

Problemas para Experimentação dos Interpretadores Haskell

Este documento apresenta a lista de problemas que servirão de base à avaliação dos interpretadores Haskell desenvolvidos como projeto da disciplina Linguagem de Programação: Conceitos e Paradigmas. Para cada problema, crie um programa escrito na linguagem implementada em seu projeto. Esses programas serão executados sobre um conjunto de testes durante a avaliação. As soluções propostas devem ser adaptadas às características de cada linguagem implementada, mas sempre observando as funcionalidades exigidas.

Problema 1:

Crie um programa que, dados três valores numéricos x , y e c , onde x e y são números racionais e c é um número inteiro, previamente armazenados no código-fonte, avalie a expressão $x^2 - y + c$ e imprime seu resultado na tela.

Problema 2:

Uma competição é organizada em três fases, e a participação na terceira fase depende da pontuação nas duas primeiras. A entrada contém um único conjunto de testes, que deve ser lido do teclado. A primeira linha da entrada contém dois números inteiros n e p , representando respectivamente o número de competidores ($1 \leq n \leq 1000$) e o número mínimo de pontos para ser convidado ($1 \leq p \leq 1000$). Cada uma das n linhas seguintes contém dois números inteiros x e y indicando a pontuação de um competidor nas duas primeiras fases ($0 \leq x \leq 400$) e ($0 \leq y \leq 400$). Crie um programa que, para cada conjunto de testes, imprime na saída padrão uma única linha contendo um único inteiro, indicando o número de competidores que poderão participar da terceira fase. Veja os exemplos abaixo.

Entrada	Saída
3 100 50 50 100 0 49 50	2

Entrada	Saída
4 235 100 134 0 0 200 200 150 150	2

Problema 3:

Crie um programa que leia duas matrizes numéricas e, quando possível, imprima a soma e o produto dessas matrizes. Caso uma operação não possa ser realizada para as matrizes lidas, imprima uma mensagem informando da impossibilidade.

Problema 4:

Crie um subprograma chamado *mdc*, com três argumentos *n*, *m* (passados por valor) e *r* (passado por referência), nesta ordem. O subprograma *mdc* deve calcular o maior divisor comum entre dois números naturais estritamente positivos *n* e *m*, de acordo com o seguinte algoritmo recursivo:

- Se *n* for um divisor de *m*, *n* é o maior divisor comum de *n* e *m*.
- Se *m* for um divisor de *n*, *m* é o maior divisor comum de *n* e *m*.
- Se *n* não for um divisor de *m*, e se *m* for maior que *n*, então o maior divisor comum de *m* e *n* é também o maior divisor comum de *n* e do resto da divisão de *m* por *n*.

O subprograma deve retornar seu resultado por meio de parâmetro *r*, que deve ser posteriormente impresso na tela pelo programa principal.

Problema 5:

Defina o tipo *rational_t* para representar números racionais. O tipo *rational_t* deve ser representado como um registro com campos inteiros *numerador* e *denominador*. Em seguida, escreva os seguintes subprogramas:

- A) Subprograma que, dados dois parâmetros inteiros *a* e *b*, onde $b \neq 0$, retorna um valor desse tipo para representar a fração a/b .
- B) Subprograma que, dados dois parâmetros do tipo *rational_t*, retorna *true* se eles representam o mesmo número racional ou *false*, em caso contrário.
- C) Subprogramas que retornem um valor *rational_t* correspondente a soma, negação, subtração, multiplicação, inverso e divisão entre valores *rational_t*, passados como parâmetros (um subprograma por operação).

No programa principal, invoque cada um dos subprogramas e imprima os resultados produzidos, no formato *rational_t*, indicando numerador e denominador.