

## **Corso di Laboratorio di linguaggi di programmazione e compilatori**

# Compito scritto di esame

3 settembre 2004

Docente: Prof. Simone Brunozi  
email: [labcompilatori@wedoit.us](mailto:labcompilatori@wedoit.us)  
www: [www.wedoit.us/labcompilatori/](http://www.wedoit.us/labcompilatori/)

file di riferimento:  
esame\_scritto\_\_2004-09-03\_v1.0

URI di riferimento:  
non disponibile

Questo documento è rilasciato sotto licenza Creative Commons ([www.creativecommons.org](http://www.creativecommons.org)),  
il testo della licenza è reperibile agli URI  
<http://creativecommons.org/licenses/by-sa/1.0/>  
<http://creativecommons.org/licenses/by-sa/1.0/legalcode>

This work is licensed under the Creative Commons Attribution-ShareAlike License.  
To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/>  
or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Attribution



Share-Alike



E' possibile reperire materiale licenziato con Creative Commons presso l'URI  
<http://commoncontent.org/>

Durata del compito scritto: **120 minuti**

Punteggi delle domande:

<b>domanda</b>	<b>punteggio</b>	<b>domanda</b>	<b>punteggio</b>
01	6,0	06	3,0
02	3,0	07	7,0
03	3,0	08	5,0
04	3,0	09	2,0
05	8,0		

**Totale: 40,0 punti**

**COGNOME e nome:** \_\_\_\_\_ / \_\_\_\_\_

**MATRICOLA:** \_ \_ \_ \_ \_

scrivere in stampatello con una calligrafia ben comprensibile

**Domanda 01 [punti 6,0]**

Indicare la risposta corretta, e indicare a quale elemento si riferisce la risposta di volta in volta. Non basate le vostre risposte su un meccanismo ad esclusione, non date per scontato che esistano risposte giuste, o che non esistano ripetizioni e/o "trabocchetti".

**Structure editor**     **Pretty printer**     **Interpreter**     **Text formatter**

a	Analizza un programma e lo ripropone in una maniera in cui la struttura sia esplicitamente visibile.
b	Analizza un sorgente e lo ripropone in una maniera in cui la struttura sia esplicitamente visibile.
c	Analizza un programma e lo processa inserendo parole chiave in maniera che la struttura sia esplicitamente visibile.
d	Analizza un sorgente e lo processa inserendo parole chiave in maniera che la struttura sia esplicitamente annidata.

**Structure editor**     **Pretty printer**     **Interpreter**     **Text formatter**

a	Prende in input dei comandi del sorgente per costruire un codice intermedio, analizza il testo del programma creando una struttura gerarchica del codice obiettivo. Può verificare la correttezza formale dell'output o fornire parole chiave del linguaggio durante la digitazione.
b	Prende in input dei comandi per costruire un sorgente, analizza il testo del programma creando una struttura gerarchica del sorgente. Può verificare la correttezza formale dell'input o fornire automaticamente parole chiave del linguaggio durante la digitazione.
c	Prende in input dei comandi del sorgente per costruire un codice intermedio, analizza il testo del programma creando una struttura gerarchica del codice intermedio. Può verificare la correttezza formale dell'input o fornire parole chiave del linguaggio durante la digitazione.
d	Prende in input dei sorgenti in un linguaggio dato, analizza il testo del programma creando una struttura gerarchica del linguaggio. Può verificare la correttezza formale dell'output o fornire automaticamente parole chiave del linguaggio durante la digitazione.
e	Prende in input dei comandi per costruire un sorgente, analizza il testo del programma creando una struttura gerarchica del sorgente. Può verificare la correttezza formale dell'output.

**Structure editor**     **Pretty printer**     **Interpreter**     **Text formatter**

a	Simula l'esecuzione delle operazioni indicate nel codice sorgente. Utilizzato anche per eseguire programmi a basso livello come ASI (Array Scripting Interpreter), o comandi di script.
b	Esegue le operazioni indicate nel codice sorgente. Utilizzati anche per eseguire programmi a basso livello come APL (Array Processing Language), oppure comandi di script, come Python, PHP, ASP.
c	Esegue le operazioni indicate nel codice sorgente. Utilizzati anche per eseguire programmi ad alto livello come APL (Array Processing Language), oppure comandi di script.
d	Prende in input dei comandi per costruire un sorgente, analizza il testo del programma creando una struttura gerarchica. Può verificare la correttezza formale dell'output. Utilizzati anche per eseguire programmi ad alto livello come APX.

**Structure editor**     **Pretty printer**     **Interpreter**     **Text formatter**

a	Analizza un sorgente e tenta di scoprire potenziali errori senza farlo eseguire, ad es. cercando parti di programma che non vengono mai eseguite, o quelle in cui una variabile può essere usata prima di essere stata definita.
b	Analizza un sorgente e tenta di scoprire potenziali errori senza farlo eseguire, ad es. cercando parti di programma che non vengono mai eseguite, o quelle in cui una variabile può essere usata prima di essere stata definita. Evidenzia anche errori logici come ad esempio tentare di usare una variabile reale come un puntatore (type-checking).
c	Prende in input uno stream binario, costituito da comandi che indicano paragrafi, figure, o strutture matematiche. Da questo input viene fatta una sintesi, e la successiva sintesi produce un output che viene utilizzato per visualizzare il testo.
d	Prende in input testo e comandi che indicano paragrafi, figure, o strutture matematiche. Da questo input viene fatta una analisi, e la successiva sintesi produce un output che viene utilizzato per visualizzare il testo.

**Domanda 02 [punti 3,0]**

Fornire la traduzione italiana delle seguenti parole in inglese, e nel caso degli ACRONIMI fornire il loro significato esteso (es. GPL= GNU Public License).

query: \_\_\_\_\_ Match: \_\_\_\_\_ Pattern: \_\_\_\_\_  
 Token: \_\_\_\_\_ set: \_\_\_\_\_ target: \_\_\_\_\_  
 EBNF: E \_\_\_\_\_ B \_\_\_\_\_ N \_\_\_\_\_ F \_\_\_\_\_  
 SUS: S \_\_\_\_\_ U \_\_\_\_\_ S \_\_\_\_\_  
 PCRE: P \_\_\_\_\_ C \_\_\_\_\_ R \_\_\_\_\_ E \_\_\_\_\_  
 LIFO: L \_\_\_\_\_ I \_\_\_\_\_ F \_\_\_\_\_ O \_\_\_\_\_

GREP: search G \_\_\_\_\_ for a R \_\_\_\_\_ E \_\_\_\_\_ and P \_\_\_\_\_ lines where instances are found.



n.	stato precedente	simbolo letto	simbolo scritto	movimento	nuovo stato
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

2) Mostrare l'aspetto del nastro bidimensionale iniziale dopo 15 e dopo 25 azioni.

Inizio	Dopo 15 azioni	Dopo 25 azioni

Lo schema è replicato per permettervi di riscriverlo in caso di errore.





