

1. Katalog produktów i zestawienie ich cen. Przykładowe zamówienie.

```
<catalog>
  <product dept="WMN">
    <number>557</number>
    <name language="en">Fleece Pullover</name>
    <colorChoices>navy black</colorChoices>
  </product>
  <product dept="ACC">
    <number>563</number>
    <name language="en">Floppy Sun Hat</name>
  </product>
  <product dept="ACC">
    <number>443</number>
    <name language="en">Deluxe Travel Bag</name>
  </product>
  <product dept="MEN">
    <number>784</number>
    <name language="en">Cotton Dress Shirt</name>
    <colorChoices>white gray</colorChoices>
  </product>
  <colorChoices>navy black</colorChoices>
  <desc>Our favorite shirt!</desc>
</product>
</catalog>

<prices>
  <priceList effDate="2006-11-15">
    <prod num="557">
      <price currency="USD">29.99</price>
      <discount type="CLR">10.00</discount>
    </prod>
    <prod num="563">
      <price currency="USD">69.99</price>
    </prod>
    <prod num="443">
      <price currency="USD">39.99</price>
      <discount type="CLR">3.99</discount>
    </prod>
  </priceList>
</prices>

<order num="00299432" date="2006-09-15" cust="0221A">
  <item dept="WMN" num="557" quantity="1" color="navy"/>
  <item dept="ACC" num="563" quantity="1"/>
  <item dept="ACC" num="443" quantity="2"/>
  <item dept="MEN" num="784" quantity="1" color="white"/>
  <item dept="MEN" num="784" quantity="1" color="gray"/>
  <item dept="WMN" num="557" quantity="1" color="black"/>
</order>
```

2. Dla katalogu produktów, napisz odpowiednie wyrażenie XPath, które wybiera:

- a. Lista produktów.
- b. Lista produktów, tylko nazwa.
- c. Lista zawierająca tylko akcesoria (nazwy), czyli dept: ACC.
- d. Numery produktów, które mają kolor navy black.
- e. Numery produktów, które mają element <desc>.
- f. Produkt, który jest trzeci na liście.
- g. Nazwa produktu, który jest trzeci na liście.
- h. Nazwa i numer produktu, który jest trzeci na liście (skonstruować sekwencję XPath).
- i. Nazwa i numer każdego produktu, ale zwracane jako nowy element XML <opis> (aby odczytał zawartość elementów, należy całą sekwencję umieścić wewnątrz nawiasów klamrowych).
- j. Jeżeli produkt jest typu MEN albo WMN, to wybrać jak wyżej.
- k. Dla każdego produktu, wybrać nazwę, numer, typ - jako sekwencja, ale tylko wartości tekstowe elementów.

3. Dla katalogu produktów, napisz odpowiednie wyrażenie XQuery.

np. 1) Lista produktów typu akcesoria, alfabetycznie wg nazw:

```
for \$prod in doc("catalog.xml")/catalog/product
where \$prod/@dept = "ACC"
order by \$prod/name
return \$prod/name
```

for

Ta klauzula określa iteracje poprzez kolejne węzły produktów. Pozostała część zapytania FLWOR jest przetwarzana dla każdego z czterech produktów. Za każdym razem, zmienna \$prod jest związana z innym elementem produktu. Znaki dolara są wykorzystywane do wskazania nazw zmiennych w XQuery.

where

Ta klauzula wybiera produkty tylko typu ACC. Ma to takie samo znaczenie, jak predykat ([@dept = "ACC"]) w wyrażeniu XPath.

order by

Ta klauzula sortuje wynik ze względu na nazwy produktów.

return

Ta klauzula określa, że elementy nazwa potomne względem elementu produkt mają zostać zwrócone.

np. 2) Wynik można umieścić wewnątrz nowego znacznika (XML lub HTML). Nawiasy klamrowe są potrzebne, w przeciwnym przypadku zawartość będzie odczytana literalnie:

```
<ul>{
  for $product in doc("catalog.xml")/catalog/product
  where $product/@dept='ACC'
  order by $product/name
  return $product/name
}</ul>
```

np. 3) wyrażenie warunkowe IF

```
for $prod in (doc("catalog.xml")/catalog/product)
return if ($prod/@dept = 'ACC')
  then <accessoryNum>{data($prod/number)}</accessoryNum>
  else <otherNum>{data($prod/number)}</otherNum>
```

Funkcja data() zwraca tylko wartość elementu. Funkcja string() zwraca wartość elementu w postaci tekstowej.

4. Strona www prezentująca ofertę sklepu (za pomocą XSLT).