Fondamenti di Informatica, A.A. 2011-2012

26/02/2013

Esercizio 1

È dato il frammento di codice Matlab

```
m = 5;

re= 3;

n = m-re;

u = ones(n,m);

a = [1  2  3  5  7  11  13  17];

d = diag(a);

nd = d(1:end-re,1:end-re);

z = u*nd;

disp(z);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

L'interprete Matlab stampa

1 2 3 5 7
1 2 3 5 7

Esercizio 2

Definire il concetto di ricorsione e dare almeno un esempio di funzione ricorsiva implementata in Matlab.

Soluzione

Per ricorsione si intende la tecnica di programmazione per cui una funzione invoca un'istanza di se stessa a cui vengono passati opportuni parametri al fine di risolvere un dato problema; la catena di chiamate ricorsive deve comunque essere terminata da una (o più) istanze in cui il valore venga calcolato direttamente. Un esempio di funzione che fa uso della ricorsione è quella che consente di calcolare il fattoriale:

```
function ret = real_factorial(num)
```

```
    if num < 0 || (num-floor(num)) > 0
        error('Bad parameter');
    else if num == 0
        ret = 1;
    else
        ret = num * real_factorial(num-1);
    end

end
```

Esercizio 3

In una prova di esame di informatica viene richiesto di calcolare l'equivalente decimale di un numero binario rappresentato nel vettore `a` in stile BigEndian (vale a dire che la cifra più significativa del numero binario è quella che si trova in posizione a(1)). Uno studente propone la seguente soluzione:

```
n = 0;

l = length(a);

for i=1:l
    idx = l-i;
    n = n + (2*a(i)).^idx;
end
```

Si chiede di trovare l'errore e di proporre una correzione adeguata.

Soluzione

Il problema risiede nel fatto che la posizione meno significativa del numero binario viene sempre elevata alla potenza 0; nel caso in cui il valore del bit meno significativo sia 0, il codice proposto calcola 0^0 e Matlab restituisce 1 come risultato. Una possibile correzione è la seguente:

```
n = 0;

l = length(a);

for i=1:l
    idx = l-i;
    n = n + (2.^idx).*a(i);
end
```

volendo adoperare le operazioni su array si ha la seguente:

```
a = logical(a); %Necessaria per trasformare interi in logici
b = length(a)-1:-1:0;
n = sum(2.^b(a));
```

Esercizio 4

È data la seguente funzione Matlab

```
function [x]=mistery(d,x,y,k,g)
  for i=1:g(k)
      y = y + d.\x;
  end
end
```

Si stimi il numero di operazioni aritmetiche eseguito dal codice, considerando le seguenti possibili scelte di $g(k)$

1. $g(k) = 3$;
2. $g(k) = k$;
3. $g(k) = 3 * k$.

Si assuma che d,x e y siano vettori della stessa lunghezza.

Soluzione

Il comando `d.\x` effettua la divisione elemento per elemento fra i vettori, questo significa n operazioni aritmetiche; tale risultato viene sommato alla variabile y causando quindi n somme, per un totale di 2n operazioni aritmetiche per iterazione. Nel caso in cui $g(k) = 3$ otteniamo quindi $6n$ operazioni aritmetiche. Nel caso in cui $g(k) = k$ otteniamo $2kn$ operazioni aritmetiche. Nel caso in cui $g(k) = 3 * k$ otteniamo $6kn$ operazioni aritmetiche. Se $k \approx n$ allora gli ultimi due casi daranno luogo ad un numero di operazioni $O(n^2)$.

Esercizio 5

È dato il frammento di codice Matlab

```
ii = 1; jj = 0; kk = 0; hh = 0;

while(ii<=length(a))
    if mod(a(ii),2)==0
        jj = 12+ii;
    else
        kk = 11+ii;
        if mod(jj,12)==0
            hh = hh+1;
        end
    end
    ii = ii+1;
end
```

Riscrivere il codice facendo uso di uno o più cicli `for` ed utilizzando obbligatoriamente il costrutto `switch` al posto degli `if`.

Soluzione

Una possibile soluzione è la seguente:

```
ii = 1; jj = 0; kk = 0; hh = 0;

for ii=1:length(a)
    switch mod(a(ii),2)
        case 0
            jj = 12+ii;
        otherwise
            kk = 11+ii;
        switch mod(jj,12)
            case 0
                hh = hh+1;
        end
    end
end
```