

Tempo a disposizione: ore 2.

1. È data la grammatica $G = (\{A, B\}, \{a, b\}, A, P)$ dove P sono le produzioni

$$\begin{aligned} A &::= aAa \mid bAb \mid B \\ B &::= aBb \mid bBa \mid \epsilon \end{aligned}$$

(ϵ indica la stringa vuota). Per ciascuna delle affermazioni seguenti si dica se è vera/falsa. Per le false si dica anche perché lo sono.

- Nel linguaggio generato da G , in ogni stringa il numero di a coincide con il numero di b ;
 - G non è ambigua;
 - Se $B \Rightarrow_* w$, con $w \in \{a, b\}^*$, in w il numero di a coincide con il numero di b ;
 - Nel linguaggio generato da G , ogni stringa contiene un numero pari di simboli.
2. In un libro di introduzione alla programmazione si trova scritto: “Java è un linguaggio interpretato e pertanto è meno efficiente di C, che è un linguaggio compilato”. Si commenti sinteticamente questa frase alla luce del corso di Paradigmi.
3. Si dica cosa stampa il seguente frammento di codice, in un linguaggio con valutazione da sinistra a destra, scope statico e passaggio per nome:

```
int x = 1;
int A[8];
int i;
for (i=0, i<8, i++) A[i]=i;

int fie(int name w,z){
    int x = 3;
    write( (w++) + z + (x++) );
}

fie(x,A[x]);
write(x);
}
```

4. Cosa stampa il frammento dell'esercizio precedente se il passaggio dei parametri avviene *per riferimento*?
5. Si consideri la seguente definizione di tipo record:

```
type S = struct{
    int x;
    int y;
};
```

Si supponga che un `int` sia memorizzato su 2 byte, su un'architettura a 16 bit con allineamento alla parola. In un blocco viene dichiarato un vettore:

```
S A[10];
```

Indicando con `PRDA` il puntatore all'RdA di tale blocco, e con `ofst` l'offset tra il valore di `PRDA` e l'indirizzo iniziale di memorizzazione di `A`, si dia l'espressione per il calcolo dell'indirizzo dell'elemento `A[5].y` (indicare tutte le costanti in decimale).

6. Un certo linguaggio di programmazione ha una gestione statica della memoria. Quali delle seguenti affermazioni *sono certamente vere*? Per ciascuna di esse si commenti brevissimamente.
- Non può calcolare tutte le funzioni calcolabili (ovvero: non è Turing-completo).
 - Non può avere funzioni ricorsive.
 - Non può avere funzioni ricorsive in coda
 - Non ha garbage collector.

7. Si dica cosa stampa il seguente frammento in un linguaggio con scope statico, passaggio per valore ed eccezioni:

```
int a = 5;

void g(){
    write (a);
    throw X;
}

void f(int x){
    int a = 9;
    if (x==0) g();
    else try{ g(); } catch X { a--; f(0); }
}

try {f(1);} catch X {a--;}
write(a);
```

8. Cosa stampa il seguente codice Java?

```
class A{
    int x;
    void f(){
        x++;
    }
}

class B extends A{
    int x;
    void f(){
        x--;
    }
}

A a = new B();
B b = (B) a;
a.x = 10;
b.x = 20;
a.f();
b.f();
System.out.println(a.x);
System.out.println(b.x);
```