



Politecnico di Milano

Facoltà di Ingegneria Civile, Ambientale e Territoriale

Informatica (ICA-LC) [091456] – Informatica A (ICA-LC) [091461]

Informatica [097256]

Prof. G. Boracchi

Allievi Ingegneria Civile e Ambientale

I prova in itinere (26 Novembre 2015)

<i>Cognome e nome</i>	
<i>Matricola</i>	
<i>Firma</i>	

Domanda	1	2	3	4	5	TOT
Punteggio max	4	4	4	8	12	32
Punteggio						

La **durata** della **prova** è di **2 ore**. Non è consentito consultare libri o appunti, non è consentito l'uso di calcolatrici.

Scrivere solo sui fogli distribuiti utilizzando il retro delle pagine in caso di necessità e cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.

Per tutti gli esercizi non è sufficiente fornire il risultato, ma è **necessario mostrare il procedimento seguito**.

Gli allievi sono invitati a privilegiare **chiarezza, proprietà di linguaggio e sinteticità** nelle risposte agli esercizi, con l'obiettivo di **dimostrare la loro conoscenza degli argomenti**.

Domanda 1. Rappresentazione binaria di valori numerici (4 punti).

Dati i tre numeri :

- N1 = 32 in base 8
- N2 = 56 in base 10
- N3 = 101010 in base 2 (senza segno)

indicare il numero minimo di bit che occorrono per rappresentare ciascuno di questi numeri in complemento a 2 e effettuare la conversione. Si indichi quindi il numero di bit necessario per rappresentare tutti questi numeri in complemento a 2 e nel seguito si utilizzino codifiche con questo numero di bit.

SOLUZIONE: con n bit codifico in CP2 $[-2^{n-1}, +2^{n-1})$

$$N1 = 32_8 \Rightarrow 26_{10} \Rightarrow \mathbf{6bit} \Rightarrow \mathbf{011010}$$

$$N2 = 56_{10} \Rightarrow \mathbf{7bit} \Rightarrow \mathbf{0111000}$$

$$N3 = 2+8+32 \Rightarrow 42_{10} \Rightarrow \mathbf{7bit} \Rightarrow \mathbf{0101010}$$

Per rappresentare tutti i numeri occorrono 7 bit

Eeguire in complemento a 2 (mostrando i passaggi, indicando esplicitamente se si verifica overflow e motivando la risposta) le operazioni:

- $N1 + N2$
- $N1 - N3$.

SOLUZIONE:

$$\begin{array}{r} 0011010 \text{ (N1)} + \\ 0111000 \text{ (N2)} = \\ \hline 0110010 \Rightarrow +82_{10} \end{array}$$

Si Overflow (operandi concordi ma risultato sullo stesso numero di bit discorde)

$$\begin{array}{r} 0011010 \text{ (N1)} + \\ 1010110 \text{ (-N3)} = \\ \hline 1110000 \Rightarrow -16_{10} \end{array}$$

No Overflow

Domanda 2. Logica booleana (4 punti).

Si costruisca la tabella di verità della seguente espressione booleana.

(not A) or (B and C) or A and (C or B)

ABC
0 0 0 → 1
0 0 1 → 1
0 1 0 → 1
0 1 1 → 1
1 0 0 → 0
1 0 1 → 1
1 1 0 → 1
1 1 1 → 1

Domanda 3. Architettura di un calcolatore (4 punti).

Si discuta il ruolo del bus di sistema, specificando le varie linee che lo compongono. Evitare di fornire una spiegazione prolissa.

Domanda 4. Tracing di programmi (8 punti).

Si consideri la seguente funzione `funz` (`funz.m`) ed i seguenti script (`script1.m` e `script2.m`) che sono salvati in files distinti.

- A. Si dica cosa restituisce `funz('A')`, `funz('a')`, `funz(1)` specificando anche il tipo di eventuali variabili restituite. Si descriva (due righe massimo) cosa fa la funzione `funz`
- B. Si dica cosa restituisce `script1` se l'utente inserisce `'Ciao Mamma'`. Se ne descriva quindi il funzionamento (5 righe massimo)
- C. Si dica cosa restituisce `funz('avs')` e si fornisca una giustificazione (2 righe massimo)
- D. Si descriva il funzionamento di `script2` (due righe massimo) indicando il ruolo della variabile `pippo`. E' possibile dire quanto vale `pippo` alla fine dello script?

```
% funz.m
function cc = funz(tt)
    cc = tt(1);
    if cc >= 'a' && cc <= 'z'
        cc = cc - ('a' - 'A');
        cc = char(cc);
    end
end

% script1.m
txt = input('inserire: ', 's');
mm = [];
for tt = txt
    mm = [funz(tt), mm];
end
disp(mm);

% script2.m
txt = input('inserire: ', 's');
pippo = 0;
for tt = txt
    if funz(tt) < 'A'
        pippo = 1;
    end
end
disp(pippo);
```

N.B. La funzione `char(d)` trasforma un intero `d` nel corrispettivo carattere della tabella ASCII. Si ricordi che `char(65)` restituisce `'A'`, `char(66)` restituisce `'B'` e che `char(97)` restituisce `'a'`.

--

SOLUZIONE

A)

```
funz('A') = 'A'  
funz('a') = 'A'  
funz(1) = 1  % è un double
```

`funz` trasforma i caratteri minuscoli in maiuscoli. Numeri, caratteri maiuscoli ed altri caratteri della tabella ASCII, rimangono invariati.

B)

Lo script richiede all'utente una stringa, la trasforma in lettere maiuscole mediante la `funz` ogni e la scrive in mm dall'ultimo carattere al primo. Quindi viene visualizzata la stringa al contrario. Lo script visualizzerà quindi

```
'AMMAM OAIC'
```

C) la funzione restituisce 'A' perché nella prima riga di `funz` viene preso solo il primo carattere dell'input

D) `pippo` opera come una variabile di flag che ci dice se il testo inserito contiene un numero o un carattere ascii minore di 'A'.

Domanda 5. Linguaggio Matlab (12 punti).

Scrivere una funzione `notazioneEsponenziale.m` che riceve un numero e restituisce una stringa di testo contenente la rappresentazione in notazione esponenziale di un numero.

Ad esempio la funzione `notazioneEsponenziale` invocata su 1245 restituisce la stringa `'1245 = 1 x 1000 + 2 x 100 + 4 + 10 + 5 x 1'`

Si proceda come segue:

- A) Scrivere l'header della funzione `notazioneEsponenziale`
- B) Scrivere uno script che richiede all'utente un numero n , controlla che n sia un intero positivo (massimo 10 cifre), e che ripete la richiesta fino a quando il numero inserito non soddisfa i requisiti. Si invochi quindi la funzione `notazioneEsponenziale` e si stampi a schermo la stringa restituita.
- C) Si scriva il corpo della funzione `notazioneEsponenziale` utilizzando lo pseudocodice (si utilizzi SIMITA o diagrammi di flusso)
- D) si scriva la funzione `notazioneEsponenziale` in linguaggio Matlab

N.B. E' possibile estrarre le cifre di un intero utilizzando l'algoritmo delle divisioni successive rispetto alla base 10 (lo stesso algoritmo utilizzato per la conversione da decimale a binario). Esistono però metodi più rapidi per ottenere questo risultato.

SOLUZIONE

```
function out = notazioneEsponenziale(intero)
str = num2str(intero);

out = [str, ' = '];
exp = length(str) - 1;

for ss = str
    out = [out, ' ', ss, ' x ', num2str(10^exp)];
    if exp > 0
        out = [out, ' +'];
    end
    exp = exp - 1;
end
```

%% oppure, soluzione utilizzando l'algoritmo delle divisioni successive

```
function out = notazioneEsponenziale(intero)
str = num2str(intero);

out = [num2str(intero), ' = '];
%
ii = 1;
while(intero > 0)
    cifra(ii) = mod(intero, 10);
    intero = floor(intero / 10);
    ii = ii + 1;
end
```



```
exp = length(cifra) - 1;
for ii = length(cifra) : -1 : 1
    out = [out, ' ', num2str(cifra(ii)), ' x ',
num2str(10^exp)];
    if exp > 0
        out = [out , ' +'];
    end
    exp = exp - 1;
end
```

