

Tempo a disposizione: ore 2.

1. Si consideri la seguente grammatica  $G = (\{S, T\}, \{a, b\}, S, P)$  dove  $P$  è dato dalle seguenti produzioni:

$$\begin{aligned} S & ::= SaS \mid TbT \mid T \\ T & ::= TaT \mid S \mid a \end{aligned}$$

Si mostri che  $G$  è ambigua.

2. Si descriva brevemente cosa è una chiusura e a che cosa serve.
3. Considerando un qualsiasi (pseudo)linguaggio di programmazione implementato in modo compilativo, si diano esempi di:
- (1) un errore lessicale (rilevato dallo scanner);
  - (2) un errore sintattico (rilevato dal parser);
  - (3) un errore di semantica statica;
  - (4) un errore di semantica dinamica; in questo caso il codice prodotto dal compilatore può rilevare l'errore? Motivare la risposta.
4. Si considerino le seguenti dichiarazioni in Java:

```
class A{
    int x = 5;
    int f(int n){return x+2*n;}
}
class B extends A{
    int x = 2;
    int f(int n){return x+n;}
}
A a = new B();
```

Nello scope di tali dichiarazioni, qual è il valore dell'espressione  $a.f(1)$  ?

5. Si consideri il seguente frammento di codice in uno pseudo-linguaggio che ammetta passaggio dei parametri per riferimento e per nome.

```
int[] V = new int[6];
int n=0;
```

```
int f (name int x) {
    int y;
    y = x++;
    y = x++;
    return x }
}
```

```
void foo(reference int x,name int y){
    x++; y++; x++; y++;}
```

```
V[0]=V[1]=V[2]=V[3]=V[4]=V[5] = V[6]= 1;
```

```
foo(V[f(n)], V[f(n)]);
```

Si dia lo stato del vettore  $V$  al termine dell'esecuzione del codice esposto (si ricordi che un comando della forma `return w++`; restituisce il valore corrente di  $w$  e poi incrementa  $w$  di uno).

6. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope statico:

```
int a = 1;
int f (reference int x){
    int a = 15;
    int g (reference int x){
        return f(x)+a;
    }
    if (x==0) then return 1;
```

```

    else {
        x=x-1;
        a=a-1;
        return g(x)+a;
    }
}
write (f(a));

```

7. Si descriva sinteticamente in cosa consiste, e quale problema risolve, la tecnica del *mark and sweep*.
8. Si dica cosa stampa il seguente frammento in un linguaggio con scope statico, passaggio per valore ed eccezioni:

```

int a = 9;

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

void f(int x){
    int a = 6;
    if (x==0) g();
    else try{ g(); } catch X { a--; write(a); f(0); }
}

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

```