

1. Considere a seguinte Expressão Regular (ER)

[FfEeVv]

e selecione então as afirmações verdadeiras:

- 0/0 **A** esta ER faz match com o texto
fEv
- 0/0 **B** esta ER é equivalente a
(F|f)(E|e)(V|v)
- 0/0 **C** se a ER acima for usada como argumento do comando 'grep' aplicado ao texto
Este Exemplo serve

serão assinalados 7 matches.

2. Considere a seguinte Expressão Regular (ER)

[a-z][a-zA-Z]*

e selecione então as afirmações verdadeiras:

- 0/0 **A** a ER acima apanha/concorda com qualquer palavra de um texto de entrada.
- 0/0 **B** a ER acima é equivalente à ER a seguir:
[a-zA-Z]+
- 0/0 **C** o texto de entrada
aquiVai um Exemplo
concorda com a ER acima.
- 0/0 **D** o texto de entrada
aquiVaiUmExemplo
concorda com a ER acima.

3. Observe com atenção a especificação FLex seguinte

%%

01 { printf("Janeiro\n"); }

02 { printf("Fevereiro\n"); }

03 { printf("Março\n"); }

[0-9]+ { printf("(%d) ",atoi(yytext)); }

%%

e selecione então as afirmações corretas:

- 0/0 **A** O Filtro de Texto gerado pelo FLex a partir da especificação acima, se aplicado ao texto de entrada

Aqui temos um exemplo de 01/02 que dá 0,5.

produzia como saída o resultado:

**Aqui temos um exemplo de Janeiro
/Fevereiro
que dá (0) ,(5) .**

- 0/0 **B** O Filtro de Texto gerado pelo FLex a partir da especificação acima, se aplicado ao texto de entrada

Aqui temos um exemplo de 01/02 que dá 0,5.

produzia como saída o resultado:

**Janeiro
Fevereiro
(0) (5)**

- 0/0 **C** O Filtro de Texto gerado pelo FLex a partir da especificação acima, se aplicado ao texto de entrada

Outro exemplo pode ser 2020-02-5.

produzia como saída o resultado

**Outro exemplo pode ser (2020)-Fevereiro
-(5) .**

- 0/0 **D** Se à especificação Flex acima acrescentássemos uma 5ª regra na forma:

.\n {;}

nada se alterava no resultado final produzido pelo Filtro de Texto gerado a partir da nova especificação.