

1. L'assembler e il linguaggio macchina sono la stessa cosa?
2. Un programma scritto in assembler per il processore Intel Pentium III può essere utilizzato per il processore AMD Athlon 64?
3. L'assembler consente di utilizzare dei nomi simbolici per individuare operazioni e variabili?
4. Il compilatore produce codice oggetto a partire da codice sorgente?
5. In generale, i programmi applicativi per poter funzionare hanno bisogno della presenza di un sistema operativo sottostante?
6. Due linguaggi di programmazione diversi possono avere due alfabeti diversi?
7. Due linguaggi di programmazione diversi possono avere lo stesso alfabeto?
8. La semantica di un linguaggio di programmazione può essere definita tramite il formalismo di Bakus-Naur (BNF)?
9. In BNF, l'alfabeto terminale e l'alfabeto non terminale sono due insiemi disgiunti?
10. Si consideri una grammatica BNF e una frase scritta utilizzando i simboli terminali del relativo alfabeto. Se non esiste un albero sintattico di derivazione di tale frase mediante tale grammatica, significa...
 - 10.1. che la frase non è corretta?
 - 10.2. che la frase è corretta ma è priva di significato?
 - 10.3. non è possibile! esiste sempre e comunque un albero sintattico di derivazione di una data frase mediante una data grammatica?
11. La regola di produzione BNF
 $\langle \text{cifra} \rangle ::= 0 \mid 1$
può essere utilizzata per sostituire, mediante processo di derivazione, al simbolo non terminale $\langle \text{cifra} \rangle$ il simbolo 0100101?
12. Data la seguente grammatica:
V: { 0, 1, a, b, c, d, <, >, = }
N: { $\langle \text{expr} \rangle$, $\langle \text{var} \rangle$, $\langle \text{value} \rangle$, $\langle \text{digit} \rangle$, $\langle \text{op} \rangle$, $\langle \text{assign} \rangle$ }
S: $\langle \text{assign} \rangle$
P:
 $\langle \text{expr} \rangle ::= \langle \text{value} \rangle \mid \langle \text{logical_expr} \rangle$
 $\langle \text{logical_expr} \rangle ::= \langle \text{value} \rangle \langle \text{op} \rangle \langle \text{value} \rangle \mid \langle \text{var} \rangle \langle \text{op} \rangle \langle \text{value} \rangle$
 $\langle \text{value} \rangle ::= 0 \mid 1 \mid 0\langle \text{value} \rangle \mid 1\langle \text{value} \rangle$
 $\langle \text{op} \rangle ::= < \mid >$
 $\langle \text{var} \rangle ::= a \mid b \mid c \mid d$
 $\langle \text{assign} \rangle ::= \langle \text{var} \rangle = \langle \text{expr} \rangle$
 - 12.1. Esiste un albero sintattico per derivare la frase "a = b" ?
 - 12.2. Esiste un albero sintattico per derivare la frase "a = b < 11010" ?
 - 12.3. Esiste un albero sintattico per derivare la frase "b < 11010" ?
 - 12.4. Esiste un albero sintattico per derivare la frase "a = b < c" ?
 - 12.5. Esiste un albero sintattico per derivare la frase "1 = b < c" ?
 - 12.6. Esiste un albero sintattico per derivare la frase "a = b = 11010" ?
 - 12.7. La frase "a = b < c" è corretta?
 - 12.8. La frase "a = b < 11010" è corretta?
13. Si scriva una grammatica BNF che definisce le frasi del tipo "somme di due numeri binari": ad esempio, 11010 + 100 + 100101 + 10010, e si faccia

in modo che i numeri binari che cominciano per '0' non siano considerati corretti. Ad esempio, **11010 + 0100** non deve essere una frase corretta.

14. In un linguaggio di programmazione tipato, è possibile che una variabile appartenga a più di un tipo?
15. La "portabilità" ha a che fare con la possibilità di rimanere indipendenti dall'architettura?
16. In un programma C, dichiarazione e istruzioni sono due parti distinte? è possibile dichiarare una variabile tra due istruzioni di assegnamento?
17. I tipi scalari in C sono tipi di dato non atomici, ovvero logicamente scomponibili in altri componenti?
18. Quali sono i quattro attributi di una variabile?
19. Possono esistere in C due variabili con lo stesso nome e lo stesso scope?
20. I tipi di dato primitivi in C sono quelli definiti dall'utente?
21. Come si definisce in C una variabile di tipo intero senza segno?
22. In C esiste un tipo di dato che definisce i numeri con la virgola?
23. In C esiste un tipo di dato che definisce il tipo booleano?
24. Si considerino le seguenti istruzioni:

```
int x=2;  
char A='a', B='f';
```

- 24.1. Qual è il risultato dell'espressione **A-B**? di che tipo è?
- 24.2. Qual è il risultato dell'espressione **A+x**? di che tipo è?

25. Si considerino le seguenti istruzioni:

```
typedef int x;  
x A, B;  
int p;  
char k;
```

- 25.1. È corretto scrivere **A+B**? Di che tipo è il risultato?
- 25.2. È corretto scrivere **A+p**? Di che tipo è il risultato?
- 25.3. È corretto scrivere **A+k**? Di che tipo è il risultato?