

Cognome e Nome: \_\_\_\_\_ Matr.: \_\_\_\_\_

**linguaggi di programmazione – A**  
**2 luglio 2019**

Esercizio 1.A – rispondere, in maniera concisa, alle seguenti domande (12 punti)

1. Quali sono i vantaggi della valutazione lazy rispetto alla valutazione eager?

---

---

---

2. Com'è strutturato un RdA (record di attivazione)?

---

---

---

3. Elencare alcune motivazioni per la programmazione concorrente.

---

---

---

4. In che modo la programmazione orientata agli oggetti permette di realizzare l'information hiding?

---

---

---

5. Quali sono le caratteristiche di un strong type system?

---

---

---

6. Dal punto di vista dei sistemi di tipi, in cosa differiscono Haskell e Scheme?

---

---

---

7. Cosa si intende per memorizzazione per righe di una matrice?

---

---

---

8. Cosa distingue l'iterazione determinata da quella indeterminata?

---

---

---

9. Nella compilazione, in cosa consiste la tecnica del bootstrapping?

---

---

---

10. Cos'è la BNF (Backus Naur Form)?

---

---

---

11. Quali sono i vantaggi dello scoping statico, rispetto a quello dinamico?

---

---

---

12. Presentare un esempio di polimorfismo ad hoc.

---

---

---

13. Quali sono le caratteristiche del passaggio dei parametri per valore?

---

---

---

14. Nei comandi in assegnazione, cosa si intende per L-value?

---

---

---

15. Come accade quando viene generata un'eccezione?

---

---

---

16. In un linguaggio di programmazione, cosa s'intende per ambiente?

---

---

---

17. Cosa sono le classi astratte? (Chiamate "interface" in Java)

---

---

---

18. Cosa sono le operazioni di test-and-set?

---

---

---

1. Cognome e Nome: \_\_\_\_\_ Matr.: \_\_\_\_\_

– 2 luglio 2019– A

Esercizio 2.A – Grammatiche (4 punti)

1. Si consideri la grammatica:

$$S \rightarrow a \mid SbS$$

- Si scrivano tutte le parole di lunghezza minore di 5 generate dalla grammatica  $S$ .
- Si stabilisca se la grammatica è ambigua, giustificando la risposta.
- È vero che “le sequenze generate dalla grammatica contengono un numero di  $a$  uguale al numero di  $b$  più 1”? Giustificare la risposta.
- Si descriva il linguaggio generato.
- Si stabilisca se è un linguaggio regolare.
- Nel caso, lo si descriva come espressione regolare.

### Esercizio 3.A – Stack di attivazione (6 punti)

1. Si mostri l'evoluzione dello stack di attivazione e dell'output dei due frammenti di programma seguenti. Si ipotizzi che il linguaggio C-like abbia scoping dinamico, deep binding, assegnamento che valuta prima r-value e poi l-value, valutazione delle espressioni da destra a sinistra, valutazione degli argomenti da destra a sinistra e indici vettori iniziati da 1:

```
int w[3]={2,4,6}, y=3, z=5;
int foo(valres int[] v, name int z){
    foreach(int x:v)
        {write(v[y--] = x + z)};
    return v[z];
}
write(foo(w, y + z--));
write(w[z]);
```

```
int x = 5;
int y = 7;
void P(ref int x, int z, int R(name int)){
    z = R(x + y);
    write(x, y, z);
}
int Q(name int w){
    return(w + x++);
}
P(y, x, Q);
write (x, y);
```

1. Cognome e Nome: \_\_\_\_\_ Matr.: \_\_\_\_\_

– 2 luglio 2019– A

Esercizio 4.A – Haskell (7 punti)

1. Si scriva una funzione Haskell che, presa in ingresso una matrice quadrata, memorizzata per righe, restituisca in uscita la sottomatrice ottenuta eliminando la prima riga e la prima colonna alla matrice originaria. Si scriva inoltre una funzione Haskell che, preso in ingresso una matrice quadrata, memorizzata per righe, restituisca la lista degli elementi contenuti nella diagonale della matrice. Si commenti il codice e si definisca il tipo di ogni funzione definita.

Esercizio 5.A – Sistema di assegnazione di tipo (4 punti)

1. Nel sistema di tipi per il linguaggio imperativo presentato nei lucidi, costruire la derivazione di tipo per il comando:

```
{ Nat x = 2; add(Nat y){x = x+y}; while (x < 20) { x = x+1; add(x) } }
```

Nota: i primi passi di derivazione, a partire dagli assiomi, se elementari o ripetizioni di derivazioni già fatte, possono essere omessi.

In alternativa si consideri il seguente comando semplificato:

```
{ Nat x = 4; Nat y = 4); while (x < 20) { y = y+1; x = x+y } }
```