

## Gruppo 2

**Attenzione:** una volta letto il presente testo è *obbligatorio* consegnare alla scadenza quanto elaborato, indipendentemente dal fatto che lo si consideri adeguato o meno.

Entro il termine stabilito si *deve* inviare via email dentro un unico file compresso, il cui nome sia “ProgettoLC parte4 Gruppo 2”, tutti (e soli) i sorgenti sviluppati, in modo che sia possibile verificare il funzionamento di quanto realizzato. In tale file compresso *non* devono comparire file oggetto e/o binari o qualsiasi altro file che viene generato a partire dai sorgenti. Sempre in detto file va inoltre inclusa una relazione in formato PDF dal nome “ProgettoLC parte4 Gruppo 2 Relazione”. La relazione può contenere immagini passate a scanner di grafici o figure fatte a mano.

Si richiede:

- una descrizione dettagliata di tutte le tecniche **non**-standard impiegate (le tecniche standard imparate a lezione non vanno descritte).
- una descrizione delle assunzioni fatte riguardo alla specifica, sia relativamente a scelte non previste espressamente dalla specifica stessa, che a scelte in contrasto a quanto previsto (con relative motivazioni).

Non è assolutamente utile perdere tempo per includere nella relazione il testo dei vari esercizi o un suo riassunto o una qualsiasi rielaborazione, incluso descrizioni del problema da risolvere. Ci si deve concentrare solo sulla descrizione della soluzione e delle eventuali variazioni rispetto a quanto richiesto.

- di fornire opportuni Makefile (compliant rispetto allo standard [GNU make](#)) per
  - poter generare automaticamente dai sorgenti i files necessari all’esecuzione (lanciando **make**);
  - far automaticamente una demo (lanciando **make demo**).
- di implementare la soluzione in C o C++ o Java, utilizzando BNFC o Bison/Flex o CUP/JLex o strumenti analoghi (che generino il sorgente di parser/lexer a partire da grammatica/espressioni regolari).

### Esercizio

---

Si consideri un file di configurazione “alla XML” composto da varie sezioni, identificate da tag **section** e il cui nome è indicato da un attributo **name**; ogni sezione contiene tag **field** che assegnano valori a variabili indicate dagli attributi **name**. Etichette e variabili sono stringhe alfanumeriche (inizianti con un carattere), mentre i valori possono essere numeri interi, booleani, stringhe tra doppi apici o altre variabili (precedute da “\$” ed un eventuale prefisso “*nomesezione*.” per variabili non locali).

All’inizio del file possono esserci delle direttive `<import> file</import>` per importare il contenuto di un altro file. L’implementazione non deve limitarsi a riconoscere sintatticamente queste direttive, ma deve effettivamente processare il file. Si scelga di realizzare questa funzionalità direttamente nel lexer o nel parser a seconda delle preferenze.

Commenti (da ignorarsi) da “#” a fine linea.

Un esempio è il seguente:

```
<import>../glob/file1.cnf</import>
<section name = # questo e' un commento
  nomesez1 >
# questo e' un commento
  <field name=var1 > 3</field >
  <field name =var2>"non dire gatto"  </field>
  <field name= var3>$var1</field >
</section ># questo e' un commento

<section  name=nomesez2>
  <field
    name=
      miao
    >true</field>
  <field name=var1>false</field>
```

# Progetto di Linguaggi e Compilatori – Parte 4

## A.A. 2019-20

```
</section>
```

```
<section name=nomesez3>  
  <field name=miao>$nomesez1.var3</field>  
  <field name=var1>false</field>  
</section>
```

Si definisca la grammatica di tale sintassi.

Si implementi un parser (con relativo lexer) che dato un file di configurazione valido, costruisca una struttura dati (ad esempio mediante una lista di liste che associ sezioni a liste etichetta-valore) contenente i valori associati ad ogni variabile di ogni sezione. La definizione della stessa variabile in sezioni diverse è legittima, mentre la ridefinizione nella stessa sezione deve essere segnalata come warning. In caso di ripetizione di una sezione invece deve essere segnalato errore.

Si implementi inoltre un “pretty-printer” che, quando invocato su una qualsiasi modifica dell’output del parser, riscriva il file di configurazione con i valori aggiornati, perdendo eventualmente la struttura di assegnamenti a variabile con altre variabili, ma mantenendo gli eventuali commenti presenti.