

Listing 1: Singers and songs paired up

```
<Songs>
  <row><Sgr name="Eagles" genre="rock"/>
    <Song lg="en">Hotel California</Song></row>
  <row><Sgr name="Eagles" genre="rock"/>
    <Song lg="en">Seven Bridges Road</Song></row>
  <row><Sgr name="H_Belafonte" genre="reggae"/>
    <Song lg="cpe">Day O</Song></row>
  <row><Sgr name="H_Belafonte" genre="reggae"/>
    <Song lg="en">Jamaica Farewell</Song></row>
  <row><Sgr name="J_Prasad" genre="classical"/>
    <Song lg="him">Ajahoon Na Bhejiho</Song></row>
  <row><Sgr name="J_Prasad" genre="classical"/>
    <Song lg="pa">Mera Dil Darda</Song></row>
  <row><Sgr name="J_Prasad" genre="classical"/>
    <Song lg="en">Never a Result</Song></row>
</Songs>
```

Listing 2: Singers nesting songs with language and genre subelements

```
<Songs>
  <Sgr name="Eagles">
    <Song>
      <lg>en</lg>Hotel California<genre>rock</genre>
    </Song>
    <Song>
      <lg>en</lg>Seven Bridges Road<genre>rock</genre>
    </Song>
  </Sgr>
  <Sgr name="H_Belafonte">
    <Song>
      <lg>cpe</lg>Day O<genre>reggae</genre>
    </Song>
    <Song>
      <lg>en</lg>Jamaica Farewell<genre>reggae</genre>
    </Song>
  </Sgr>
  <Sgr name="J_Prasad">
    <Song>
      <lg>him</lg>Ajahoon Na Bhejiho<genre>classical</genre>
    </Song>
    <Song>
      <lg>pa</lg>Mera Dil Darda<genre>classical</genre>
    </Song>
    <Song>
      <lg>en</lg>Never a Result<genre>classical</genre>
    </Song>
  </Sgr>
</Songs>
```

1. Mark the appropriate choices to complete the following XQuery listing to transform Listing 1 into Listing 2.

```
declare function local:nestWithinSingers($root)
{
  element { name($root) } {
    let $songs := $root//Song
    let $singers := $root//Sgr
    ----- (: PART (a) :) -----

    return
    element { "Sgr" } {
      attribute { "name" } { $singername },

      for $song in $songs
    ----- (: PART (b) :) -----

      return
    ----- (: PART (c) :) -----

      element { "lg" } {
        text { $song/@lg } },

      $song/text(),
    ----- (: PART (d) :) -----
      element { "genre" } {
    }
  }
}
};

local:nestWithinSingers(doc("paired-new.xml")/Songs)
```

(a) (12 points) Identify all correct snippets for the for clause of PART (a)

A. for \$singername in
distinct-values(\$songs/preceding-sibling::*[1]/@name)

B. for \$singername in
distinct-values(\$songs/preceding-sibling::*[1]/@*)

C. for \$singername in
distinct-values(\$songs/preceding-sibling::*[1]["@name"]/@*)

D. for \$singername in
distinct-values(\$songs/preceding::node()[@name]/@name)

Solution:

```
for $singername in
  distinct-values($songs/preceding-sibling::*[1]/@name)
AND
for $singername in
  distinct-values($songs/preceding::node()[@name]/@name)
```

(b) (12 points) Identify all correct snippets for the where clause of PART (b)

E. where every \$x in \$song/preceding-sibling::node()[1]/@name
satisfies \$x = \$singername

F. where some \$x in \$song/preceding-sibling::node()[last()]/@name
satisfies \$x = \$singername

G. where not (some \$x in \$song/preceding-sibling::node()[1]/@name
satisfies \$x != \$singername)

H. where some \$x in \$song/preceding-sibling::*, \$y in \$x[1]/@name
satisfies \$y = \$singername

Solution:

```
where some $x in $song/preceding-sibling::node()[last()]/@name
satisfies $x = $singername
AND
where some $x in $song/preceding-sibling::*, $y in $x[1]/@name
satisfies $y = $singername
```

(c) (12 points) Identify all correct snippets for the element constructor of PART (c)

I. element { name(\$singers[1]/following-sibling::*[1]/..) } {

J. element { name(\$songs[name(.)="Song"]) } {

K. element {
name(\$singers[@name=\$singername][1]/following-sibling::*[1]) } {

L. element {name(\$singers[@name=\$singername][1]/following::Song) } {

Solution:

```
element {  
name($singers[@name=$singername][1]/following-sibling::*[1]) } {
```

(d) (12 points) Identify all correct snippets for the text constructor of PART (d)

M. `text { $song/preceding::*[last()]/@genre } }`

N. `text { $song/preceding-sibling::*[1]/@genre } }`

O. `text { $song/parent::*//@genre } }`

P. `text { $song/preceding::*[1]/@genre } }`

Solution:

```
text { $song/preceding-sibling::*[1]/@genre } }  
AND  
text { $song/parent::*//@genre } }  
AND  
text { $song/preceding::*[1]/@genre } }
```