

Aritmética e Álgebra

Conceitos e Exemplos em Linguagem C

Aritmética

A aritmética é o ramo mais elementar e antigo da matemática, lida com as operações possíveis entre os números. As quatro operações matemáticas mais elementares (adição, subtração, multiplicação e divisão) são a aritmética elementar.

Álgebra

Álgebra é o ramo da Matemática que generaliza a aritmética. Isso significa que os conceitos e operações provenientes da aritmética (adição, subtração, multiplicação, divisão etc.) serão testados e sua eficácia será comprovada para todos os números pertencentes a determinados conjuntos numéricos.

Na álgebra incógnitas são introduzidas com o uso de letras. Em programação um análogo é feito com o uso de variáveis.

Operações Básicas

- Declaração de Variáveis úteis para problemas matemáticos

Os dois tipos de variáveis mais utilizados para resolução de problemas matemáticos mais simples são o 'int' para valores inteiros e o 'float' para valores reais.

Exemplo:

```
int t = 0; //variável de nome t inicializada com o valor 0
```

```
float M = 5.0; //variável de nome M inicializada com o valor 5.0
```

Operações Básicas

- Impressão de valores 'int' e 'float'

Usa-se o especificador de formato %d para 'int' e %f para 'float'.

Exemplo:

```
int x = 10;  
float y = 20;  
printf("%d", x);  
  
printf("%d %f", x, y);
```

Operações Básicas

- Leitura de valores 'int' e 'float'

Os mesmos especificadores de formato %d para 'int' e %f para 'float' caso queira ler dados da entrada padrão.

Exemplo:

```
int x, z;  
float y;  
//Leitura de um valor inteiro  
scanf("%d", &x);  
//Leitura de um valor real  
scanf("%f", &y);
```

Operações Básicas

- Operações Aritméticas

São as quatro operações elementares. Há também um operador que devolve o resto de uma divisão inteira. Divisão, multiplicação e resto têm precedência, pode se usar parênteses para alterar a precedência.

Operador	Significado	Exemplo
+	Adição de dois valores	$z = x + y$
-	Subtração de dois valores	$z = x - y$
*	Multiplicação de dois valores	$z = x * y$
/	Quociente de dois valores	$z = x / y$
%	Resto de uma divisão	$z = x \% y$

Operações Básicas

- Operações Aritméticas

Em uma operação utilizando o operador de quociente /, se o numerador e o denominador forem números inteiros, por padrão o compilador retornará apenas a parte inteira da divisão.

```
int main() {  
    float x;  
    x = 5/4; // x = 1.000000  
    printf("x = %f\n", x);  
  
    x = 5/4.0; // x = 1.250000  
    printf("x = %f\n", x);  
  
    return 0;  
}
```


Alguns Problemas de Aritmética e Álgebra

Mínimo Múltiplo Comum

É um tipo de operação matemática utilizada para encontrar o menor número positivo, diferente de 0, que é múltiplo ao mesmo tempo de dois ou mais números.

Os múltiplos de qualquer número podem ser obtidos através da multiplicação dele pelos números naturais.

Observe o exemplo:

$3 \times 0 = 0$; $3 \times 1 = 3$; $3 \times 2 = 6$; $3 \times 3 = 9$; $3 \times 4 = 12$; $3 \times 5 = 15$; $3 \times 6 = 18$...

$5 \times 0 = 0$; $5 \times 1 = 5$; $5 \times 3 = 15$; $5 \times 4 = 20$; $5 \times 5 = 25$; $5 \times 6 = 30$; $5 \times 7 = 35$...

Em $M(3) = \{0, 3, 6, 9, 12, 15, 18, \dots\}$ e $M(5) = \{0, 5, 10, 15, 20, 25, 30, 35, \dots\}$, o menor número da interseção desses dois conjuntos é 15, logo ele é MMC.

Mínimo Múltiplo Comum

Exemplo em C:

```
1  #include <stdio.h>
2
3  int main() {
4
5      int mmc, aux, i, num1 = 3, num2 = 5;
6
7      for (i = 2; i <= num2; i++) {
8          aux = num1 * i;
9          if ((aux % num2) == 0) {
10             mmc = aux;
11             i = num2 + 1;
12         }
13     }
14     printf("%d\n", mmc);
15 }
```

15

Máximo Divisor Comum

Divisor são todos os números naturais que dividem outro número natural de forma exata, isto é, sem resto. Todo número natural têm pelo menos dois divisores (1 e ele mesmo) e caso só tenha esses dois, é chamado de número primo. O MDC é o maior número natural que é divisor de ao mesmo tempo de dois ou mais números.

Exemplo:

Divisores de 30: 1,2,3,5,6,10,15 e 30.

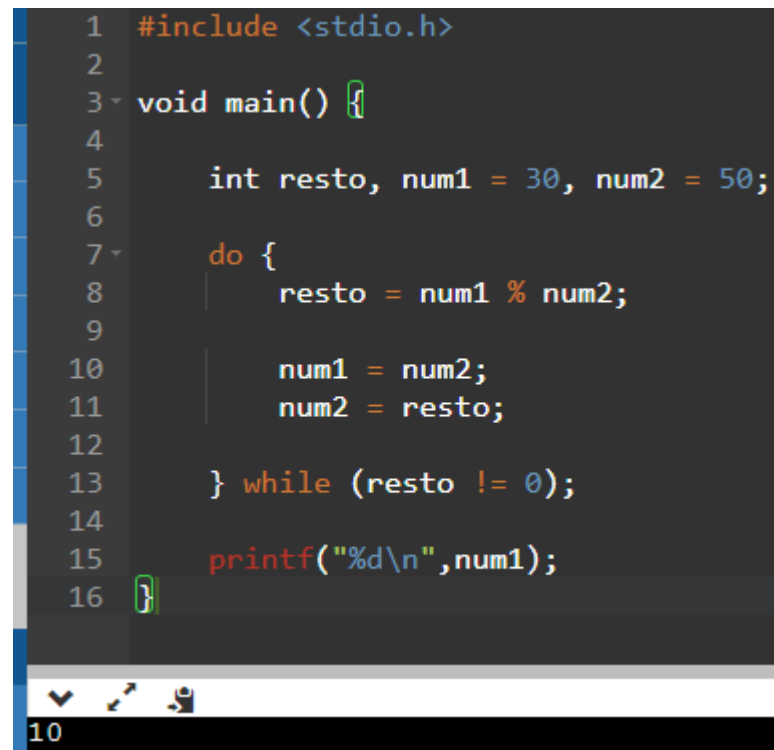
Divisores de 50: 1,2,5,10,25.

O MDC desses dois números é 10, pois é o maior numero natural que está em ambos os conjuntos.

Máximo Divisor Comum

Exemplo em C:

```
1  #include <stdio.h>
2
3  void main() {
4
5      int resto, num1 = 30, num2 = 50;
6
7      do {
8          resto = num1 % num2;
9
10         num1 = num2;
11         num2 = resto;
12     } while (resto != 0);
13
14     printf("%d\n", num1);
15
16 }
```



10

Potenciação

A potenciação expressa um número na forma de potência. Quando um mesmo número é multiplicado diversas vezes, podemos fazer a substituição por uma base (número que se repete) elevada a um expoente (número de repetições).

Exemplo:


$$2 \cdot 2 \cdot 2 = 2^3 = 8$$

Potenciação

Exemplo em C:

Além de simplesmente replicar o número usando o operador '*', pode-se usar a função 'pow' da biblioteca 'math.h', útil para expoentes grandes.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main() {
5
6      float N = 5;
7
8      printf("%f\n", N*N*N*N);
9      printf("%f\n", pow(N,4));
10 }
```



625.000000
625.000000

Radiciação

Por outro lado, a radiciação é a operação oposta da potenciação. Ao elevar um número ao expoente e extrairmos a sua raiz, voltamos ao número inicial.

Exemplo:

$$\sqrt{25} = \sqrt{5^2} = 5$$

Radiciação

Exemplo em C:

Em C a biblioteca 'math.h' contém a função 'sqrt' que realiza a conta de uma raiz quadrada (índice 2) de um número real.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void main() {
5
6     float N = 25;
7
8     printf("%f\n", sqrt(N));
9 }
```

5.000000

Porcentagem

Porcentagem, representada pelo símbolo %, é a divisão de um número qualquer por 100. A expressão 25%, por exemplo, significa que 25 partes de um todo foram divididas em 100 partes. Há várias formas de calcular porcentagem.

Exemplo de uma delas:

$$25\% \text{ de } 300 = 25/100 * 300 = 75$$

Porcentagem

Exemplo em C:

```
1
2  #include <stdio.h>
3
4  void main()
5  {
6      float N = 300;
7
8      printf("25%% de %.2f = %.2f\n", N, 25.0/100.0*N);
9  }
10
```

input

25% de 300.00 = 75.00

Juros Simples

O juro simples é o acréscimo feito a determinado capital no decorrer do tempo. Para determiná-lo, calculamos o produto entre o capital, a taxa de juro e o tempo. A fórmula do juro simples é $J = C \cdot i \cdot t$, em que J é o juro, C é o capital, i é a taxa de juro e t é o tempo.

Exemplo:

Capital de R\$500 valoriza a juros simples de 3% ao mês. Ou seja R\$15 por mês. Depois de um ano foi acumulado R\$180 de juros (15x12).

Na fórmula:

$$J = C \cdot i \cdot T \rightarrow J = 500 \cdot \frac{3}{100} \cdot 12 = 180$$

C i T

Juros Simples

Exemplo em C:

```
1 #include <stdio.h>
2
3 void
4 main ()
5 {
6     float J, C = 500, i = 3;
7     int t = 12;
8     J = C*i/100*t;
9     printf
10     ("Capital de R$%.2f submetido a juros simples de %.2f%% mensais.\nJuros acumulados depois de %d meses = R$%.2f.\n",
11      C, i, t, J);
12 }
13
```

input

```
Capital de R$500.00 submetido a juros simples de 3.00% mensais.
Juros acumulados depois de 12 meses = R$180.00.
```

Juros Compostos

A diferença dos juros compostos para o simples é que o simples sempre é calculado sobre o valor do capital inicial, já o composto é calculado sobre o valor do exercício anterior, ou seja no segundo mês é calculado sobre o valor acumulado no primeiro mês. A fórmula é $M = C (1 + i)^t$ onde M é valor final da transação.

Exemplo:

Capital de R\$500 valoriza a juros compostos de 3% ao mês. Quanto dinheiro é acumulado em 12 meses?

Na fórmula:

$$M = C (1 + i)^t \rightarrow M = 500 \cdot (1 + 3/100)^{12} = 712,88$$

C I T

Juros Compostos

Exemplo em C:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void
5 main ()
6 {
7     float M, C = 500, i = 3;
8     int t = 12;
9     M = C*pow(1+i/100.0, t);
10    printf
11    ("Capital de R$%.2f submetido a juros compostos de %.2f%% mensais.\nMontante acumulado depois de %d meses = R$%.2f.\n",
12     C, i, t, M);
13 }
```

input

```
Capital de R$500.00 submetido a juros compostos de 3.00% mensais.
Montante acumulado depois de 12 meses = R$712.88.
```

Média Simples

A Média Aritmética de um conjunto de dados é obtida somando todos os valores e dividindo o valor encontrado pelo número de dados desse conjunto. **Exemplo:**

Sabendo que as notas de um aluno foram: 8,2; 7,8; 10,0; 9,5; 6,7, qual a média que ele obteve no curso?

$$M_s = \frac{8,2 + 7,8 + 10,0 + 9,5 + 6,7}{5}$$

$$M_s = \frac{42,2}{5}$$

$$M_s = 8,4$$

Média Simples

Exemplo em C:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void
5  main ()
6  {
7      float notas[5] = {8.2, 7.8, 10.0, 9.5, 6.7}, soma = 0.0, media;
8
9      for(int i = 0; i < 5; i++)
10         soma = soma + notas[i];
11
12     media = soma / 5;
13     printf("media = %.2f", media);
14 }
15
```

input

media = 8.44

Média Ponderada

A média ponderada leva o peso de cada informação em consideração no cálculo, seja ele um valor atribuído, seja uma quantidade. Cada valor tem um peso associado. Multiplica-se cada valor por seu peso e divide-se a soma desses valores pela soma dos pesos.

Exemplo:

Considerando as notas e os respectivos pesos de cada uma delas, indique qual a média que o aluno obteve no curso.

Disciplina	Nota	Peso
Biologia	8,2	3
Filosofia	10,0	2
Física	9,5	4
Geografia	7,8	2
História	10,0	2
Língua Portuguesa	9,5	3
Matemática	6,7	4

Média Ponderada

$$M_p = \frac{3 \cdot 8,2 + 2 \cdot 10,0 + 4 \cdot 9,5 + 2 \cdot 7,8 + 2 \cdot 10 + 3 \cdot 9,5 + 4 \cdot 6,7}{3 + 2 + 4 + 2 + 2 + 3 + 4}$$

$$M_p = \frac{24,6 + 20 + 38 + 15,6 + 20 + 28,5 + 26,8}{20}$$

$$M_p = \frac{173,5}{20}$$

$$M_p = 8,7$$

Média Ponderada

Exemplo em C:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void
5  main ()
6  {
7      float notas[7] = {8.2, 10.0, 9.5, 7.8, 10.0, 9.5, 6.7}, soma = 0.0, media;
8      int pesos[7] = {3, 2, 4, 2, 2, 3, 4}, soma_pesos = 0;
9
10     for(int i = 0; i < 7; i++)
11     {
12         soma = soma + notas[i] * pesos[i];
13         soma_pesos = soma_pesos + pesos[i];
14     }
15
16     media = soma / soma_pesos;
17     printf("media = %.2f", media);
18 }
19
```

input

media = 8.68

Biblioteca <math.h>

Algumas funções úteis:

Função	Descrição do comando
floor()	arredonda para baixo
ceil()	arredonda para cima
sqrt()	Calcula raiz quadrada
pow(variável, expoente)	potenciação

Biblioteca <math.h>

Exemplo em C:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main() {
5
6      float N = 2.45;
7
8      printf("%f\n", floor(N));
9
10     printf("%f\n", ceil(N));
11
12     printf("%f\n", sqrt(N));
13
14     printf("%f\n", pow(N, 3));
15 }
```

2.000000
3.000000
1.565248
14.706126