

1A

/real-estate/owners/owner
/child::real-estate/child::owners/child::owner
//owner
//child::owner
/descendant-or-self::node()/owner
/descendant-or-self::node()/child::owner

1B

/real-estate/agencies/agency/name/text()
/real-estate/agencies/agency/name/child::text()
//agency/name/text()
Incorrect:
//name/text()

1C

distinct-values(/real-estate/owners/owner/address/country)
distinct-values(/real-estate/owners/owner/address/country/text())
distinct-values(//owner/address/country)
distinct-values(//owner//country)
Incorrect:
distinct-values(//country)

1D

/real-estate/owners/owner[@idOwner = "o2"]/name/text()
//owner[@idOwner = "o2"]/name/text()
//owner[@idOwner = 'o2']/name/text()
//owner[./@idOwner = "o2"]/name/text()
//owner[self::node()/@idOwner = "o2"]/name/text()
//owner[attribute::idOwner = "o2"]/name/text()
//owner[./attribute::idOwner = "o2"]/name/text()

2

//owner/address[country/text() = 'United States']/postalCode

//owner/address[country = 'United States']/postalCode

//owner/address[child::country = 'United States']/postalCode

//owner/address[./country = 'United States']/postalCode

//owner/address[./country = 'United States']/postalCode

//owner[address/country = 'United States']/address/postalCode

//owner[./country = 'United States']/postalCode

//owner//country[text() = 'United States']/../postalCode

//owner//country[. = 'United States']/preceding-sibling::postalCode

3

```
//flat[@comfort = "C" and @property != "p2"]/rate/text()
//flat[(@comfort = "C") and (@property != "p2")]/rate/text()
//flat[@comfort = "C"][@property != "p2"]/rate/text()
//rate[parent::flat/@comfort = "C" and parent::flat/@property != "p2"]/text()
//rate[parent::flat[@comfort = "C"] and parent::flat[@property != "p2"]]/text()
//rate[../@comfort = "C" and ../@property != "p2"]/text()

//flat[@comfort = "C"][not(@property = "p2")]/rate/text()
```

4

/real-estate/properties/property[1]/features/feature[3]

/real-estate/properties/child::property[1]/features/feature[3]

//property[1]/features/feature[3]

//property[1]//feature[3]

//property[position() = 1]//feature[position() = 3]

5

```
//property[not(@owner) and count(features/feature) >= 2]/name  
//property[not(@owner) and count(./feature) >= 2]/name  
//property[not(@owner) and count(self::node()//feature) >= 2]/name  
//property[count(@owner) = 0 and count(./feature) >= 2]/name  
  
//property[not(@owner) and features/feature[2]]/name
```

Incorrect:

```
//property[@owner = "" and count(features/feature) >= 2]/name  
//property[@owner = null and count(features/feature) >= 2]/name  
//property[not(@owner) and count(./feature) >= 2]/name
```

6

```
//owner[not(@idOwner = //property/@owner)]
```

Incorrect:

```
//owner[@idOwner != //property/@owner]
```

```
//owner[not(//property/[@owner = @idOwner])]
```

```
//owner[count(@idOwner = //property/@owner) = 0]
```

```
//owner[count(@idOwner = //property/@owner) = 1]
```

```
//owner[count(@idOwner != //property/@owner) = 0]
```

```
//owner[count(@idOwner != //property/@owner) = 1]
```

7

```
//flat[rate/text() < avg(//flat/rate/text())]  
//flat[rate < avg(//flat/rate)]  
//flat[rate < avg(//rate)]  
//flat[rate < avg(..flat/rate)]  
//flat[rate < avg(..//rate)]  
//flat[rate < avg(data(//rate))]
```

8

```
//owner[
  @idOwner = //property[
    @idProperty = //flat[features/feature = 'bath']/@property
  ]/@owner
]/name
```


9

```
sum(  
  //flat[@comfort = "C" or @comfort = "D"] [  
    @property = //property[@owner = 'o1']/@idProperty  
  ]/rate  
)
```

... //flat[@comfort = ("C", "D")] ...

10

```
/descendant::flat[position() = 1 or position() = last()]/name  
/descendant-or-self::flat[position() = 1 or position() = last()]/name  
//flat[  
    (not(preceding::flat) and not(ancestor::flat))  
    or  
    (not(following::flat) and not(descendant::flat))  
]  
//flat[  
    (not(preceding::flat or ancestor::flat))  
    or  
    (not(following::flat or descendant::flat))  
]  
//flat[  
    (count(preceding::flat) + count(ancestor::flat) = 0)  
    or  
    (count(following::flat) + count(descendant::flat) = 0)  
]  
((//flat)[position() = 1 or position() = last()]/name
```

Incorrect:

```
/descendant::flat[1 or last()]/name  
//flat[position() = 1 or position() = last()]/name  
/descendant-or-self::node()/flat[position() = 1 or position() = last()]/name
```

11

```
/real-estate/flats/flat[descendant-or-self::flat[@idFlat = 'f4a1']]  
/real-estate/flats/flat[descendant-or-self::flat/@idFlat = 'f4a1']  
/real-estate/flats/flat[.//@idFlat = 'f4a1']
```

```
//flat[@idFlat = 'f4a1']/ancestor-or-self::flat[last()]  
//flat[@idFlat = 'f4a1']/ancestor-or-self::flat[not(ancestor::flat)]
```

Incorrect:

```
/real-estate/flats/flat[descendant::flat/@idFlat = 'f4a1']  
/real-estate/flats/flat[.//flat/@idFlat = 'f4a1']  
/real-estate/flats/flat[//flat/@idFlat = 'f4a1']  
//flat[@idFlat = 'f4a1']/ancestor-or-self::flat[not(ancestor-or-self::flat)]
```

12

```
distinct-values(  
  //owner[  
    @idOwner = //property[  
      @idProperty = //flat[@comfort = 'A' or rate > 20000]/@property  
    ]/@owner  
  ]/address/country  
)  
//owner[  
  @idOwner = //property[  
    @idProperty = //flat[@comfort = 'A' or rate > 20000]/@property  
  ]/@owner  
]/address/country[not(. = preceding::owner/address/country)]
```

13

`data(/real-estate/flats/flat[not(flat)]/@comfort)`
`data(/flats/flat[not(flat)]/@comfort)`
`data(/flat[not(ancestor::flat)][not(flat)]/@comfort)`
`data(/flat[parent::flats])[not(flat)]/@comfort)`

Incorrect:

`data(/flat[not(ancestor-or-self::flat)][not(flat)]/@comfort)`

14

```
//*[text() = '46001'][not(starts-with(name(), "d"))][count(ancestor-or-self::* ) >= 5]  
//*[text() = '46001'][not(starts-with(name(), "d"))][count(ancestor-or-self::node()) >= 6]  
//*[text() = '46001'][not(starts-with(name(), "d"))][count(ancestor::* ) >= 4]  
//*[text() = '46001'][not(starts-with(name(), "d"))][count(ancestor::node()) >= 5]  
  
/*/*/*/*/*[text() = '46001'][not(starts-with(name(), "d"))]
```

15

```
//owner/address/country[not(. = preceding::owner/address/country)]/text()  
//owner/address/country[not(text() = preceding::owner/address/country/text())]/text()  
//owner/address/country[not(. = following::owner/address/country)]/text()
```

```
//owner/address/country[not(. = ../preceding-sibling::owner/address/country)]/text()  
//owner/address/country[not(. = ../following-sibling::owner/address/country)]/text()
```

```
//owner[not(address/country = preceding::owner/address/country)]/address/country/text()  
//owner[not(address/country = preceding-sibling::owner/address/country)]/address/country/text()  
//owner[not(address/country = following::owner/address/country)]/address/country/text()  
//owner[not(address/country = following-sibling::owner/address/country)]/address/country/text()
```

Incorrect:

```
//owner/address/country[. != preceding::owner/address/country]/text()  
//owner/address/country[not(. = preceding::country)]/text()  
//owner/address/country[not(. = following::country)]/text()
```

16A

```
//agency/employee[
  not(/lastName = //property[address/state = "New York"]/employee/lastName)
]
```

Incorrect:

```
//property/employee[...]
```

16B

Not possible in XPath 1.0!

Possibly correct?

```
//agency/employee[
  not( = //property[address/state = "New York"]/employee)
]
```

Incorrect:

```
//agency/employee[
  not(/firstName = //property[address/state = "New York"]/employee/firstName)
  and
  not(/lastName = //property[address/state = "New York"]/employee/lastName)
]
```


17

```
/real-estate/flats/flat[
  (@property = preceding-sibling::flat/@property)
  or
  (@property = following-sibling::flat/@property)
]/descendant-or-self::flat/name
```

Incorrect:

```
/real-estate/flats/flat[
  (@property = preceding-sibling::flat/@property)
  or
  (@property = following-sibling::flat/@property)
]//flat/name
```

17 (N = 3)

```
/real-estate/flats/flat[
  (@property = preceding-sibling::flat[
    @property = preceding-sibling::flat/@property
  ]/@property)
  or
  (@property = following-sibling::flat[
    @property = following-sibling::flat/@property
  ]/@property)
  or
  (
    (@property = preceding-sibling::flat/@property)
    and
    (@property = following-sibling::flat/@property)
  )
]/descendant-or-self::flat/name
```

17 (N > 2)

Not possible in XPath 1.0 for arbitrary N!

```
//flat[
  @property = //property[
    count(/real-estate/flats/flat[@property = ???]) >= N
  ]/@idProperty
]/name
```

```
//flat[
  count(/real-estate/flats/flat[@property = ???]) >= N
]/name
```

```
/real-estate/flats/flat[
  count(preceding-sibling::flat[@property = ???]) + count(following-sibling::flat[@property = ???])
  >= N - 1
]/descendant-or-self::flat/name
```