

TRABALHO 1

Implementar computacionalmente, em linguagem acordada com o professor, um programa que seja capaz de:

- a) (Pontuação: 2,5) Aceitar, como entrada fornecida pelo usuário, alfabetos formatados segundo formalismo da teoria de conjuntos, isto é, cada símbolo deve ser separado por uma vírgula e devem estar dentro de chaves. Os símbolos aceitos pelo programa devem ser somente letras e dígitos numéricos. Não é permitido símbolos repetidos no alfabeto. Espaços devem permitidos como entrada, mas devem ser ignorados, ou seja, não são símbolos do alfabeto.

Exemplos:

- 1) {a,b,c}
 - 2) {0,1}
 - 3) { a , b , 0 , 1 }
- b) (Pontuação: 2,5) Realizar união dos alfabetos entrados pelo usuário. O Resultado da união não pode ter símbolos repetidos.
- c) (Pontuação: 2,5) Verificar sobre quais alfabetos, dentre os entrados pelo usuário, uma sequência de símbolos, também fornecida pelo usuário, é palavra.
- d) (Pontuação: 2,5) Determinar os prefixos, sufixos e subpalavras de uma palavra. O programa deve: eliminar prefixos repetidos; eliminar sufixos repetidos; eliminar supalavras repetidas. Utilize o símbolo representado por 136 do código ASCII para representar a palavra vazia.

Abaixo está o *screenshot* de um exemplo de programa (disponível no moodle) que atende a todos os requisitos deste trabalho.

Lista de alfabetos:	
1	{a,b,c}
2	{0,1}