

Qualidade de Software - 2019/20

<p>A) Determina o índice do último elemento de x que é igual a y (devolve -1 caso nenhum elemento verifique a condição).  <b>1.2: x=[]; y=-1</b>          Teste: x=[2,3,5]; y=2; <b>1.3: x=[1,2]; y=2</b>          Resultado esperado: 0</p>	<p>B) Determina o índice do último zero em x (devolve -1 caso 0 não ocorra em x)  <b>1.2: x=[]</b>  <b>1.3: x=[1,2,0]</b>          Teste: x=[0,1,0]          Resultado esperado: 2 <b>Está a devolver o primeiro 0</b></p>
<pre>public static int findLast (int[] x, int y) {     for (int i=x.length-1; i&gt;0; i--)     {         if (x[i]==y)         {             return i;         }     }     return -1; }</pre>	<pre>public static int lastZero (int[] x) {     for (int i=0; i&lt;x.length; i++)     {         if (x[i]==0)         {             return i;         }     }     return -1; }</pre>
<p>C) Determina o número de elementos positivos em x.          Teste: x=[-4,2,0,2]          Resultado esperado: 2  <b>1.2: x=[]</b>  <b>1.3: x=[-4,2,-1,2]</b></p>	<p>D) Determina o número de elementos em x que são ímpares ou positivos.          Teste: x=[-3,-2,0,1,4]          Resultado esperado: 3  <b>1.2: x=[]</b>  <b>1.3: x=[3,-2,0,1,4]</b></p>
<pre>public static int countPositive (int[] x) {     int count=0;     for (int i=0; i&lt;x.length; i++)     {         if (x[i]&gt;=0) <b>Devia estar "&gt;"</b>         {             count++;         }     }     return count; }</pre>	<pre>public static int oddOrPos (int[] x) {     int count=0;     for (int i=0; i&lt;x.length; i++)     {         if (x[i]%2==1    x[i]&gt;0) <b>Devia estar "Math.abs(x[i])%2==1"</b>         {             count++;         }     }     return count; }</pre>

1. Sabendo que os 4 programas apresentam defeitos, responda às seguintes questões:
  - 1.1. Identifique o(s) defeito(s) de cada um dos programas.
  - 1.2. Se possível, indique um caso de teste em que o defeito não é verificado.
  - 1.3. Se possível, indique um caso de teste que verifique o defeito, mas não resulte em falha.

- 1.4. Se possível, indique um caso de teste que verifique o defeito, e resulte em falha.
- 1.5. Retifique o programa de forma a eliminar o defeito e verifique se o caso de teste apresenta o valor esperado.

2. Dado o seguinte programa:

Determina o número de ocorrências de 0 em x.
<pre>public static int numZero (int[] x) {     int count = 0;     for (int i=1; i&lt;x.length; i++)     {         Devia estar "i=0"         if (x[i]==0)         {             count++;         }     }      return count; }</pre>

- 2.1. Identifique o defeito do programa.
- 2.2. Se possível, indique um caso de teste que não resulte em falha.  $x=[2,0,3]$
- 2.3. Se possível, indique um caso de teste que resulte em falha.  $x=[0,2,3]$
- 2.4. Se possível, indique um caso de teste que resulte em erro.  $x=null$

3. **Role-playing:** Forme um grupo com outro aluno. Um dos elementos do grupo assume o papel de programador *java* para que o colega de grupo; o *tester*, possa responder às seguintes questões:

- 3.1. Identifique o(s) defeito(s) de cada um dos programas.
- 3.2. Se possível, indique um caso de teste em que o defeito não é verificado.
- 3.3. Se possível, indique um caso de teste que verifique o defeito, mas não resulte em falha.
- 3.4. Se possível, indique um caso de teste que verifique o defeito, e resulte em falha.
- 3.5. Retifique o programa de forma a eliminar o defeito e verifique se o caso de teste apresenta o valor esperado.

Uma vez terminada e validada esta tarefa, troque de papel com o seu colega de grupo (i.e. o programador passa a ser o *tester* e vice-versa).