

1 Instruções Importantes

Nesta seção são apresentadas diversas informações relevantes referente a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem funcional Haskell.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- documentação do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código compila ou roda no *ghci*. Programas com erros de sintaxe receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo de documentação tenha a identificação do autor do trabalho;
4. arquivo compactado com os artefatos estão devidamente identificados com nome e matrícula.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, documentação e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

A documentação do código deve conter informações relevantes para auxiliar no entendimento do código fonte. Ressalta-se que na documentação não deve conter cópias do fonte – afinal o seu fonte é um dos artefatos enviado, mas deve apresentar as decisões de projetos tomadas, em especial as estruturas de dados para modelar e solucionar o problema.

O trabalho deverá ser apresentado ao professor da disciplina e, só serão avaliados após a realização da entrevista, i.e., trabalhos que não forem apresentando não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, documentação e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina.**

Dias de Atraso	Nota
1	$n \cdot 0.98$
2	$n \cdot 0.96$
3	$n \cdot 0.92$
4	$n \cdot 0.84$
5	$n \cdot 0.68$
6	$n \cdot 0.36$
7	0

Figura 1: Modificações na pilha para soma de dois valores

Atrasos serão penalizados por uma função exponencial de dias de atrasos, i.e., será reduzido do percentual da nota a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

2 Especificação Técnica do Trabalho

Uma máquina de pilha é uma máquina abstrata que usa uma estrutura de pilha para armazenar os valores de entradas para suas operações. Por exemplo, para realizar a soma de dois números, x e y , é necessário empilhar os valores de x e y e, posteriormente, executar a operação de adição. O resultado da operação é, então, inserido no topo da pilha. A Figura 1 ilustra o funcionamento da máquina para realizar a soma de dois valores. Inicialmente a pilha está vazia, em seguida o valor de x é empilhado e, na sequência, o valor de y também é empilhado. Posteriormente, a soma é realizada, a qual desempilha os dois valores do topo, y e x , e empilha o resultado da soma no topo da pilha.

O objetivo desse trabalho é implementar um simulador de uma máquina de pilha simples. A Tabela 2 apresenta o conjunto de instruções da máquina de pilha. Assim, implemente um interpretador na linguagem Haskell que, dada uma sequência finita de instruções da máquina (programa), as interprete, i.e., faça as devidas modificações na pilha de dados, memória e entrada/saída padrão. Observe que a “memória” da máquina é um mapeamento de nome de variáveis para valores, ou seja, há instruções para empilhar o valor de uma variável na pilha (*push v*) e para armazenar o valor do topo da pilha em uma variável (*pop v*).

O interpretador deverá começar a execução a partir da primeira instrução e interpretar as demais instruções em sequência. Somente quando alguma instrução de desvio, *jump* ou *dcond*, for executada que a próxima instrução não será, necessariamente, a seguinte. Nesse caso, a instrução a ser executada é a instrução da i -ésima posição do programa.

A Figura 2 mostra um programa para a máquina de pilha que ler uma sequência de valores digitados pelo usuário e imprime, na saída padrão, o somatório desses números. Observe que as instruções das linhas 7, *dcond* 15, e 14, *jump* 4, alteram qual a próxima instrução executada para a instrução da linha 15 e 4, respectivamente.

Instrução	Nº de operandos	Descrição
push x	0	empilha o valor de x
pop x	1	desempilha o valor do topo da pilha e salva na variável x
add	2	desempilha os dois últimos valores da pilha e empilha a sua soma
sub	2	desempilha os dois últimos valores e empilha a diferença entre o elemento abaixo topo pelo elemento do topo
mult	2	desempilha os dois últimos valores da pilha e empilha o seu produto
div	2	desempilha os dois últimos valores e empilha a divisão inteira entre o elemento abaixo topo pelo elemento do topo
mod	2	desempilha os dois últimos valores e empilha o resto da divisão entre o elemento abaixo topo pelo elemento do topo
read	0	empilha o valor lido da entrada padrão
print	1	desempilha o valor do topo e imprime na saída padrão
jump i	0	altera a próxima instrução a ser executada para a instrução i
dcond i	1	desempilha o topo da pilha e, se for o valor <i>true</i> , altera a próxima instrução a ser executada para a instrução i
cmp	2	desempilha os dois últimos valores da pilha e empilha <i>true</i> se forem iguais ou <i>false</i> caso contrário
cint i	0	empilha a constante inteira i na pilha
cstr s	0	empilha a string s na pilha
nop	0	sem efeito

Tabela 1: Conjunto de instruções da máquina de pilha

```

1  cint 0
2  cint 10
3  pop i
4  push i
5  cint 0
6  cmp
7  dcond 15
8  read
9  add
10 push i
11 cint 1
12 sub
13 pop i
14 jump 4
15 print
    
```

Figura 2: Exemplo de programa que soma 10 inteiros inserido pelo usuário.

3 Entrega do Trabalho

A data da entrega do trabalho será até o dia **23 de julho de 2021**.