

# Prova Scritta di Linguaggi di Programmazione I

13/04/2007

Si noti che quanto messo nei riquadri è una *bozza* fornita solo a *titolo indicativo*.  
Quindi **non** è un modello di soluzione completa che ci si aspetta ad un esame.

1. Con riferimento al seguente programma in Pascal, si rappresenti il P-code relativo alla procedura `ins`, cioè la sezione di codice all'interno del riquadro.

```
program esercizio;  
  ...  
  const N = ...;  
  var v: array [ 0 .. N ] of integer;  
  ...  
  procedure inssort( n: integer );  
  
    var k: integer;  
  
    procedure ins( x, i: integer );
```

```
begin  
  while v[i] > x do  
    begin v[i+1] := v[i]; i := i-1 end;  
  v[i+1] := x  
end;
```

```
begin  
  v[0] := -maxint;  
  for k := 2 to n do ins( v[k], k-1 )  
end;
```

```
begin ... .. end.
```

2. Sia  $C_{L_1}^{L_2 \rightarrow L_3}$  un compilatore da  $L_2$  a  $L_3$  scritto in  $L_1$  e  $I_{L_1}^{L_2}$  un interprete di  $L_2$  scritto in  $L_1$ . Sia  $\llbracket P \rrbracket$  la funzione calcolata dal programma  $P$ . Si dica cosa produce la valutazione  $\llbracket I_{L_0}^{L_1} \rrbracket (\llbracket I_{L_0}^{L_2} \rrbracket (C_{L_2}^{L_2 \rightarrow L_1}, C_{L_2}^{L_3 \rightarrow L_0}), I_{L_3}^{L_1})$ .
3. Sia  $G$  la grammatica individuata dalle produzioni

$$S ::= a A a A$$
$$A ::= a \mid b \mid a A a \mid b A b$$

pensate per generare il linguaggio  $L := \{ww \in \{a,b\}^* \mid aw = w^R a\}$ , dove  $w^R$  è la stringa  $w$  rovesciata. Si diano le stringhe di  $L$  di lunghezza  $\leq 4$  e un esempio dell'albero di parsing di una stringa generata per sbaglio ( $\in \mathcal{L}(G) \setminus L$ ). Qualora non fosse possibile dare un esempio si giustifichi il perché.

```
{w ∈ L | |w| ≤ 4} = {ε, aa, aaaa, baba}.  
abab ∈ ℒ(G) \ L.
```

4. Si mostri l'evoluzione delle variabili e l'output del seguente frammento di programma in un linguaggio C-like con assegnamento che calcola  $l$ -value prima di  $r$ -value, valutazione argomenti chiamate da sinistra a destra e indici vettori iniziati da 0:

```
char x[10] = {il_proprio_cognome};  
int i=0, k=3, y[3] = {3,2,0};
```

# Prova Scritta di Linguaggi di Programmazione I

13/04/2007

```
char mess(int i, ref char z) {
  x[++y[--i]] = x[--y[i--]] = x[y[i]++];
  write(y[0]+y[1]+y[2], z, i--k--);
  return x[y[i]];
}
write(mess(k++, x[++i]));
write(x[--i], k);
```

5. Assumendo di utilizzare nell'Esercizio 8 la tecnica di implementazione CRT con pila nascosta, si mostri schematicamente la situazione sullo stack nascosto e nel vettore centralizzato quando entra in esecuzione F per la seconda volta.

6. Rappresentando QuadTrees con il tipo di dato

```
data Eq a => QT a = C a |
  Q (QT a) (QT a) (QT a) (QT a)
```

si scriva una funzione Haskell `limitAll` che dato un colore  $c$  e una lista di QuadTrees costruisca la lista dei QuadTrees che codificano le immagini i cui pixels sono limitati al colore  $c$  (pixel originale se il colore è  $< c$ ,  $c$  altrimenti).

Si usino variabili anonime quando possibile.

**Si dia esplicitamente il tipo di tutte le funzioni.**

```
limitAll n = Prelude.map (QT.map (min n))
con QT.map soluzione dell'Esercizio apposito dell'eserciziario.
```

7. Si scriva una funzione `frame` che dato un QuadTree  $q$  restituisce **Just**  $c$  se il bordo esterno dell'immagine codificata da  $q$  è tutto composto da pixels di colore  $c$ , **Nothing** altrimenti.

Ad esempio

```
let z = C 0; d = C 2; q1 = Q d d d z
    q3 = Q d z d d
in frame $ Q q1 d q3 d
```

restituisce **Just** 2

Si usino variabili anonime quando possibile.

**Si dia esplicitamente il tipo della funzione.**

8. Si mostri l'evoluzione delle variabili e l'output del seguente frammento di programma espresso in un linguaggio C-like con scoping *dinamico*, *deep binding*, assegnamento che calcola  $l$ -value *dopo*  $r$ -value e valutazione delle espressioni da sinistra a destra:

```
{
  int x = -3, y = 1;
  int Q(ref int v, name int y) {
    int w = y;
    v -= F(w, v++ + x);
    write(--v);
    return (--x + w);
  }
  write(Q(x, x++ + F(x, y++)));
}
int x = 5, y = 7;
int F(valueresult int v, name int u) {
  v += x--; write(x,v);
  v += u; write(x,v);
}
```

# Prova Scritta di Linguaggi di Programmazione I

13/04/2007

```
    return v++;  
}  
write(x,y);
```

-3, -4, -3, -3, -3, -8, -2, -13, 10, -3, 5, 7