

Disciplina	Curso	Turno	Período	Valor
Lab. Ling. de Programação	Engenharia da Computação	Tarde	4º	15+1 Pontos
Aluno(a)		Matrícula		
Aluno(a)		Matrícula		
Ao entregar esse exame eu dou minha palavra que eu o fiz somente com minha dupla, entendendo que eu posso consultar qualquer material publicamente disponível, exceto aqueles disponibilizados por colegas que estão fazendo esse curso ou que já o fizeram no passado.				

Prova 2

Instruções

- 1) Não é permitido usar nenhuma função/predicado da biblioteca padrão nem de terceiro, exceto aquelas mencionadas nos slides das aulas. Ou seja, todas as funções/predicados auxiliares devem estar presentes em sua solução.
- 2) A solução de cada questão deve ser enviada em um arquivo separado com o nome da função/predicado seguido da extensão correspondente:
 - a. **codificacao.hs**
 - b. **codificacao.pl**
- 3) **A entrega deve ser feita exclusivamente via sistema acadêmico (SIGAA) indicando a dupla em um arquivo README.txt.**

Questão) (16pts) Um esquema para compressão de textos sem perda de dados é substituir uma sequência de elementos repetidos por quantas vezes ele apareceu seguido de uma única ocorrência dele. Por exemplo, a sequência AABCCCCCCCCCCCCDDABBB pode ser codificada como 2A1B11C2D1A3B. A operação reversa também pode ser aplicada para recuperar o texto original. Considere que o texto original nunca possui dígitos. Desenvolva as operações de `codificar` e `decodificar` como funções em Haskell e predicados em Prolog.

a) (8pts) Em Haskell.

```
> codificar "AABCCCCCCCCCCCCDDABBB"
"2A1B11C2D1A3B"
> decodificar "2A1B11C2D1A3B"
"AABCCCCCCCCCCCCDDABBB"
```

OBS1: Use a função `isDigit` para verificar se um caractere é um dígito e use as funções `int2str` e `str2int` para converter um inteiro em uma string e uma string em um inteiro respectivamente:

```
isDigit :: Char -> Bool
isDigit c = c >= '0' && c <= '9'

int2str :: String -> Int
int2str i = show i

str2int :: String -> Int
str2int s = read s
```

b) (8pts) Em Prolog.

```
?- codificar("AABCCCCCCCCCDDABBB", R).  
R = "2A1B11C2D1A3B".  
?- decodificar("2A1B11C2D1A3B", R).  
R = "AABCCCCCCCCCDDABBB".
```

OBS1: Use o predicado a seguir para verificar se um caractere é um dígito:

```
is_digit(X) :- atom_length(X,1), X @>= '0', X @=< '9'.
```

OBS2: Use os predicados `string_chars` e `number_chars` de Prolog que convertem strings ou inteiros em lista de caracteres. Exemplos de uso:

```
?- string_chars("abc", L).  
L = ['a', 'b', 'c'].  
?- string_chars(S, ['a', 'b', 'c']).  
S = "abc".  
?- number_chars(123, L).  
L = ['1', '2', '3'].  
?- number_chars(N, ['1', '2', '3']).  
N = 123.
```