



## Politecnico di Milano

Quinta Facoltà di Ingegneria  
p.le Gerbetto 6, 22100 Como  
Tel.: 031-332.7332 Fax: 031-332.7321  
e-mail: giuseppe.pozzi@polimi.it  
prof. Giuseppe Pozzi - Informatica II

---

### Informatica II - Prova del 15 luglio 2002

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Matricola \_\_\_\_\_

Corso di studi in ingegneria \_\_\_\_\_

*Compilare con cura il presente foglio.* L'elaborato dovrà essere riportato per intero sul presente foglio. Non verranno considerati fogli aggiuntivi<sup>1</sup>.

---

**Norme.** La prova scritta odierna costituisce il recupero della seconda prova in itinere. Affinchè la prova sia ritenuta superata è necessario ottenere almeno 6.5 punti su un totale disponibile di 12 punti. Non si possono consultare libri, manuali, appunti: è richiesto di usare solo la carta fornita dal docente. Il candidato deve **affrontare tutti i temi** proposti in 2 ore.

---

### Temi

(1) Si descriva *sinteticamente* il funzionamento di un microprocessore di tipo pipeline a 4 stadi, riportando le attività svolte da ogni stadio su un'osservazione dalla durata di 8 cicli di clock.

spazio per la risposta

---

<sup>1</sup>**Osservazione.** Completare le specifiche ove necessario. La chiarezza e l'ordine dello svolgimento partecipano a stabilire l'entità del voto.

(2) Si costruisca un banco di memoria, dotato di *tutti* i collegamenti necessari, da 4 megaparole con lunghezza 16 bit, avendo a disposizione solo componenti da 1 megaparole con lunghezza 4 bit.

spazio per la risposta

(3) Si costruisca un registro a 4 bit, con caricamento di ingresso seriale e parallelo e con uscita parallela con scorrimento verso destra.

spazio per la risposta

(4) Si effettui la traduzione in linguaggio JVM e la relativa codifica binaria, utilizzando i dati di tabella 1, della funzione C `max` che restituisce il massimo di una sequenza di numeri positivi acquisiti da tastiera. Si supponga che esista la funzione `leggi` che acquisisce da tastiera un numero intero.

Opcode	Mnemonic
<code>0x15</code>	<code>ILOAD <i>varnum</i></code>
<code>0x13</code>	<code>LDCW <i>index</i></code>
<code>0x36</code>	<code>ISTORE <i>varnum</i></code>
<code>0x59</code>	<code>DUP</code>
<code>0x5F</code>	<code>SWAP</code>
<code>0xA7</code>	<code>GOTO <i>offset</i></code>
<code>0x99</code>	<code>IFEQ <i>offset</i></code>
<code>0x9B</code>	<code>IFLT <i>offset</i></code>
<code>0x9F</code>	<code>IF_ICMPEQ <i>offset</i></code>
<code>0xB6</code>	<code>INVOKEVIRTUAL <i>offset</i></code>
<code>0xAC</code>	<code>IRETURN</code>
<code>0x57</code>	<code>POP</code>
<code>0x60</code>	<code>IADD</code>
<code>0x64</code>	<code>ISUB</code>
<code>0x10</code>	<code>BIPUSH <i>byte</i></code>
<code>0x84</code>	<code>IINC <i>varnum const</i></code>
<code>0xC4</code>	<code>WIDE</code>

Tabella 1: Opcode delle istruzioni

```
int max()
{ int m, curr;
  m=-1;
  scanf("%d", &curr);
  while(curr>-1){
    if (curr>m) m=curr;
    scanf("%d", &curr);
  }
  return(m);
}
```

spazio per la risposta

spazio per la risposta

---

**Parte riservata al docente**

Es. 1	Es. 2	Es. 3	Es. 4	<b>Totale</b>

---