

Atividade Linguagem Haskell

1. Quais operadores aritméticos possui?

R.: A linguagem de programação Haskell possui os seguintes operadores:

Operador	Descrição	Exemplo
+	Soma entre dois inteiros	<pre>Prelude> 1 + 2 3</pre>
-	Subtração entre dois inteiros	<pre>Prelude> <-> 5 4 ou 1 5 - 4</pre>
*	Multiplicação entre dois inteiros	<pre>Prelude> 2 * 2 4</pre>
/	Divisão entre dois inteiros	<pre>Prelude> </> 5 4 ou 1.25 5 / 4</pre>
^	Exponenciação (base qualquer e expoente inteiro)	<pre>Prelude> 3.2^3 32.768000000000001</pre>
div	Parte inteira da divisão	<pre>Prelude> div 20 6 3</pre>
mod	Resto da divisão	<pre>Prelude> mod 5 3 2</pre>
abs	Valor absoluto de um inteiro	<pre>Prelude> abs 6 6</pre>
negate	Inverte o sinal de um inteiro	<pre>Prelude> negate 5 -5</pre>

2. Possui conversão implícita (coersão)? Caso sim, quais são os tipos? O que ocorre se atribuir um double com tamanho máximo para um float, por exemplo?

R.: Possui conversão (coersão) do tipo implícita e explícita:

Coersão	Tipo	Exemplo
Sim	Implícita	<pre>Prelude> 2 + 13.2 15.2</pre>
	Explícita	<pre>Prelude> <2 :: Float> + 10.0 12.0</pre>

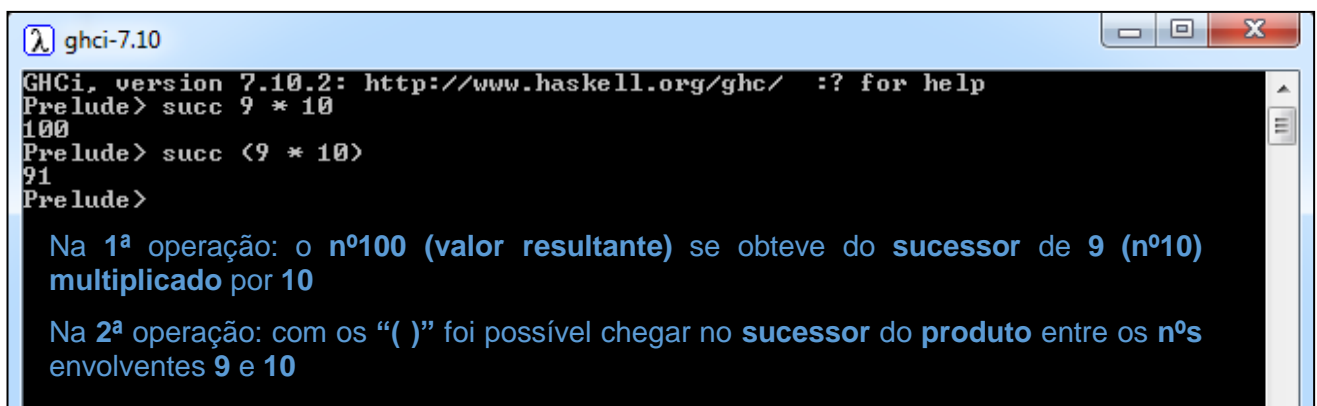
[illegible]

3. Permite chamada de função na expressão aritmética?

R.: Haskell permite conforme a tabela abaixo a chamada de função na expressão aritmética:

Função	Descrição	Exemplo
succ	Sucessor de um número inteiro	<pre>Prelude> <succ 9> + <max 5 4> + 1 16 ou Prelude> succ 9 + max 5 4 + 1 16</pre>
max	Retorna o maior número inteiro	

Outras operações testadas em Haskell:



```
λ ghci-7.10
GHCi, version 7.10.2: http://www.haskell.org/ghc/  :? for help
Prelude> succ 9 * 10
100
Prelude> succ <9 * 10>
91
Prelude>
```

Na 1ª operação: o n°100 (valor resultante) se obteve do **sucessor** de 9 (n°10) **multiplicado** por 10

Na 2ª operação: com os “()” foi possível chegar no **sucessor** do **produto** entre os n°s envoltos 9 e 10

4. Quais os tipos de operadores (unário, binário e ternário)?

R.: Em Haskell somente existe operadores unários.

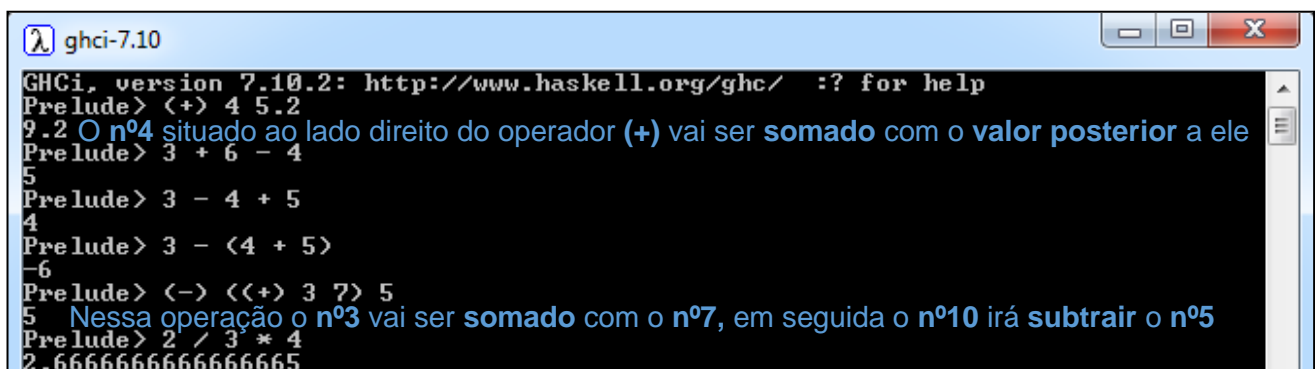
Operador	Tipo	Exemplo
sim	unário	<pre>Prelude> let y += 10</pre>
não	binário	<pre><interactive>:2:12: parse error <possibly incorrect indentation or mismatched brackets> Prelude></pre>
não	ternário	

5. Quais notações suporta?

R.: Suporta essas notações descritas na tabela:

Notação	Descrição	Exemplo
infixa	Operador entre os operandos	<code>Prelude> 2 + 3</code> 5
pré-fixado	Operador antes dos operandos	<code>Prelude> <+> 2 3</code> 5

Outras operações testadas em Haskell:



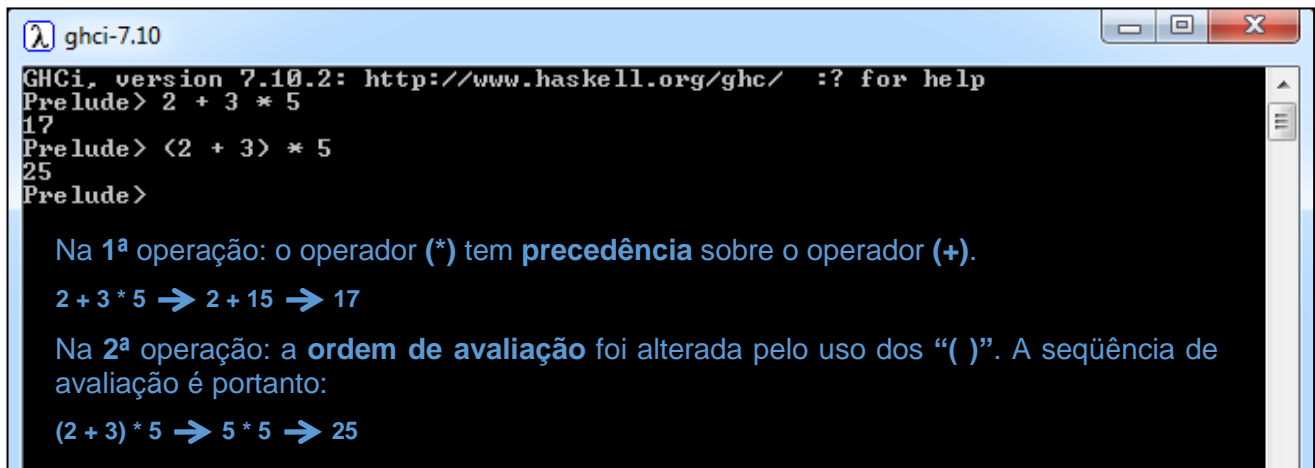
```
ghci-7.10
GHCi, version 7.10.2: http://www.haskell.org/ghc/  :? for help
Prelude> <+> 4 5.2
9.2
O nº4 situado ao lado direito do operador (+) vai ser somado com o valor posterior a ele
Prelude> 3 + 6 - 4
5
Prelude> 3 - 4 + 5
4
Prelude> 3 - <4 + 5>
-6
Prelude> <-> <<+> 3 7> 5
5
Nessa operação o nº3 vai ser somado com o nº7, em seguida o nº10 irá subtrair o nº5
Prelude> 2 / 3 * 4
2.6666666666666665
```

6. Qual a ordem de precedência dos operadores?

R.: Em Haskell a ordem de precedência em que os operadores são considerados é a seguinte:

Precedência	Operadores	Exemplo
7	div, mod, abs, sqrt... (operadores não-associativos)	<code>Prelude> 3 + 9 / 2 > 30</code> False <code>Prelude></code>
6	\wedge , $\wedge\wedge$, ** (operadores associativos à direita)	Ordem de precedência: 5 → 4 → 3
5	*, / (operadores assotictivos à esquerda)	
4	+, - (operadores assotictivos à esquerda)	
3	==, /=, <, <=, >, >= (operadores não-associativos)	
2	&& (operadores associativos à direita)	
1	(operadores associativos à direita)	
0	>>, >>= (operadores assotictivos à esquerda)	

Outras operações testadas em Haskell:



```
ghci-7.10
GHCi, version 7.10.2: http://www.haskell.org/ghc/  :? for help
Prelude> 2 + 3 * 5
17
Prelude> (2 + 3) * 5
25
Prelude>
```

Na 1ª operação: o operador (*) tem **precedência** sobre o operador (+).

$2 + 3 * 5 \rightarrow 2 + 15 \rightarrow 17$

Na 2ª operação: a **ordem de avaliação** foi alterada pelo uso dos "()". A sequência de avaliação é portanto:

$(2 + 3) * 5 \rightarrow 5 * 5 \rightarrow 25$

7. Quais são os operadores relacionais?

R.: Os operadores relacionais utilizados em Haskell são:

Operador	Descrição	Exemplo
==	Igualdade	<pre>Prelude> (2 + 3) == (8 - 3) True</pre>
/=	Diferença	<pre>Prelude> 5 /= (4 * 2 - 3) False</pre>
<	Menor	<pre>Prelude> (2 + 3) < 6 True</pre>
<=	Menor ou igual	<pre>Prelude> (2 * 3) <= 6 True</pre>
>	Maior	<pre>Prelude> (4 + 2) > (2 * 3) False</pre>
>=	Maior ou igual	<pre>Prelude> (8 - 3 * 2) >= (div 15 3) False</pre>

8. Quais são os operadores lógicos?

R.: Em Haskell os operadores lógicos ficam representados dessa maneira:

Operador	Descrição	Exemplo
&&	E	<pre>Prelude> <4 > 2> && <2 < 3> True</pre>
	Ou	<pre>Prelude> <8 >= 3 * 2> <8 <= 3 * 2> True</pre>
not	Negação	<pre>Prelude> not <5 < 3> True</pre>

9. Implementa avaliação de curto-circuito?

R.: Sim implementa a avaliação de curto-circuito. Os operadores de curto-circuito utilizados em Haskell são && e || (com tipo de resultado booleano).

Curto-circuito	Operadores	Exemplo
Sim	&&	<pre>Prelude> let a = 10 Prelude> let b = 20 Prelude> <a > b> && a> && <a /= b> False</pre>

10. Quais os tipos de atribuição que são utilizados (entre os 6)?

R.: Possui: atribuição simples, condicional, unário, expressão, lista e modo misto.

Atribuição	Tipo	Exemplo
Sim	Simple	<pre>Prelude> let a = 5</pre>
Sim	Condicional	
Não	Composto	
Sim	Unário	
Sim	Expressão	
Sim	Lista	
Sim	Modo misto	<pre>Prelude> let a = 5 :: Integer Prelude> let b = 5.5 :: Double Prelude> let a = b Prelude> a 5.5</pre>

Exercícios de operadores:

1. Analise os exemplos abaixo e verifique a ordem que as operações foram executadas:

a) $5 + 6 * 4$ resultado = 29

R.: Primeiro foi realizado o **produto**, resultante dos **nºs** envolventes **6** e **4**, finalizando com a **soma** do **nº5**.

b) $5 + 6 * 4$ resultado = 44

R.: Primeiro foi realizado a **soma** entre o **nº5** e **nº6**, finalizando com o valor resultante **multiplicado** pelo **nº4**.

c) $5 + 6 - 3$ resultado = 8

R.: Ordem indiferente.

d) $5 * 6 + 4$ resultado = 34

R.: Primeiro foi realizado o **produto**, resultante dos **nºs** envolventes **5** e **6**, finalizando com a **soma** do **nº4**.

2. Apresente o resultado das seguintes expressões:

a) $2 * (5/20) + 30 / (15 * 2)$

R.: 1.5.

b) $35 / 6 - 2$

R.: 3.8333333333333333.

c) $2 * 5/20 + 30 \% 15 * 2$

R.: 9,5.

d) $3 + 5 \leq 2 * 4$

R.: True.

e) $i = 10; a = i++; i = 10 / a = 11$

R.: 0,909090909090909090909090909091.

f) $i = 10; a = ++i; i = 10 / a = 11$

R.: 0,909090909090909090909090909091.

g) Escreva $(10 - 4 > 7)$;

R.: False.

h) Escreva ("Resultado = " + $2 + 4$);

R.: Resultado = $2 + 4$.

i) Escreva ("Resultado = " + $(2 + 4)$);

R.: Resultado = 6.

j) $12 / 3 * 45 * + 13 * -$

R.: 18,5.

k) 48

R.: 0,5.

l) $48 /$

R.: 0,5.

m) $* + 342$

R.: 14.

3. Escreva as expressões seguintes em notação pré-fixada e pós-fixada:

a) $a+b$

R.: Pré-fixada: $+ a b$

Pós-fixada: $a b +$

b) $(a+b)/c$

R.: Pré-fixada: $/ (+ a b) c$

Pós-fixada: $(a b +) c /$

c) $((a+b+c)/(a-b-c)+a)/y$

R.: Pré-fixada: $/ (+ (/ (+ + a b c) (- - a b c)) a) y$

Pós-fixada: $(((a b c + +) (a b c - -) /) a +) y /$

d) $a*a*a+a+c+d*b$

R.: Pré-fixada: $+ + + * * a a a a c * d b$

Pós-fixada: $a a a * * a c + + d b * +$

e) $(2+4) * (3+5)$

R.: Pré-fixada: $* (+ 2 4) (+ 3 5)$

Pós-fixada: $(2 4 +) (3 5 +) *$

f) $5 + 2 * 9$

R.: Pré-fixada: $+ 5 * 2 9$

Pós-fixada: $5 2 9 * +$

4. Escreva as expressões seguintes em notação infixa:

a) $(* (+ 5 (+ 2 3)) 4)$

R.: $((5 + (2 + 3)) * 4).$

b) $(* (+ 3 4) 2)$

R.: $((3 + 4) * 2).$

c) $(/ (* (+ 8 a) (- b 4)) 2)$

R.: $(((8 + a) * (b - 4)) / 2).$

d) $(/ (* (+ (- (+ 7 6) 5) 4) 3) 2)$

R.: $(((((7 + 6) - 5) + 4) * 3) / 2).$