

Tempo a disposizione: ore 2.

1. È dato l'alfabeto (di terminali)  $T = \{\bullet, \diamond, 2, 3, 4\}$  e la grammatica  $G = (\{A, B, C\}, T, A, P)$  con  $P$  dato da

$$\begin{aligned} A &::= A \bullet B \mid B \\ B &::= B \diamond C \mid C \\ C &::= 2 \mid 3 \mid 4 \end{aligned}$$

Si consideri ora la stringa  $s = 2 \bullet 3 \diamond 4$ .

- (i)  $s$  appartiene al linguaggio generato da  $G$ ? In caso positivo, si dia un albero di derivazione; in caso negativo, si dia una breve motivazione.  
(ii) È possibile utilizzare la grammatica  $G$  per compilare in modo univoco  $s$ ? In caso positivo, qual è il suo valore, se  $\bullet$  è interpretato come prodotto e  $\diamond$  come somma? In caso negativo, si modifichi opportunamente la grammatica.
2. L'esecuzione del seguente frammento di codice su una certa implementazione risulta nella stampa del valore 9.

```
int v[10];
int x = 4;
for (int i=0, i<10, i++) v[i]=i;
v[x] = v[x++] + v[x++];
write (v[x]);
```

Si fornisca una possibile spiegazione.

3. In un linguaggio con scope statico, implementato mediante catena statica, si prenda in considerazione l'operazione di "accesso ad una variabile non locale" nel corpo di una funzione; nel seguito indicheremo tale variabile con  $x$ . A quale dei seguenti parametri è proporzionale il tempo necessario all'esecuzione di tale operazione? Si motivi brevemente la risposta.

(i) Il numero  $nv$  di variabili presenti nel programma. (ii) Il numero di variabili  $nvb$  presenti nei blocchi compresi tra quello di dichiarazione di  $x$  e quello in cui vi si accede. (iii) Il numero di blocchi  $nb$  sintatticamente compresi tra quello di dichiarazione di  $x$  e quello in cui vi si accede; (iv)  $nb + nvb$ . (v) Il tempo è costante, e quindi indipendente da questi parametri. (vi) Nessuna delle risposte precedenti.

4. Si dica cosa stampa il seguente frammento di programma, in uno pseudolinguaggio con scope statico e passaggio dei parametri per riferimento (indicato con **reference**) e per valore (default).

```
{int x = 0;
  int f(reference int y) {
    int x=2;
    y=y+1;
    return g(y);
  }
  int g(int y){
    int h(int y){
      int x = 3;
      return f(y);
    }
    if (y==1) return h(x);
    else return x+y;
  }
  write (f(x));
}
```

5. In riferimento al frammento dell'esercizio precedente, si supponga che lo scope statico sia gestito mediante display. Si dia graficamente la situazione del display e della pila dei record di attivazione al momento in cui il controllo entra per la *seconda* volta nella funzione  $g$ . Per ogni record di attivazione si dia soltanto: il valore delle variabili locali; il valore del campo destinato a salvare il display.
6. Si considerino le seguenti dichiarazioni di un record con campi varianti e di un array multidimensionale:

```

struct Impiegato{
    char nome[6];
    int matricola;
    int pensionato;
    union{
        int ultimo_anno_lavorativo;
        struct{
            char sigla_azienda[2];
            int codice;
        } non_in_pensione;
    } campivarianti;
};
Impiegato Pippo;
int A[10][10];

```

Sappiamo che: un `int` è memorizzato su 4 byte; un `char` su un byte; l'array è memorizzato in ordine di riga, con indirizzi di memoria crescenti (cioè se un elemento è all'indirizzo  $i$ , il successivo è a  $i + 4$  ecc.); siamo in un'architettura a 32 bit con allineamento alla parola; l'array `A` è memorizzato immediatamente dopo la variabile `Pippo` che è memorizzata all'inizio del record di attivazione. Qual'è l'offset (in parole) dell'elemento `A[2][5]` rispetto all'inizio del record di attivazione ?

7. Si consideri la seguente dichiarazione

```

int x = 0;
void->int F (int y) {
    int z = y+1;
    int g (int n) {
        return n+z+1;
    }
    return g;
}
void->int gg = F(3);
x = gg(2);

```

dove `void->int` denota il tipo "funzione da void a intero". Si dica dove sono memorizzate le variabili `x`, `y` e `z` motivando la risposta.

8. **Solo per il corso AL** È dato il seguente programma Prolog (ricordiamo che `X` e `Y` sono variabili mentre `a`, `b` e `b` sono costanti):

```

p(a):- p(a).
p(b):- p(c),r(b).
p(X):- r(a).
r(b):-r(b).
r(c).

```

Si dica, giustificando la risposta se il goal `p(b)` termina con successo, fallisce o produce una computazione infinita.

9. **solo MZ**

Si considerino le seguenti definizioni di classe in Java:

```

class A{
    int x;
    int f (int y){return y+1;}
}
class B extends A{
    int y;
    void g (int z){...}
}
class C extends B{
    int f (int y){return y+2;}
}

```

Si supponga che la gerarchia delle classi sia implementata mediante `vtable`. Qual è la struttura della `vtable` di `C`?