

Tempo a disposizione: ore 2.

1. Si descrivano sinteticamente le principali modalità di realizzazione di una macchina astratta ed i relativi vantaggi/svantaggi.
2. Si illustri una tecnica per la gestione a run-time dello scoping dinamico.
3. Si presenti *sinteticamente* la tecnica del *reference count* per ovviare al problema delle *dangling references*; si discutano vantaggi e svantaggi.

4. È data la grammatica $G = (\{S, T, a, b, +, *\}, \{a, b, +, *\}, S, P)$ dove l'insieme P delle produzioni è costituito da

$$\begin{aligned} S &\rightarrow S * S \mid T \\ T &\rightarrow T + T \mid (S) \mid a \mid b \end{aligned}$$

Si dia una derivazione (o un albero di derivazione) per ciascuna delle stringhe seguenti: $a + b * a + b$ e $a + (b * a) + b$. Si dica qual è la precedenza indotta dalla grammatica tra gli operatori $+$ e $*$.

5. Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usi scoping statico (regola dello scope annidato più vicino) e dove la primitiva `read(Y)` permette di leggere nella variabile `Y` un intero dall'input standard, mentre `write(X)` permette di stampare il valore della variabile `X`.

```
...
int X = 0;
int Y;

void pippo() {
    X++;
}

void pluto() {
    X++;
    pippo;
}

read(Y);

if Y > 0 then { int X = 5;
                pluto();
            }
            else { pluto();
            }

write(X);
```

Si dica quali sono (o qual è) i valori stampati.

6. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che ammette parametri per *nome*.

```
int X = 2;
void foo (name int Y)
    {X++;
    write(Y);
    X++;
    }
foo(X+1);
write(X);
```

7. Si dica cosa viene stampato dall'esecuzione della classe Java SubClasse, definita come segue:

```
class SuperClasse {
    public String s = "SUPER";
    public void stampa() {
        System.out.println(s);
    }
}

class SubClasse extends SuperClasse {
    public String s = "SUB";
    public void stampa() {
        System.out.println(s);
    }
    public static void main(String[] args) {
        SubClasse sub = new SubClasse();
        SuperClasse sup = sub;
        sup.stampa();
        sub.stampa();
        System.out.println(sup.s);
        System.out.println(sub.s);
    }
}
```

8. Si considerino le seguenti dichiarazioni (Pascal):

```
type stringa = packed array [1..16] of char;
type punt_stringa = ^stringa;
type persona = record
    nome = stringa;
    case studente: Boolean of
        true: (matricola: integer);
        false: (codicefiscale: punt_stringa)
end;
```

e si supponga che la variabile C contiene il puntatore alla stringa "CODICEPIPP0". Si descriva il layout di memoria dopo ognuna delle seguenti istruzioni:

```
...
var pippo persona;

pippo.studente:= true;
pippo.matricola := 223344;
pippo.studente:= true;
pippo.codicefiscale := C;
```