

ESERCIZIO 1

Scrivere una descrizione testuale ed un documento XML valido per il linguaggio di Markup specificato dal seguente dtd

```
<!ELEMENT recipes (recipe+)>
<!ELEMENT recipe (recipe_head, recipe_body, recipe_footer?, document_info)> <!ELEMENT
recipe_head (recipe_name, recipe_author?, meal_type)> <!ELEMENT recipe_name
(#PCDATA)>
<!ELEMENT recipe_author (#PCDATA)>
<!ELEMENT meal_type (#PCDATA)>
<!ELEMENT recipe_body (ingredients, directions)>
<!ELEMENT ingredients (ingredient+)>
<!ELEMENT ingredient (#PCDATA)>
<!ELEMENT directions (direction)+ >
<!ELEMENT direction (#PCDATA)>
<!ELEMENT recipe_footer (serving?, preparation_time?, cooking_time?)> <!ELEMENT
serving (#PCDATA)>
<!ELEMENT preparation_time (#PCDATA)>
<!ELEMENT cooking_time (#PCDATA)>
<!ELEMENT document_info (document_author, date_updated, source)> <!ELEMENT
document_author (#PCDATA)>
<!ELEMENT date_updated (#PCDATA)>
<!ELEMENT source (#PCDATA)>
```

ESERCIZIO 2

Si progetti il DTD e si scriva un XML valido per un linguaggio di markup per lettere. Ogni lettera ha un mittente, una data, un destinatario, un oggetto, una forma cortese di saluto, un corpo, una chiusura, una firma. Il corpo della lettera ha quanto meno un paragrafo.

ESERCIZIO 3

Scrivere una DTD che formalizzi le seguenti regole:

1. sono ammessi solo gli elementi <a>, , <c>, <d>, <e>;
2. <a> è l'elemento radice;
3. <a> può contenere solo elementi di tipo o di tipo <c>, e deve contenere almeno un elemento di tipo ;
4. può contenere solo elementi di tipo <d> o di tipo <e>, deve contenere almeno un elemento <e>, e tutti gli elementi <d> devono precedere gli elementi <e>;
5. <c> può contenere solo elementi di tipo <d> o di tipo <e>, deve contenere almeno due elementi <d>, e tutti gli elementi <d> devono precedere gli elementi <e>;
6. gli elementi <d> possono contenere ogni tipo di elemento (compreso #PCDATA);
7. <e> è un elemento vuoto;
8. <a> ha un attributo x obbligatorio di tipo CDATA e un attributo y che può assumere solo i valori 0, 1, o 2, e 0 è il valore di default;
9. <e> ha un attributo z obbligatorio di tipo CDATA.

ESERCIZIO 4

Dato il seguente documento XML: 3

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE a [
  <!ELEMENT a (b | c | d)*>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c EMPTY>
  <!ELEMENT d (#PCDATA | a)*>
  <!ATTLIST a val CDATA #REQUIRED>
  <!ATTLIST b val CDATA #IMPLIED>
]>
<a val="v1">
<b val="1">riga 1</b>
<b val="pippo">riga 2</b>
<c/>
<b x="3">riga 3</b>
<d>riga 4
<a val="v2">
<c val="xyz"/>
</a>
</d>
</a>
```

dire se il documento è valido. In caso contrario, evidenziare le violazioni della DTD da parte del documento.

ESERCIZIO 5

Dato il seguente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE a [
  <!ELEMENT a (b | c | d)*>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c EMPTY>
  <!ELEMENT d (#PCDATA | a)*>
  <!ATTLIST a val CDATA #REQUIRED>
  <!ATTLIST b id ID #IMPLIED>
  <!ATTLIST c link IDREF #REQUIRED>
]>
<a val="Text">
<b id="id01">Text</b>
<b id="id02">Text</b>
<c link="id03"/>
<b id="id03">Text</b>
<d>Text
<a val="Text">
<c link="id01"/>
</a>
</d>
</a>
```

dire se il documento è valido. In caso contrario, evidenziare le violazioni della DTD da parte del documento.