



Dipartimento di Elettronica e Informazione

Politecnico di Milano

Prof. Luca Mottola

20133 Milano (Italia)

Piazza Leonardo da Vinci, 32

Tel. (39) 02-2399.3400

Fax (39) 02-2399.3411

Fondamenti di Informatica (ELT)

15 Settembre 2015

Cognome

Nome

Matricola

Note

1. La mancanza dei dati di cui sopra comporta l'annullamento della prova.
2. Restituire **solo** queste pagine. Verranno ignorati fogli aggiuntivi. **Non** si può scrivere con la matita.
3. È proibito l'uso di dispositivi elettronici (computers, calcolatrici, cellulari, macchine fotografiche,...).
4. Qualunque frammento di codice richiesto deve essere scritto in **stampatello**.
5. Sebbene non obbligatorio, si consiglia di inserire commenti nel codice per facilitarne la correzione.
6. Non si può tenere una copia del testo dell'esame quando si lascia l'aula.
7. Un punteggio inferiore a 17 preclude la registrazione del voto complessivo anche se sufficiente.
8. Il tempo a disposizione è di 2 ore.

Domanda 1: Macchina di Von Neumann (7 punti)

Si descriva l'algoritmo che viene implementato dal seguente programma per la macchina di Von Neumann e l'output che produce sul nastro in uscita in funzione dell'input. Si descriva inoltre il ruolo che ciascuna delle celle di memoria utilizzate svolge nell'algoritmo. Si considerino come input esclusivamente *sequenze numeriche* terminate da uno zero.

1. LOAD= 0
2. STORE 101
3. LOAD= 110
4. STORE 102
5. READ
6. BEQ 12
7. STORE@ 102
8. LOAD 102
9. ADD= 1
10. STORE 102
11. BR 5
12. LOAD 102
13. SUB= 1
14. STORE 103
15. LOAD= 110
16. STORE 102
17. LOAD 103
18. SUB 102
19. BLE 33
20. LOAD@ 102
21. SUB@ 103
22. BNE 30
23. LOAD 102
24. ADD= 1
25. STORE 102
26. LOAD 103
27. SUB= 1
28. STORE 103
29. BR 17
30. LOAD= 1
31. WRITE
32. END
33. LOAD= 0
34. WRITE
35. END

Domanda 2: Codifica dell'informazione (3 punti)

Rappresentare in codifica binaria in complemento a 2 la seguente operazione:

- $(345)_{16} + (-23E)_{16}$

Si supponga di avere a disposizione 10 bit per rappresentare sia gli addendi che il risultato. Sono sufficienti?

Domanda 3: Linguaggio C (5 punti)

Si consideri il seguente codice in linguaggio C. Per *ciascuna funzione* e *ciascun blocco all'interno di ogni funzione*, elencare le variabili visibili da quel blocco e il loro tipo. Per identificare funzioni e blocchi, riferirsi ai commenti all'interno del codice in corrispondenza di ciascuna funzione e blocco.

```
// Ambiente globale
int foo(char c);
int a,b;

int main(int argc, const char * argv[]) {

    // Funzione main
    char a = '0';
    char b = 'a';
    int d;

    a=b;
    d=foo(a);

    while (b!='0') {
        // Blocco while
        int c;
        c++;
    }

    return 0;
}

int foo(char c){

    // Funzione foo
    float a=1.0;
    int i;

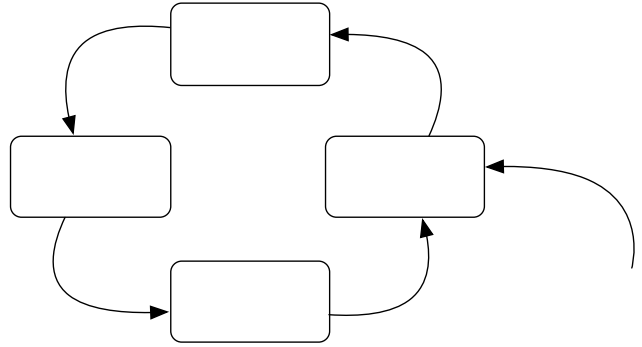
    for (i=0;i<10;i++){
        // Blocco for
        float c;
        c=c+1;
        if (a==1.0) {
            // Blocco if
            c=c+1;
        }
    }
    return c+a;
}
```


Domanda 4: Linguaggio C (6 punti)

Scrivere in linguaggio C una funzione `valori` che legge una array di interi (di dimensione a priori non nota alla funzione) e restituisce la somma dei valori più piccolo e più grande contenuti nell'array.

Domanda 4: Strutture dati dinamiche (9 punti)

Una *lista circolare* è una lista dinamica in cui l'ultimo elemento, invece che indicare un valore NULL come successivo, punta di nuovo al primo elemento. Il puntatore d'inizio lista punta ad un elemento arbitrario. Ad esempio, una lista di tal tipo con quattro elementi può essere graficamente rappresentata come in figura.



Il singolo elemento della lista è descritto dalla dichiarazione di tipo che segue. Si supponga inoltre che ciascun elemento della lista sia *unico*, cioè non esistono nella lista due elementi con lo stesso valore per il campo *data*.

```
typedef struct circularList{  
    int data;  
    struct circularList* next;  
} circularList;
```

Si implementi una funzione in linguaggio C che, data una lista circolare in input e un intero, aggiunga tale intero in una qualunque posizione della lista, controllando nel contempo che tale intero non sia già presente. In questo caso, la lista rimane inalterata.

