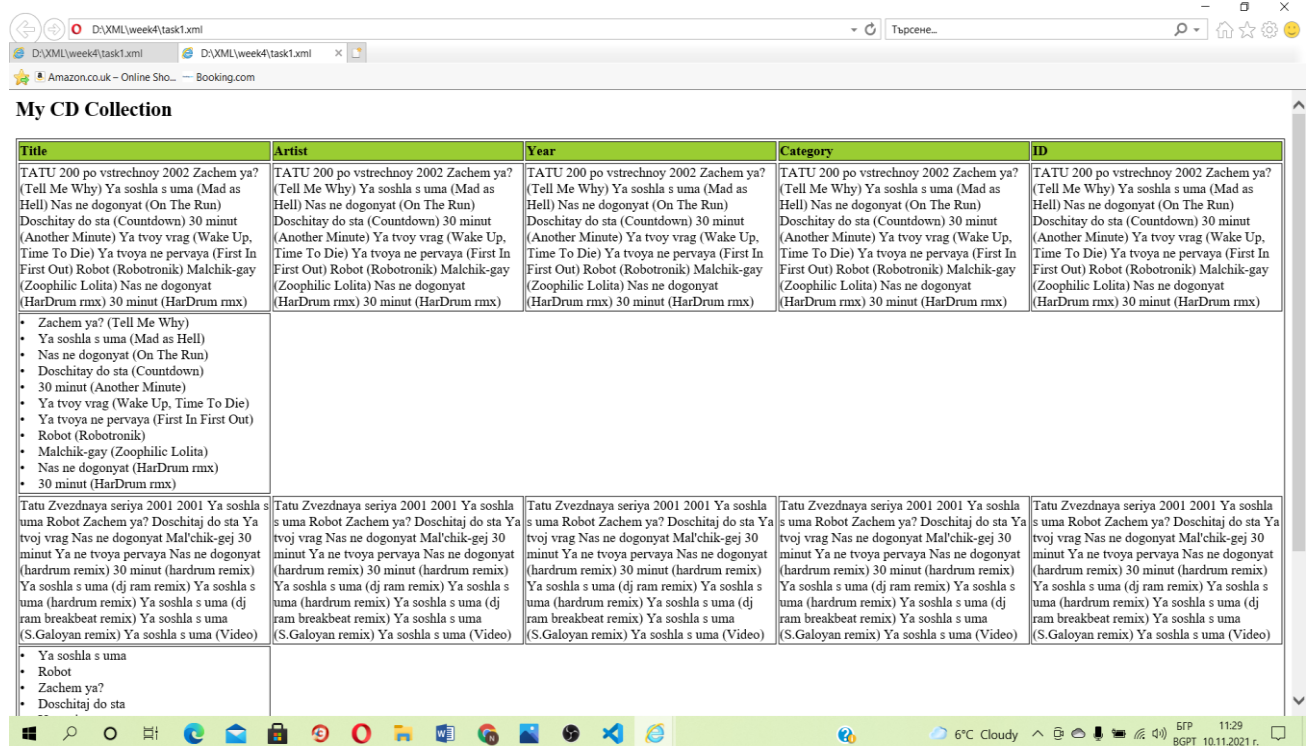


1-ва задача

Заменям стойността на атрибута select с „.” и когато отворя дадения xml-документ с Internet Explorer получавам таблица, запълнена с всичката дадена информация за съответната категория във всяко поле на таблицата. Прилагам скрийншот:



Title	Artist	Year	Category	ID
TATU 200 po vstrechnoy 2002 Zachem ya? (Tell Me Why) Ya soshla s uma (Mad as Hell) Nas ne dogonyat (On The Run) Doschitay do sta (Countdown) 30 minut (Another Minute) Ya tvoy vrug (Wake Up, Time To Die) Ya tvoya ne pervaya (First In First Out) Robot (Robotronik) Malchik-gay (Zoophilic Lolita) Nas ne dogonyat (HarDrum rmx) 30 minut (HarDrum rmx)	TATU 200 po vstrechnoy 2002 Zachem ya? (Tell Me Why) Ya soshla s uma (Mad as Hell) Nas ne dogonyat (On The Run) Doschitay do sta (Countdown) 30 minut (Another Minute) Ya tvoy vrug (Wake Up, Time To Die) Ya tvoya ne pervaya (First In First Out) Robot (Robotronik) Malchik-gay (Zoophilic Lolita) Nas ne dogonyat (HarDrum rmx) 30 minut (HarDrum rmx)	TATU 200 po vstrechnoy 2002 Zachem ya? (Tell Me Why) Ya soshla s uma (Mad as Hell) Nas ne dogonyat (On The Run) Doschitay do sta (Countdown) 30 minut (Another Minute) Ya tvoy vrug (Wake Up, Time To Die) Ya tvoya ne pervaya (First In First Out) Robot (Robotronik) Malchik-gay (Zoophilic Lolita) Nas ne dogonyat (HarDrum rmx) 30 minut (HarDrum rmx)	TATU 200 po vstrechnoy 2002 Zachem ya? (Tell Me Why) Ya soshla s uma (Mad as Hell) Nas ne dogonyat (On The Run) Doschitay do sta (Countdown) 30 minut (Another Minute) Ya tvoy vrug (Wake Up, Time To Die) Ya tvoya ne pervaya (First In First Out) Robot (Robotronik) Malchik-gay (Zoophilic Lolita) Nas ne dogonyat (HarDrum rmx) 30 minut (HarDrum rmx)	TATU 200 po vstrechnoy 2002 Zachem ya? (Tell Me Why) Ya soshla s uma (Mad as Hell) Nas ne dogonyat (On The Run) Doschitay do sta (Countdown) 30 minut (Another Minute) Ya tvoy vrug (Wake Up, Time To Die) Ya tvoya ne pervaya (First In First Out) Robot (Robotronik) Malchik-gay (Zoophilic Lolita) Nas ne dogonyat (HarDrum rmx) 30 minut (HarDrum rmx)
Tatu Zvezdnaya seriya 2001 2001 Ya soshla s uma Robot Zachem ya? Doschitaj do sta Ya tvoy vrug Nas ne dogonyat Mal'chik-gej 30 minut Ya ne tvoya pervaya Nas ne dogonyat (hardrum remix) 30 minut (hardrum remix) Ya soshla s uma (dj ram remix) Ya soshla s uma (hardrum remix) Ya soshla s uma (dj ram breakbeat remix) Ya soshla s uma (S.Galoyan remix) Ya soshla s uma (Video)	Tatu Zvezdnaya seriya 2001 2001 Ya soshla s uma Robot Zachem ya? Doschitaj do sta Ya tvoy vrug Nas ne dogonyat Mal'chik-gej 30 minut Ya ne tvoya pervaya Nas ne dogonyat (hardrum remix) 30 minut (hardrum remix) Ya soshla s uma (dj ram remix) Ya soshla s uma (hardrum remix) Ya soshla s uma (dj ram breakbeat remix) Ya soshla s uma (S.Galoyan remix) Ya soshla s uma (Video)	Tatu Zvezdnaya seriya 2001 2001 Ya soshla s uma Robot Zachem ya? Doschitaj do sta Ya tvoy vrug Nas ne dogonyat Mal'chik-gej 30 minut Ya ne tvoya pervaya Nas ne dogonyat (hardrum remix) 30 minut (hardrum remix) Ya soshla s uma (dj ram remix) Ya soshla s uma (hardrum remix) Ya soshla s uma (dj ram breakbeat remix) Ya soshla s uma (S.Galoyan remix) Ya soshla s uma (Video)	Tatu Zvezdnaya seriya 2001 2001 Ya soshla s uma Robot Zachem ya? Doschitaj do sta Ya tvoy vrug Nas ne dogonyat Mal'chik-gej 30 minut Ya ne tvoya pervaya Nas ne dogonyat (hardrum remix) 30 minut (hardrum remix) Ya soshla s uma (dj ram remix) Ya soshla s uma (hardrum remix) Ya soshla s uma (dj ram breakbeat remix) Ya soshla s uma (S.Galoyan remix) Ya soshla s uma (Video)	Tatu Zvezdnaya seriya 2001 2001 Ya soshla s uma Robot Zachem ya? Doschitaj do sta Ya tvoy vrug Nas ne dogonyat Mal'chik-gej 30 minut Ya ne tvoya pervaya Nas ne dogonyat (hardrum remix) 30 minut (hardrum remix) Ya soshla s uma (dj ram remix) Ya soshla s uma (hardrum remix) Ya soshla s uma (dj ram breakbeat remix) Ya soshla s uma (S.Galoyan remix) Ya soshla s uma (Video)

Като съответните промени в xsl-документа са:

```
<tr>  
  <td><xsl:value-of select="."/></td>  
  <td><xsl:value-of select="."/></td>  
  <td><xsl:value-of select="."/></td>  
  <td><xsl:value-of select="."/></td>  
  <td><xsl:value-of select="."/></td>  
</tr>
```

2-ра задача

- 1. Стойностите на всички track елементи, чиито атрибут length е равен на '4:04' и са включени в елемент cd, имащ id равно на 8c0a600b

Израз:

```
//cd[@id='8c0a600b']//track[@length='4:04']
```

- 2. Всички track елементи на четни/нечетни позиции

За четни позиции:

```
//track[position() mod 2 = 0]
```

За нечетни позиции:

```
//track[position() mod 2 != 0]
```

Коментар:

position() връща позиция, номерът по ред на възела

mod 2 връща по модул 2

- 3. Стойностите на всички track елементи, чиято дължина на текста е по-голяма от 35

Израз:

```
//track[string-length(text()) > 35]
```

Коментар:

Функцията text() връща истина за всеки текстов възел. Ако функцията върне стойност истина, то възелът ще бъде избран.

Функцията string-length(text()) връща число, равно на броя на знаците в text().

- 4. Дължината на текста на всички track елементи, чиято дължина на текста е по-голяма от 15

Израз:

```
//track[string-length(text()) > 15]/string-length(text())
```

Коментар:

Функцията string-length(text()) връща число, равно на броя на знаците в text().

- 5. Последния track елемент от всяко cd

Израз:

```
//cd//track[last()]
```

Коментар:

Функцията last() избира последния елемент на контекстния възел с име track

- 6. Петия track елемент от всяко cd

Израз:

```
//cd//track[position()=5]
```

Коментар:

Position()=5 връща 5-тия елемент от track

- 7. Броя на track елементите за всяко cd

Израз:

```
//cd/tracklist/count(track)
```

Коментар:

Функцията count() изчислява броя на елементите в подадения списък – track

- 8. Всички track елементи, които съдържат 'Ya soshla s uma'

Израз:

```
//track[contains(text(),'Ya soshla s uma')]
```

Коментар:

Функцията contains() връща всички елементи track, които съдържат в съдържанието си исканото.

- 9. Всички track елементи, които започват с буквата 'D'

Израз:

```
//track[starts-with(text(),'D')]
```

Коментар:

Функцията starts-with проверява дали първият низ започва с втория низ и връща true или false.

- 10. Всички track елементи, които завършват с изрази 'sta'

Израз:

```
//track[ends-with(text(),'sta')]
```

Коментар:

Функцията ends-with проверява дали първият низ завършва с втория низ и връща true или false.

- 11. Стойностите на всички track елементи, разпечатани с главни букви

Израз:

```
catalog/cd/tracklist/track/upper-case(text())
```

- 12. Стойността на елемента year, който е под-елемент на елемента cd, съдържащ под-под-елемент track с length = '3:55' и имащ стойност 'Robot (Robotronik)'

Израз:

```
//cd[//track[@length='3:55' and text()='Robot (Robotronik)']]/year
```

- 13. Среден брой track елементи от всички налични cd елементи

Израз:

```
sum(//cd//count(track)) div catalog/count(cd)
```

Коментар:

count(//cd//track) дава броя на всички елементи track

div – функция за разделяне

catalog/count(cd) дава броят cd елементите в catalog = 2

- 14. За всеки елемент cd изведете стойността на под-елементите му title и year, спазвайки следния модел:
Заглавие: title_value; Година на издаване: year_value

Израз:

```
//cd/concat('Заглавие:', title/text(), '; Година на издаване:', year)
```

Коментар:

Функцията concat обединява два или повече низа и връща резултантния низ.

3та задача

- 1. Избират елемента:

1. tracklist

2. year

и извеждат негова стойност или стойност на негов атрибут

1.

Израз:

```
self::node()/descendant-or-self::node()/child::tracklist/attribute::*
```

Коментар:

задава всички възли с име tracklist в йерархията на контекстния възел, т.е. всички негови преки или непреки наследници с това име в йерархията на контекстния възел и избира всички атрибути на tracklist.

2.

Израз:

```
self::node()/descendant-or-self::node()/child::year
```

- 2. Избират атрибутите num и id съответно на елементите tracklist и cd

Израз:

```
self::node()/descendant-or-self::node()/child::tracklist/attribute::num
```

Коментар:

attribute::num избира атрибута num на tracklist.

Израз:

```
self::node()/descendant-or-self::node()/child::cd/attribute::id
```

- 3. Избират всички елементи track, които се намират преди track елемента със стойност 'Doschitay do sta (Countdown)'

Израз:

```
self::node()/descendant-or-self::node()/self::track[text()='Doschitay do sta (Countdown)']/preceding-sibling::*
```

Коментар:

self::track[text()='Doschitay do sta (Countdown)'] избира track елемента със стойност 'Doschitay do sta (Countdown)'

preceding-sibling::* избира всички предшестващи братя и сестри на контекстния възел вляво от него

- 4. Избират всички елементи track, които се намират след track елемента със стойност 'Doschitay do sta (Countdown)'

Израз:

```
self::node()/descendant-or-self::node()/self::track[text()='Doschitay do sta (Countdown)']/following-sibling::*
```

Коментар:

following-sibling::* избира всички братя и сестри на контекстния възел следващи вдясно от него

- 5. Избират всички елементи track, които се намират след track елемента със стойност 'Doschitay do sta (Countdown)' и имат стойност на атрибута length '4:04'

Израз:

```
self::node()/descendant-or-self::node()/self::track[text()='Doschitay do sta (Countdown)']/following-sibling::*[attribute::length='4:04']
```

- 6. Всички стойности на под-елементите на всички елементи cd в документа

Израз:

```
self::node()/descendant-or-self::node()/self::cd/child::*
```

Коментар:

child::* селектира всички преки наследници на контекстния възел

- 7. Стойностите на всички елементи, които имат атрибут с име id

Израз:

```
self::node()/descendant-or-self::node()/self::*[attribute::id]
```

- 8. Стойностите на всички елементи, които имат какъвто и да било атрибут

Израз:

```
self::node()/descendant-or-self::node()/self::*[attribute::*]
```

- 9. Атрибута num с максимална стойност

Израз:

```
max(self::node()/descendant-or-self::node()/attribute::num)
```

Коментар:

Функцията max() връща максималната стойност на своя аргумент

4та задача

- 1. Селектираш всички стойности на елемента track, който е под-елемент на tracklist, имащ атрибут num равен на 1.
2. Подредете резултата от 1. по азбучен ред
3. Подредете резултата от 1. в обратен азбучен ред

1.

```
//tracklist[@num=1]/track
```

2. Във for-цикъл обхождам резултатите от 1. И с order by \$i ascending се подреждат по азбучен ред:

```
for $i in //tracklist[@num=1]/track
```

```
order by $i ascending
```

```
return $i/text()
```

3. Във for-цикъл обхождам резултатите от 1. И с order by \$i descending

се подреждат в обратен азбучен ред:

```
for $i in //tracklist[@num=1]/track
```

```
order by $i descending
```

```
return $i/text()
```

- 2. Създавам следната структура:

```
<records>
```

```
  <record cd_ID="CD_ID_VALUE" artist="ARTIST_VALUE">
```

```
    <info>Title: TITLE_VALUE, Year: YEAR_VALUE, Track numbers:  
COUNT_OF_TRACKS</info>
```

```
  </record>
```

```
</records>
```

В която стойностите на CD_ID_VALUE, ARTIST_VALUE, TITLE_VALUE, YEAR_VALUE, COUNT_OF_TRACKS отговарят съответно на стойностите на атрибута id на елемента cd, на елемента artist, на елемента title, на елемента year, на броя на елементите track за съответния елемент cd

В елемента records с for-цикъл обхождаме всички cd-та и за всяко създавам елемента record, като изпълняваме условията на задачата:

```
element records
```

```
{
```

```
for $i in catalog/cd
```

```
return
```

```
  element record
```

```
  {
```

```
    attribute cd_ID {$i/@id}, attribute artist {$i/artist},
```

```
    element info
```

```
    {
```

```
      text { "Title:" }, text {$i/title}, text { " ", Year:" }, text {$i/year}, text { " ", Track numbers:" },  
text { count($i/tracklist/track) }
```

```
    }
```

```
}  
}
```

- 3. Създаващ списък със стойността на всички track елементи от всички cd елементи, следващ модела:

```
<tracks>  
  <track> TRACK_NAME_1</track>  
  <track> TRACK_NAME_2</track>  
  <track> .....</track>  
  <track> TRACK_NAME_N</track>  
</tracks>
```

С for-цикъл обхождаме всички cd-та и за всяко създаваме елемент tracks, пълен с елементите track от tracklist:

```
for $i in catalog/cd  
return  
element tracks  
{  
  for $j in $i/tracklist/track/text()  
  return  
  element track{$j}  
}
```

5-та задача

1:

1.1) Функцията ни се казва task1_1 с параметър – елемента \$catalog; връща списък от елементи:

```
declare function local:task1_1($catalog as element()) as element()*  
{  
  $catalog//tracklist[@num=1]/track  
};
```

```
<task1_1>  
  {local:task1_1(catalog)}  
</task1_1>
```


1.2) Функцията ни се казва task1_2 с параметър – елемента \$catalog; връща списък от елементи:

```
declare function local:task1_2($catalog as element()) as element()*  
{  
  for $i in $catalog/cd/tracklist[@num = "1"]/track  
  order by $i ascending  
  return $i  
};
```

```
<task1_2>  
  {local:task1_2(catalog)}  
</task1_2>
```

1.3) Функцията ни се казва task1_3 с параметър – елемента \$catalog; връща списък от елементи:

```
declare function local:task1_3($catalog as element()) as element()*  
{  
  for $i in $catalog/cd/tracklist[@num = "1"]/track  
  order by $i descending  
  return $i  
};
```

```
<task1_3>  
  {local:task1_3(catalog)}  
</task1_3>
```

2: Функцията ни се казва task1_2 с параметър – елемента \$catalog; връща списък от елементи:

```
declare function local:task2($catalog as element()) as element()*  
{  
  for $i in $catalog/cd
```

```

return
    element record
    {
        attribute cd_ID {$i/@id}, attribute artist {$i/artist},
        element info
        {
            text { "Title:"}, text {$i/title}, text { " , Year:"} ,text{$i/year}, text { " , Track numbers:"},
            text{count($i/tracklist/track)}
        }
    }
};

```

element records

```

{
<task2>
    {local:task2(catalog)}
</task2>
}

```

3. Функцията ни се казва task1_3 с параметър – елемента \$catalog; връща списък от елементи:

```

declare function local:task3($catalog as element()) as element()*
{
    for $j in $catalog/tracklist/track/text()
    return
        element track{$j}
};

```

for \$i in catalog/cd

return

element tracks

```
{  
  <task3>  
    {local:task3($i)}  
  </task3>  
}
```