resposta da “Pergunta 3” é: $(15.7)/2 = 7.85$

Devemos lembrar que dentro dos parênteses, a tabela de precedência continua valendo, veja o exemplo abaixo:

```
float nota1 = 7.3;
float nota2 = 8.4;
float media;
media = ( ++nota1 + nota2 ) / 2; //Expressão 4
```

Pergunta 4: Qual o valor da variável *media*?

Na expressão 4, primeiro será analisado o que está entre parênteses, ou seja, $++nota1 + nota2$. Só que temos dois operadores dentro desses parênteses, então de acordo com a tabela de precedência, primeiro *nota1* será incrementada e depois esse resultado será somado à variável *nota2*. Essa soma então será dividida por 2 e o resultado da divisão será atribuído à variável *média*.

Logo, a resposta da “Pergunta 4” é: $(8.3 + 8.4)/2 = (16.7)/2 = 8.35$

Estamos prontos para resolver o seguinte problema: A média de um aluno é calculada da seguinte forma:

Existem duas provas, uma mensal e uma bimestral, cada uma valendo de 0 a 10.

Existe um trabalho valendo de 0 a 10.

80% da média correspondem à média de provas

20% da média correspondem à nota de trabalho

Problema: Implemente um programa em Linguagem C que calcule a média de um aluno de acordo com o critério de avaliação acima proposto.

Um algoritmo para resolver o problema proposto poderia ser:

- Receba a nota mensal
- Receba a nota bimestral

- Receba a nota de trabalhos
- Calcule a média de provas com a fórmula $mediaProvas = (mensal + bimestral) / 2$
- Calcule a média do aluno com a fórmula $media = 0.8 * mediaProvas + 0.2 * trabalhos$
- Exiba a média do aluno a tela

Implementando em Linguagem C:

Exemplo 30

```

/*
Exemplo de cálculo de média - Precedência de Operadores
Linguagem C - Uma introdução
--
Diego M. Rodrigues
*/
#include <stdio.h>

int main() {
    //Declarando as variáveis
    float mensal=0, bimestral=0, trabalhos=0;
    float mediaProvas=0, media=0;

    //Recebendo a nota mensal
    printf("Digite o nota mensal (ex:8.7):\n");
    scanf("%f", &mensal);

    //Recebendo a nota bimestral
    printf("\nDigite a nota bimestral (ex:7.9):\n");
    scanf("%f", &bimestral);

    //Recebendo a nota bimestral
    printf("\nDigite a nota de trabalhos (ex:9.2):\n");
    scanf("%f", &trabalhos);

    //Calculando a média
    mediaProvas = (mensal + bimestral)/2;
    media=0.8*mediaProvas + 0.2*trabalhos;

    //Mostrando a média na tela
    printf("\nMedia: %4.1f", media);

    //Esperando o <ENTER> para sair com getch() ao invés de getchar()
    getch();

    return 0;
}

```

Exemplo de execução do programa anterior:

```

C:\Documents and Settings\Diego\Meus documentos\ExemplosC\exemplo30.exe
Digite o nota mensal (ex:8.7):
8.5
Digite a nota bimestral (ex:7.9):
8
Digite a nota de trabalhos (ex:9.2):
9.1
Media: 8.4_

```

Poderíamos deixar o programa anterior um pouco mais elegante, calculando a média do aluno em uma única linha, sem usarmos a variável auxiliar `mediaProvas`. O novo algoritmo para resolver o problema proposto poderia ser:

- Receba a nota mensal
- Receba a nota bimestral
- Receba a nota de trabalhos
- Calcule a média fórmula $media = 0.8 * ((mensal + bimestral) / 2) + 0.2 * trabalhos$
- Exiba a média do aluno a tela

Implementando em Linguagem C:

```
Exemplo 31
/*
Exemplo de cálculo de média - Precedência de Operadores
Linguagem C - Uma introdução
--
Diego M. Rodrigues
*/
#include <stdio.h>

int main() {
    //Declarando as variáveis
    float mensal=0, bimestral=0, trabalhos=0;
    float media=0;

    //Recebendo a nota mensal
    printf("Digite o nota mensal (ex:8.7):\n");
    scanf("%f", &mensal);

    //Recebendo a nota bimestral
    printf("\nDigite a nota bimestral (ex:7.9):\n");
    scanf("%f", &bimestral);

    //Recebendo a nota bimestral
    printf("\nDigite a nota de trabalhos (ex:9.2):\n");
    scanf("%f", &trabalhos);

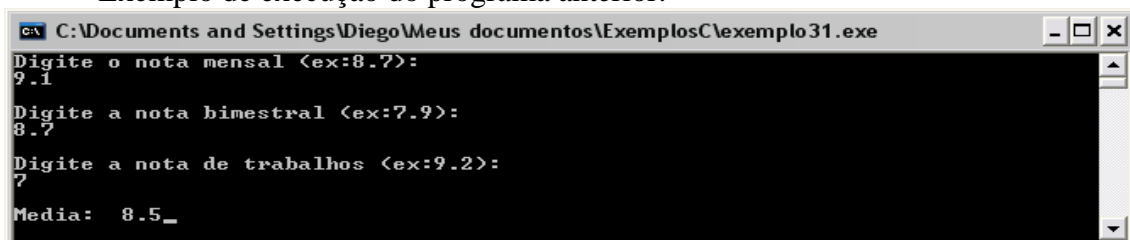
    //Calculando a média
    media=0.8*((mensal + bimestral)/2) + 0.2*trabalhos;

    //Mostrando a média na tela
    printf("\nMedia: %4.1f", media);

    //Esperando o <ENTER> para sair com getch() ao invés de getch()
    getch();

    return 0;
}
```

Exemplo de execução do programa anterior:



Problema: Dado um número n maior do que 10, calcule o dobro do último algarismo desse número. Exemplos:

$$72 \Rightarrow 2 * 2 = 4$$

$$257 \Rightarrow 2 * 7 = 14$$

$$1095 \Rightarrow 2 * 5 = 10$$

Um algoritmo para resolver o problema proposto poderia ser:

- Receba o número n
- Calcule o resto da divisão de n por 10
- Multiplique o resto por 2
- Exiba o resultado na tela

```
Exemplo 32
/*
Exemplo de Precedência de Operadores
Linguagem C - Uma introdução
--
Diego M. Rodrigues
*/
#include <stdio.h>

int main() {
    //Declarando as variáveis
    int n=0;
    int resto=0;

    //Recebendo a nota mensal
    printf("Digite um número inteiro maior que 10:\n");
    scanf("%d", &n);

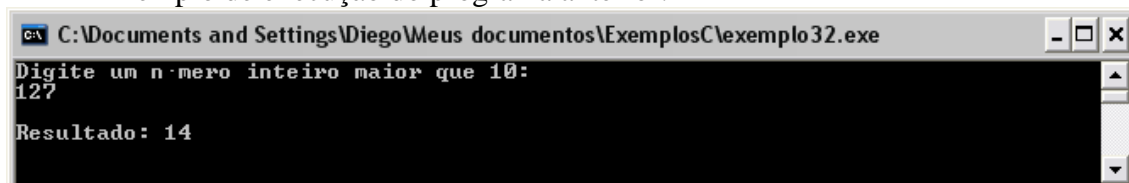
    //Calculando a média
    resto = n%10;
    resto = resto*2;

    //Mostrando a média na tela
    printf("\nResultado: %d", resto);

    //Esperando o <ENTER> para sair com getch() ao invés de getchar()
    getch();

    return 0;
}
```

Exemplo de execução do programa anterior:



Podemos agrupar as linhas...

```
resto = n%10;  
resto = resto*2;
```

...de várias maneiras, por exemplo:

```
resto = (n%10)*2;
```

Na linha acima, como o operador resto (%) possui a mesma precedência do operador multiplicação (*), é recomendado colocarmos os parênteses para garantir que o cálculo seja realmente feito na ordem que desejamos.

2. Operadores Relacionais

Os operadores relacionais são usados para determinar o relacionamento de uma quantidade com outra. Eles sempre retornam **0 (falso)** ou **1 (verdadeiro)**, dependendo do resultado do teste.

Alguns programadores preferem usar as palavras “true” que vem do inglês (verdadeiro) e “false” (do inglês falso) para representar os valores 0 ou 1 em resultados de testes lógicos.

Tabela de Operadores relacionais:

Operador relacional	Descrição
>	Maior que
>=	Maior ou igual que
<	Menor que
<=	Menor ou igual que
==	Igual
!=	Diferente (não igual)

O programa seguinte ilustra o uso de cada operação e exibe o resultado de cada uma como 0 ou 1:

Exemplo 33

```
/*  
Exemplo de Operadores Relacionais  
Linguagem C - Uma introdução  
--  
Diego M. Rodrigues  
*/  
#include <stdio.h>  
  
int main() {  
    //Recebendo num1 e num2  

```

```

printf(">= resultado: %d \n", num1 >= num2);
printf("> resultado: %d \n", num1 > num2);
printf("< resultado: %d \n", num1 < num2);

//Esperando o <ENTER> para sair com getch() ao invés de getch()
getch();
return 0;
}

```

Exemplos de execução do programa anterior:

```

C:\Documents and Settings\Diego\Meus documentos\Exemplos\exemplo33.exe
Informe um numero: 5
Informe outro numero: 9

Aplicando operadores em 5 e 9
== resultado: 0
!= resultado: 1
<= resultado: 1
>= resultado: 0
> resultado: 0
< resultado: 1

```

Tela A

```

C:\Documents and Settings\Diego\Meus documentos\Exemplos\exemplo33.exe
Informe um numero: 8
Informe outro numero: 8

Aplicando operadores em 8 e 8
== resultado: 1
!= resultado: 0
<= resultado: 1
>= resultado: 1
> resultado: 0
< resultado: 0

```

Tela B

```

C:\Documents and Settings\Diego\Meus documentos\Exemplos\exemplo33.exe
Informe um numero: 3
Informe outro numero: 1

Aplicando operadores em 3 e 1
== resultado: 0
!= resultado: 1
<= resultado: 0
>= resultado: 1
> resultado: 1
< resultado: 0

```

Tela C

Comentários sobre os resultados anteriores:

- Como já era esperado, todos os resultados das expressões com operadores relacionais foram 0 ou 1, que equivalem à falso ou verdadeiro (false ou true).
- Na **Tela A** entramos com os valores 5 e 9, então:
 - No teste de igualdade com o operador == (5==9) o resultado foi 0 (falso).

- o No teste de diferença com o operador $!=$ ($5!=9$) o resultado foi 1 (verdadeiro).
 - o No teste “menor ou igual que” com o operador $<=$ ($5<=9$) o resultado foi 1 (verdadeiro, já que 5 é menor do que 9).
 - o No teste “maior ou igual que” com o operador $>=$ ($5>=9$) o resultado foi 0 (falso, já que 5 não é ‘maior ou igual que’ 9).
 - o No teste “maior que” com o operador $>$ ($5>9$) o resultado foi 0 (falso, já que 5 não é ‘maior que’ 9).
 - o No teste “menor que” com o operador $<$ ($5<9$) o resultado foi 1 (verdadeiro, já que 5 é ‘menor que’ 9).
- Na **Tela B** entramos com os valores 8 e 8, então:
 - o No teste de igualdade com o operador $==$ ($8==8$) o resultado foi 1 (verdadeiro).
 - o No teste de diferença com o operador $!=$ ($8!=8$) o resultado foi 0 (falso).
 - o No teste “menor ou igual que” com o operador $<=$ ($8<=8$) o resultado foi 1 (verdadeiro, já que 8 é igual que 8).
 - o No teste “maior ou igual que” com o operador $>=$ ($8>=8$) o resultado foi 1 (verdadeiro, já que 8 é igual que 8).
 - o No teste “maior que” com o operador $>$ ($8>8$) o resultado foi 0 (falso, já que 8 não é ‘maior que’ 8).
 - o No teste “menor que” com o operador $<$ ($8<8$) o resultado foi 0 (falso, já que 8 não é ‘menor que’ 8).
- Na **Tela C** entramos com os valores 3 e 1, então:
 - o No teste de igualdade com o operador $==$ ($3==1$) o resultado foi 0 (falso).
 - o No teste de diferença com o operador $!=$ ($3!=1$) o resultado foi 1 (verdadeiro).
 - o No teste “menor ou igual que” com o operador $<=$ ($3<=1$) o resultado foi 0 (falso, já que 3 não é ‘menor ou igual que’ 8).
 - o No teste “maior ou igual que” com o operador $>=$ ($3>=1$) o resultado foi 1 (verdadeiro, já que 3 é maior que 1).
 - o No teste “maior que” com o operador $>$ ($3>1$) o resultado foi 1 (verdadeiro, já que 3 é maior que 8).
 - o No teste “menor que” com o operador $<$ ($3<1$) o resultado foi 0 (falso, já que 3 não é ‘menor que’ 1).

ATENÇÃO

O operador $=$ é usado para atribuição e o operador $==$ é usado para comparação!

Para testarmos se a variável `opcao` possui o valor 3, devemos usar `opcao==3`, nunca `opcao=3` !!