Each multiple-choice quiz problem is based on a "root question," from which the system generates different correct and incorrect choices each time you take the quiz. Thus, you can test yourself on the same material multiple times. We strongly urge you to continue testing on each topic until you complete the quiz with a perfect score at least once. Simply click the "Reset" button at the bottom of the page for a new variant of the quiz.

After submitting your selections, the system will score your quiz, and for incorrect answers will provide an "explanation" (sometimes for correct ones too). These explanations should help you get the right answer the next time around. To prevent rapid-fire guessing, the system enforces a minimum of 10 minutes between each submission of solutions.

Multiple Choice

(2/6 points)

[Q1] We're interested in well-formed XML that satisfies the following conditions:

- It has a root element "tasklist"
- The root element has 3 "task" subelements
- Each of the "task" subelements has an attribute named "name"
- The values of the "name" attributes for the 3 tasks are "eat", "drink", and "play"

Select, from the choices below, the well-formed XML that meets the above requirements.

```
<tasklist>
  <task name="eat"/>
  <task name="drink"/>
  <task name="play"/>
</tasklist>
```

```
<tasklist>
  <task name=eat/>
  <task name=drink/>
  <task name=play/>
</tasklist>
```

◉

```
<tasklist>
  <task name="eat">
  <task name="drink">
  <task name="play">
</tasklist>
```

✖

○

```
<tasklist>
  <task name="eat"/>
  <task name="drink"/>
  <task name="play"/>
<tasklist>
```

ANSWER-SELECTION FEEDBACK

In well-formed XML, all opening tags, such as the three "task" tags, must be closed.

[Q2] An XML document contains the following portion:

```
    <INFO>
        <ADDR>101 Maple St.</ADDR>
        <PHONE>555-1212</PHONE>
        <PHONE>555-4567</PHONE>
    </INFO>
```

Which of the following could be the INFO element specification in a DTD that the document matches?

○ <!ELEMENT INFO (NAME*,ADDR,PHONE+)>

- <!ELEMENT INFO (ADDR?,PHONE?)>
- <!ELEMENT INFO (ADDR+,PHONE+,MANAGER)>
- <!ELEMENT INFO (PHONE*,ADDR)> ✖

ANSWER-SELECTION FEEDBACK

This declaration requires the PHONE subelements to appear before the ADDR subelement.

[Q3] An XML document contains the following portion:

```
<EMP name = "Kermit">
    <ADDR>123 Sesame St.</ADDR>
    <PHONE type = "cell">555-1212</PHONE>
</EMP>
```

Which of the following could NOT be part of a DTD that the document matches? Note that there can be multiple ATTLIST declarations for a single element type; do not assume the only attributes allowed for an element type are the ones shown in the answer choice.

- <!ATTLIST PHONE type IDREF #IMPLIED> ✖
- <!ATTLIST EMP name ID #IMPLIED>
- <!ATTLIST ADDR zip CDATA #REQUIRED>
- <!ATTLIST ADDR zip CDATA #IMPLIED>

ANSWER-SELECTION FEEDBACK

There is nothing wrong with this declaration. It says that PHONE elements may have a type attribute, but the only example of such an element that we see does indeed have this attribute.

[Q4] Here is a DTD:

```
<!DOCTYPE A [
    <!ELEMENT A (B+, C)>
    <!ELEMENT B (#PCDATA)>
    <!ELEMENT C (B?, D)>
    <!ELEMENT D (#PCDATA)>
]>
```

Which of the following sequences of opening and closing tags matches this DTD? Note: In actual XML, opening and closing tags would be enclosed in angle brackets, and some elements might have text subelements. This quiz focuses on the element sequencing and interleaving specified by the DTD.

- A C B /B D /D /C /A
- A C D /D /C /A
- A B /B B /B C B /B D /D /C /A ✔
- A B /B B /B C B /B B /B D /D /C /A

[Q5] Here is an XML DTD:

```
<!DOCTYPE meal [
    <!ELEMENT meal (person*,food*,eats*)>
    <!ELEMENT person EMPTY>
    <!ELEMENT food EMPTY>
    <!ELEMENT eats EMPTY>
    <!ATTLIST person name ID #REQUIRED>
    <!ATTLIST food name ID #REQUIRED>
    <!ATTLIST eats diner IDREF #REQUIRED dish IDREF
#REQUIRED>
]>
```

Which of the following documents match the DTD?

1.

```
<meal>
  <person name="Alice"/>
  <food name="salad"/>
  <eats diner="Alice" dish="salad"/>
  <person name="Bob"/>
  <food name="salad"/>
  <eats diner="Bob" dish="salad"/>
  <person name="Carol"/>
  <food name="sandwich"/>
  <eats diner="Carol" dish="sandwich"/>
</meal>
```

2.

```
<meal>
  <person name="Alice"/>
  <person name="Bob"/>
  <person name="Carol"/>
  <person name="Dave"/>
  <food name="salad"/>
  <food name="turkey"/>
  <food name="sandwich"/>
  <eats diner="Alice" dish="turkey"/>
  <eats diner="Bob" dish="salad"/>
  <eats diner="turkey" dish="Dave"/>
</meal>
```

3.

```
<meal>
  <person name="Alice"/>
  <person name="Bob"/>
  <food name="salad"/>
  <eats diner="Alice" dish="food"/>
  <eats diner="Bob" dish="food"/>
</meal>
```

- ○ only the third
- ◉ only the first ✖
- ○ only the second and third
- ○ only the second

ANSWER-SELECTION FEEDBACK

Focus on the ID and IDREF attributes: A valid document needs to have unique values across ID attributes. An IDREF attribute can refer to any existing ID attribute value.

[Q6] Study the following XML Schema specification:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="fname" type="xs:string"/>
        <xs:element name="initial" type="xs:string"
            minOccurs="0"/>
        <xs:element name="lname" type="xs:string"/>
        <xs:element name="address" type="xs:string"
            maxOccurs="2"/>
        <xs:choice>
          <xs:element name="major" type="xs:string"/>
          <xs:element name="minor" type="xs:string"
              minOccurs="2" maxOccurs="2"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Select, from the choices below, the XML that is valid according to the XML Schema specification above.

○

```
  <person>
    <lname>Public</lname>
    <fname>John</fname>
    <initial>Q</initial>
    <title>Sr.</title>
    <address>123 Public Avenue, Seattle, WA
98001</address>
    <major>Computer Science</major>
  </person>
```

◉

```
<person>
  <fname>John</fname>
  <lname>Public</lname>
  <address>123 Public Avenue, Seattle, WA
98001</address>
  <major>Computer Science</major>
</person>
```

✔

```
<person>
  <fname>John</fname>
  <initial>Q.</initial>
  <lname>Public</lname>
  <address>123 Public Avenue</address>
  <address>Seattle</address>
  <address>WA 98001</address>
  <minor>History</minor>
</person>
```

```
<person>
  <fname>John</fname>
  <lname>Public</lname>
  <initial>Q.</initial>
  <address>123 Public Avenue, Seattle, WA
98001</address>
  <minor>History</minor>
</person>
```