

(BOCA: P2\_2019\_Q4\_a) Problema: Faça um programa para analisar as senhas de grupos de usuários e ao final apresentar o grupo com maior quantidades de senhas fortes, o com maior de senhas fracas e o com maior de senhas inválidas. Uma senha é dita forte quando ela é válida e contém pelo menos uma letra maiúscula, uma letra minúscula, um número e um caractere não alfanumérico. Uma senha é dita fraca quando ela é válida, mas não é forte. Uma senha é inválida quando ele possui menos de 6 caracteres.

O programa deve ser entregue modularizado, para isso crie:

- a) Uma função que recebe um caractere e retorna verdadeiro se ele for uma letra maiúscula e falso caso contrário. Cabeçalho: int EhLetraMaiuscula(char c);
- b) Uma função que recebe um caractere e retorna verdadeiro se ele for uma letra minúscula e falso caso contrário. Cabeçalho: int EhLetraMinuscula(char c);
- c) Uma função que recebe um caractere e retorna verdadeiro se ele for um número e falso caso contrário. Cabeçalho: *int EhNumero(char c);*
- d) Uma função responsável por analisar a senha de um usuário e retornar 0 se ela for inválida, 1 se for fraca e 2 se for forte. A função deverá ler a senha da entrada padrão para fazer a análise. Assumir que a senha será dada no formato "(SENHA)", ou seja, entre parênteses. Cabeçalho: int AnalisaSenha();
- Entrada: Um número n indicando a quantidade de grupos seguido dos grupos em si. Cada grupo é composto por uma lista de usuários. Um usuário é dado por um código e uma senha fornecida entre parênteses. A lista de usuários de um grupo termina quando o código de um usuário é -1. Cada usuário é dado em uma linha. Os grupos vão de 1 a n e podem ser vazios (ou seja, sem usuários). Veja o formato de entrada nos exemplos abaixo.
- Saída: A saída deverá imprimir o tipo das senhas por usuário (fraca, forte ou inválida) e por fim os grupos com maior quantidade de cada uma delas. Para cada um dos grupos, a formatação seguirá com: uma linha contendo o índice do grupo com a mensagem "GRUPO:" seguida do índice; posteriormente, uma linha para cada usuário do grupo informando "FRACA", "FORTE" ou "INVALIDA" dependendo do tipo da senha de cada usuário. Por fim, o programa deverá apresentar 3 mensagens sendo uma em cada linha: "FORTES: GRUPO(#1) QTD(#2)", "FRACAS: GRUPO(#1) QTD(#2)" e "INVALIDAS: GRUPO(#1) QTD(#2)", em que #1 e #2 representam respectivamente o índice e a quantidade do grupo com a maior quantidade de senhas do tipo daquela mensagem (fortes, fracas e inválidas respectivamente). Veja o formato da saída nos exemplos abaixo.



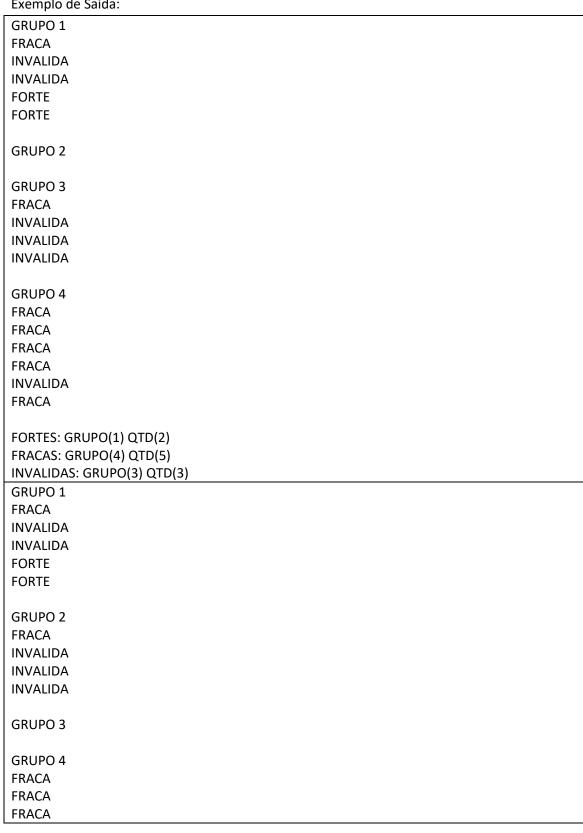
#### Exemplo de Entrada:

```
100(abcdefg)
101(abc)
102(Ab2&r)
103(Ab2&ifdz)
104(AB2$ifdz)
-1
-1
100(AAAAAAA)
1002(AAA)
2(123)
103(ABCD)
-1
10(Ab2sger)
20(Ab&sger)
30(jb&s1er)
40(jb&sger)
50()
60(123456)
-1
6
100("&&7868sd)
101(3453)
102(Ab2&r)
103(Ab2&ifdz)
104(AB2$ifdz)
-1
100(UJGw98c)
1002(hyas)
2(7890)
103(Z)
-1
-1
10(Ab2sger)
20(Ab&sger)
30(jb&s1er)
40(jb&sger)
50()
60(123456)
-1
10(!@#$%"&*)
20(A!@#$%"&*123c)
30(ABCDZa*)
40(1234AA5912)
50()
60(123456)
1001(*&"%$%IHUTug3423)
1002(*&"%$%IHUTug3425)
```



1003(2345)			
1004(\$eA1Z)			
-1			

#### Exemplo de Saída:





FRACA	 	
INVALIDA		
FRACA		
GRUPO 5		
FRACA		
FORTE		
FRACA		
FRACA		
INVALIDA		
FRACA		
GRUPO 6		
FORTE		
FORTE		
INVALIDA		
INVALIDA		
FORTES: GRUPO(1) QTD(2)		
FRACAS: GRUPO(4) QTD(5)		
INVALIDAS: GRUPO(2) QTD(3)		