

Cognome e Nome: _____ Matr.: _____

Linguaggi di programmazione – 7 settembre 2021– B

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

1. Nei comandi in assegnazione, cosa si intende per R-value?

2. Nel passaggio dei parametri per nome, cosa si intende per chiusura?

3. Cos'è una macchina virtuale?

4. Quali sono i vantaggi del polimorfismo?

5. Nella programmazione concorrente, cosa sono le porte?

6. Cosa s'intende per aliasing?

7. Nella gestione dello scoping dinamico, cosa si intende per CRT (tabella centrale dell'ambiente)?

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

9. Come si caratterizza l'ereditarietà singola?

10. Cos'è un linguaggio regolare?

11. In cosa consiste un'operazione di slicing su di un array?

12. In che parte del programma una dichiarazione è valida?

13. Cosa si intende per SOC (Service Oriented Computing)?

14. Elencare alcune differenze tra la programmazione funzionale e quella imperativa?

15. Cosa sono i moduli?

16. In quali linguaggi di programmazione le funzioni sono definite oggetti di secondo livello?

17. Quando la definizione ricorsiva di una funzione è tail recursive?

18. Cos'è un blocco protetto?

1. Cognome e Nome: _____ Matr.: _____

7 settembre 2021– B

Esercizio 2.B – Grammatiche (4 punti)

1. Date le seguenti regole di produzione:

$$\begin{aligned} P &::= \epsilon \mid aPa \mid aPb \\ S &::= aS \mid P \end{aligned}$$

siano $L(P)$ e $L(S)$ i linguaggi generati dalle grammatiche con simboli iniziali P e S

- ✓ • Si diano le stringhe di $L(S)$ e $L(P)$ di lunghezza ≤ 4 ;
- ✓ • si descrivano a parole l'insieme $L(S)$;
- ✓ • si dica se la grammatica con simbolo iniziale S è ambigua (mostrando un testimone dell'ambiguità oppure argomentando opportunamente sulla non ambiguità);
- ? • si stabilisca se il linguaggio $L(S)$ è regolare (argomentando opportunamente la risposta), e nel caso, lo si descriva in linguaggio mediante un'espressione regolare.
- • si abbozzino delle regole Lex-Alex o YACC-Happy, per la creazione di un programma che riconosca le parole di $L(S)$.

Esercizio 3.B - 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 scope dinamico e deep-binding, assegnamento che calcola prima r-value e poi l-value, valutazione delle espressioni da sinistra a destra, valutazione degli argomenti da sinistra a destra, e indici vettori iniziati da 0.

↳ dinamico link statico
↳ display

```
int v[3]={9,11,6}, x=2, y=4;  
int f(val int x, ref int y, name int z){  
    foreach(int w : v)  
        write (v[(y++)%3] = w + z);  
    return x+y;  
}  
write (f(y, x, x+y));
```

```
int i=5, j=7, a[]={i,j,--j};  
int F (int R(valres int)){  
    int i=10;  
    write(R(i), j++);  
    return(i + a[j%3]);  
}  
int G(name int i){  
    int H(valres int j){  
        write(a[i%3]);  
        return(j++);  
    }  
    return(F(H)+i);  
}  
write(G(++j));  
};
```


1. Cognome e Nome:_____ Matr.:_____

7 settembre 2021– B

Esercizio 4.B – Haskell (7 punti)

1. Scrivere le seguenti funzioni in Haskell:

- una funzione che, dati un lista e un suo possibile elemento, determina se l'elemento appartiene alla lista;
- una funzione che, dati un lista e un suo elemento, restituisce la lista originale senza l'elemento;
- usando le due funzioni precedenti, definire una funzione che date due liste determina se una è la permutazione dell'altra;
- una funzione che data una matrice, memorizzata per righe, determina se le righe della matrice sono le sono tutte permutazioni di una stessa lista.

Si definisca il tipo di ogni funzione definita. Si commenti il codice.

Esercizio 5.B – Dimensione dati (2 punti)

1. Nelle ipotesi che si usino parole di memoria di 4 byte con memorizzazione a parole allineate e che un intero occupi una parola di memoria ed un carattere un byte, determinare quanti byte di memoria occupano i due seguenti tipi di dato:

```
struct A {  
    char b;  
    int i[6];  
    char a[5];  
    int j;  
}
```

```
struct B {  
    char a;  
    char b;  
    int i;  
    char c;  
}
```

Esercizio 6.B – Sistema di assegnazione di tipo (3 punti)

1. Nel sistema di tipi per il linguaggio F1, costruire la derivazione di tipo per la seguente espressione:

$\backslash f : (\text{Nat} \rightarrow \text{Nat}) . (\backslash x : \text{Nat} . f(f(x)) + x)$

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