

1. Analise com cuidado a seguinte especificação de um Filtro de Texto escrita em Flex:

```
%x ABRE ATRIB
%%
\[a-zA-Z]+      { printf("... do ... (%s)\n",yytext+1); BEGIN ABRE; }
<ABRE>\>        { BEGIN INITIAL; }
<ABRE>[a-zA-Z]+ { printf("< %s - ",yytext); BEGIN ATRIB;}
<ABRE>.\|n
<ATRIB>[ ]*=[ ]*\ " { ; }
<ATRIB>\ "       { BEGIN ABRE; }
<ATRIB>[^"]*     { printf("%s >\n",yytext); }
.\|n             { ; }
%%
```

diga então o que faz o Filtro gerado a partir dela pelo Flex?

- 0/0 **A** Não faz nada porque estão declarados dois estados (Start-Conditions) quando só pode haver um.
- 0/0 **B** Não faz nada porque pra sair de um estado é obrigatório mudar para o estado **INITIAL**, o que não acontece na 6ª regra acima.
- 0/0 **C** Analisa um texto anotado em XML e identifica todos os pares **<nome - valor>** de cada *Atributo* de um *Elemento* que abre e retira todo o restante texto.
- 0/0 **D** Analisa um texto anotado em XML e remove da saída todos os pares **<nome - valor>** de cada *Atributo* de um *Elemento* que abre, deixando passar o restante texto.
- 0/0 **E** Analisa um texto anotado em XML e imprime todos os fragmentos de texto entre as *Marcas* de *Abertura* e de *Fecho*, retirando todo o restante texto.

2. Considere a seguinte especificação de um Filtro de Texto em Flex

```
%x ABRE ATRIB
%%
\[a-zA-Z]+ { printf("Atributos do Elemento (%s)\n",yytext+1); BEGIN ABRE; }
<ABRE>\> { BEGIN INITIAL; }
<ABRE>[a-zA-Z]+ { printf("< %s - ",yytext); BEGIN ATRIB;}
<ABRE>.\n
<ATRIB>[ ]*=[ ]*\" { ; }
<ATRIB>\" { BEGIN ABRE; }
<ATRIB>[^"]* { printf("%s >",yytext); }
.\n { ; }
%%
```

e diga então quais as afirmações abaixo verdadeiras.

- 0/0 **A** Se a última regra mudasse para
 `.\n`
o comportamento do Filtro gerado pelo Flex alterava-se.
- 0/0 **B** Se as 2 primeiras regras fossem alteradas, assumindo elem como um **char ***, para
 `\[a-zA-Z]+ { elem=strdup(yytext+1); BEGIN ABRE; }`
 `<ABRE>\> { printf("são os Atributos do Elemento (%s)\n",elem); BEGIN INITIAL; }`
o resultado do Filtro era o mesmo, mas a saída era melhor porque não apareciam *Elementos* sem *Atributos*.
- 0/0 **C** O Filtro gerado pelo Flex a partir dessa especificação imprime todos os *Elementos* que se abrem mesmo que não tenham *Atributos*.
- 0/0 **D** Se a 4ª regra fosse removida e a última regra mudasse para
 `<*>.\n { ; }`
o comportamento do Filtro gerado pelo Flex alterava-se.
- 0/0 **E** Se as 3 primeiras regras fossem alteradas, assumindo elem como um **char ***, para
 `\[a-zA-Z]+ { elem=strdup(yytext+1); BEGIN ABRE; }`
 `<ABRE>\> { if (cnt) printf("são os Atributos do Elemento (%s)\n",elem);`
 `cnt=0; BEGIN INITIAL; }`
 `<ABRE>[a-zA-Z]+ { cnt++; printf("< %s - ",yytext); BEGIN ATRIB;}`
o resultado do Filtro era o mesmo, mas a saída era melhor porque não apareciam *Elementos* sem *Atributos*.