

## Lista de exercícios 2

Operadores unários, binários, funções pré-definidas, biblioteca math.h

---

### Resumo

Este documento apresenta uma série de resolução exercícios de C++, envolvendo funções aritméticas e o uso da biblioteca math.h, desenvolvidos na disciplina de Algoritmos e Técnicas de Programação. Os exercícios têm como objetivo fornecer a prática necessária para compreender e aplicar os conceitos de processamento de dados, fundamentais para o entendimento da disciplina.

### Exercício 1

Se declararmos uma variável em C++ mas não atribuirmos um valor a ela, que valor ela possuirá?

### Resolução

Variáveis recém-criadas não possuem valor definido. Costumamos dizer que a variável recém-criada contém "lixo", isto é, pode conter um valor aleatório.

### Exercício 2

Explique o que são operadores unários e o que são operadores binários. Quais são os operadores aritméticos unários e quais são os binários?

### Resolução

Um operador binário precisa apenas de um operando. Exemplo:  $-a$ . O "a" é um operando e "-" o operador. Operadores unários:  $+$   $-$ .

Operadores binários precisam de dois operandos. Exemplo:  $a + b$  "a" e "b" são operandos e o "+" é um operador. Operadores binários:  $+$   $-$   $/$   $\%$   $*$

### Exercício 3

Qual a diferença entre divisão e div em C++? Faça um exemplo de código que usa divisão e um exemplo que usa div.

## Resolução

A operação `div` é dada quando os dois operadores são inteiros. Isso implica que, por exemplo  $9 / 4$  ( $9 \text{ div } 4$ ) será 2 e não 2.25. Exemplo:

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int a, b, c;
5     a = 9;
6     b = 4;
7     c = a/b;
8     cout << c;
9     return 0;
10 }
```

De forma geral, o `div` desconsidera a parte decimal.

A divisão ocorre quando há, pelo menos, um número real como operando, exemplo:  $9.0/4$ , que resultará em 2.25.

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     float a;
5     int b, c;
6     a = 9;
7     b = 4;
8     c = a/b;
9     cout << c;
10    return 0;
11 }
```

## Questão 4

Complete a frase: se  $n$  é um número par, o resultado de  $n \bmod 2$  sempre será \_\_\_\_\_. Porém, se  $n$  é ímpar, o resultado de  $n \bmod 2$  sempre será \_\_\_\_\_.

## Resolução

Para  $n$  par:

Temos que o resultado  $n \bmod 2$  será 0.

Para  $n$  ímpar:

O resultado de  $n \bmod 2$  será 1.

## Questão 5

Complete a frase: se  $x$  e  $y$  são números inteiros positivos e  $x$  é maior que  $y$ . O resultado de  $y \text{ div } x$  sempre será: \_\_\_\_\_ e o resultado de  $y \bmod x$  sempre será \_\_\_\_\_.

## Resolução

Como  $x > y$  e os dois números são inteiros, a operação será de *div*. Como o intervalo da operação não é real,  $y/x$  será 0. Assim,  $y \bmod x$  resultará em  $y$ , por consequência do  $y \bmod x$  que é 0, então  $y - (x \times 0) = y$ .

## Questão 6

Quais foram as funções aritméticas pré-definidas estudadas? Explique cada uma delas.

## Resolução

Funções aritméticas pré-definidas são um conjunto de funções projetadas por algum programador contidas em alguma biblioteca. Na biblioteca `math.h`, temos: `abs()`, `sqrt()`, `cbrt()`, `pow()`.

`abs()` equivale ao módulo de número. Por exemplo: o módulo de -3 ( $|-3|$ ) = 3. A função `abs()` recebe como parâmetro o valor que receberá seu módulo. A função retornará o valor absoluto (módulo) deste número.

`sqrt()` é uma função que realizará a raiz quadrada de um número.  $\sqrt{4} = 2$  pode ser escrito da seguinte forma: `sqrt(4) = 2`. A função recebe como parâmetro o número que será calculada sua raiz. A função retornará o resultado de sua raiz.

Análogo, ao `sqrt()`, a função `cbrt()` calcula a raiz cúbica de um número. Recebe como parâmetro o número que será calculado e retorna o valor do cálculo.

A função `pow()` é a forma mais completa de elevar e calcular raízes de número. A função `pow()` recebe dois argumentos: a base e o expoente `pow(base, expoente)`. A função retornará o resultado da operação. Podemos calcular a  $n$ -ésima raiz de um número real  $n$  usando a seguinte propriedade:

$$n^{\frac{a}{b}} \Leftrightarrow \sqrt[b]{n^a}$$

Desta forma:

`pow(n, float(a)/b)`

Foi utilizado typecasting no segundo argumento da função, pois o compilador interpretaria a operação  $a/b$  como uma operação do tipo *div*.

## Questão 7

Faça um programa em C++ que solicite ao usuário que digite dois números reais  $a$  e  $b$ . O programa escreverá na tela o resultado de  $a^b$ .

## Resolução

```
1 #include <iostream>
2 #include <math.h>
3 using namespace std;
4 int main () {
5     float base, expoente, resultado;
6
7     cout << "Digite um numero: ";
8     cin >> base;
```

```

9
10 cout << "Deseja elevar este numero a qual expoente? ";
11 cin >> expoente;
12
13 resultado = pow(base, expoente);
14
15 cout << base << " elevado a " << expoente << " eh " << resultado;
16
17
18 return 0;
19 }

```

```

Digite um numero: 2
Deseja elevar este numero a qual expoente? 5
2 elevado a 5 eh 32
-----
Process exited after 4.873 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

## Questão 8

Faça um programa em C++ que solicite ao usuário que digite um número. O programa exibirá na tela a soma da raiz quadrada e da raiz quinta do número digitado.

### Resolução

```

1 #include <iostream>
2 #include <math.h>
3 using namespace std;
4 int main () {
5     float numero, raizQuadrada, raizQuinta, soma;
6
7     cout << "Digite um numero: ";
8     cin >> numero;
9
10    raizQuadrada = sqrt(numero);
11    raizQuinta = pow(numero, float(1)/5);
12
13    soma = raizQuadrada + raizQuinta;
14
15    cout << "A soma da raiz quadrada e quinta de " << numero
16         << " eh " << soma;
17
18    return 0;
19 }

```

```

Digite um numero: 1024
A soma da raiz quadrada e quinta de 1024 eh 36
-----
Process exited after 2.822 seconds with return value 0
Pressione qualquer tecla para continuar. . .

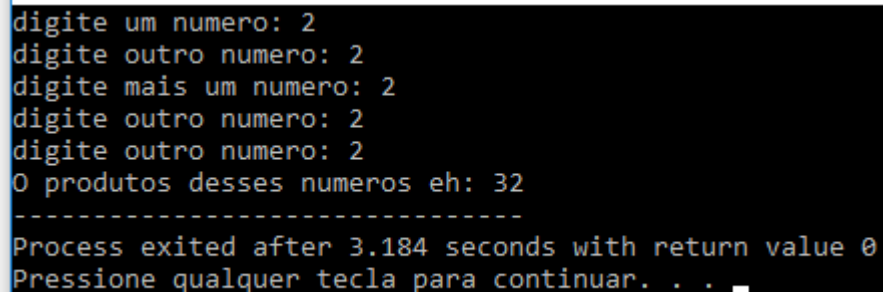
```

## Questão 9

Usando apenas duas variáveis inteiras, faça um programa em C++ que solicite ao usuário que digite 5 números. Ao final, o programa escreverá na tela o produto de todos os números digitados.

### Resolução

```
1 #include <iostream>
2 #include <math.h>
3 using namespace std;
4 int main () {
5     int varUm, varDois;
6     cout << "digite um numero: ";
7     cin >> varUm;
8     cout << "digite outro numero: ";
9     cin >> varDois;
10
11     varUm *= varDois;
12
13     cout << "digite mais um numero: ";
14     cin >> varDois;
15
16     varUm *= varDois;
17
18     cout << "digite outro numero: ";
19     cin >> varDois;
20
21     varUm *= varDois;
22
23     cout << "digite outro numero: ";
24     cin >> varDois;
25
26     varUm *= varDois;
27
28     cout << "O produtos desses numeros eh: " << varUm;
29
30     return 0;
31 }
```



```
digite um numero: 2
digite outro numero: 2
digite mais um numero: 2
digite outro numero: 2
digite outro numero: 2
O produtos desses numeros eh: 32
-----
Process exited after 3.184 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

## Questão 10

Faça um programa em C++ que solicite ao usuário que digite os valores de dois pontos (A e B) em um plano cartesiano (cada ponto é representado por suas coordenadas x e y). O programa dará como saída

a distância entre os dois pontos digitados.

## Resolução

```
1 #include <iostream>
2 #include <math.h>
3 using namespace std;
4 int main () {
5     float eixoXA, eixoYA, eixoXB, eixoYB, distanciaAB;
6
7     cout << "digite o valor do eixo x do ponto A:";
8     cin >> eixoXA;
9
10    cout << "digite o valor do eixo y do ponto A:";
11    cin >> eixoYA;
12
13    cout << "digite o valor do eixo x do ponto B:";
14    cin >> eixoXB;
15
16    cout << "digite o valor do eixo y do ponto B:";
17    cin >> eixoYB;
18
19    distanciaAB = sqrt(pow((eixoXB - eixoXA), 2) + pow((eixoYB - eixoYA), 2));
20
21    cout << "A distancia entre os dois pontos eh " << distanciaAB;
22
23
24    return 0;
25 }
```

```
digite o valor do eixo x do ponto A:9
digite o valor do eixo y do ponto A:4
digite o valor do eixo x do ponto B:8
digite o valor do eixo y do ponto B:2
A distancia entre os dois pontos eh 2.23607
-----
Process exited after 8.239 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

## Questão 11

Faça um programa em C++ que solicite ao usuário um número inteiro que representa uma quantidade de segundos. O programa exibirá na tela o valor convertido em horas, minutos e segundos. Assim, se o usuário digitar o valor 78915, o programa dará como saída 21 horas, 55 min e 15 segundos.

## Resolução

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4
5     int inputSegundos, restoSegundos, horas, minutos, segundos;
```

```

6  cout << "Informe o total de segundos: ";
7  cin >> inputSegundos;
8
9  horas = inputSegundos / (60*60);
10 restoSegundos = inputSegundos % (60*60);
11 minutos = restoSegundos / 60;
12 segundos = restoSegundos % 60;
13
14 cout << horas << "h " << minutos << " min " << segundos << "s";
15 return 0;
16
17 }

```

```

Informe o total de segundos: 3785
1h 3 min 5s
-----
Process exited after 4.056 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

## Questão 12

Faça um programa em C++ que faça exatamente o contrário da questão anterior, ou seja: o usuário deve informar a quantidade de horas, minutos e segundos, e o programa fará a conversão do valor digitado para segundos. Assim, se o usuário digitar os valores 21, 55 e 15, o programa dará como saída 78915 segundos.

### Resolução

```

1  #include <iostream>
2  using namespace std;
3  int main() {
4      int horas, minutos, segundos, total;
5      cout << "informe as horas: ";
6      cin >> horas;
7
8      cout << "informe os minutos: ";
9      cin >> minutos;
10
11     cout << "informe os segundos: ";
12     cin >> segundos;
13
14     total = (horas*3600) + (minutos*60) + segundos;
15     cout << horas << "h " << minutos << " min e " << segundos
16         << "s em segundos eh " << total;
17     return 0;
18
19 }

```

```
15 informe as horas: 21
informe os minutos: 55
informe os segundos: 15
21h 55 min e 15s em segundos eh 78915
-----
Process exited after 3.932 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```