

Fondamenti di Informatica, A.A. 2012-2013

10/07/2013 – fila A

Esercizio 1

È dato il frammento di codice Matlab

```
n = 6;  
v = [1 3 2 4 5 2];  
m = 1;  
for i=1:n  
    v(i) = v(i)^2;  
    [m j] = max(v);  
    v(i) = m+v(i);  
end  
disp(v);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

Per la corretta soluzione del problema occorre notare che il massimo valore nel vettore cambia ad ogni iterazione. L'interprete Matlab stampa

```
octave:5> disp(v);  
    6    18    22    38    63    67
```

Esercizio 2

Discutere le funzionalità legate all'I/O in Octave/Matlab con particolare riguardo ai file di testo.

Soluzione

Octave/Matlab consente di gestire agevolmente molteplici formati di file (testuali, audio, video, ...) grazie a delle funzioni di alto livello. E' inoltre possibile accedere a file testuali attraverso funzioni di basso livello che permettono una gestione più generale utile nel caso in cui si debbano maneggiare formati particolari non leggibili dalle funzioni di alto livello.

Per poter accedere ad un file di testo con tali funzioni è richiesta, come prima operazione, la sua apertura attraverso la funzione `fopen('path', 'mode')` la quale prende come parametri il percorso del file e la modalità di apertura; quest'ultima può essere di sola lettura, di sola scrittura, di scrittura alla fine del file, ecc. La funzione `fopen()` ritorna un identificativo univoco; tale numero verrà utilizzato da tutte le funzioni di basso livello per poter operare correttamente sul file in questione, attraverso ad esempio le funzioni `fgets`, `fscanf`, `fprintf`.

Una particolarità di Octave/Matlab è quella di poter salvare/caricare il valore delle variabili del workspace con i semplici comandi `save` e `load`.

Esercizio 3

In una prova di esame di informatica viene richiesto di trovare i numeri primi compresi fra 1 e 100. Uno studente propone il seguente codice

```
n = 1:100;
k=1;
p = [];
p(k)=2;
i=1;
j=1;

while(i<=length(n))
    primo=true;
    for j=1:k
        if mod(n(i),p(j)) == 0
            primo = primo & false;
        end
    end
    if primo
        k = k+1;
        p(k)=n(i);
    end
    i = i+1;
end
```

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

Soluzione

Il problema riguarda la presenza del numero 1 fra quelli candidati ad essere numeri primi; il codice proposto trova che 1 soddisfa le condizioni richieste (dato che $\text{mod}(1,2) == 1$), lo include nell'elenco dei numeri primi, e da quel momento in poi qualunque numero risulta non primo, in quanto il resto della divisione per 1 è sempre 0.

Le possibili soluzioni riguardano quindi l'esclusione di tale numero dai candidati. Una possibile soluzione è quindi la seguente:

```
n = 3:100;
k=1;
p = [];
p(k)=2;
i=1;
while((i<=length(n)))
    primo=true;
    for j=1:k
        if mod(n(i),p(j)) == 0
            primo = primo & false;
        end
    end
    end
```

```

    if primo
        k = k+1;
        p(k)=n(i);
    end
    i = i+1;
end

```

in cui si escludono i numeri 1 e 2 dai candidati.

Esercizio 4

È data la seguente funzione Matlab

```

function [x n]=simple(a,x,b)
n=length(x);
for i=1:n
    x = a\b;
    n = length(x)-1;
end
end

```

Si stimi il numero di operazioni aritmetiche eseguite assumendo che **a** sia una matrice triangolare inferiore e che **x** e **b** siano arrays.

Soluzione

Il comando **a\b** risolve un sistema lineare; essendo **a** triangolare inferiore, questo richiede n^2 operazioni aritmetiche. Per ogni iterazione del ciclo viene inoltre eseguita una singola sottrazione; dato che il ciclo viene eseguito n volte si avranno in totale $n(n^2 + 1) = n^3 + n$ operazioni aritmetiche. Da notare che la definizione della variabile n all'interno del ciclo non influenza il numero di iterazioni eseguite.

Esercizio 5

È dato il frammento di codice Matlab

```

n=100;
i = 1; j = 1;
maxt = 30000;
v = [];
x = 1:10;
t = 0;

while i<=n && t<=maxt
    while j<i
        v(i) = j+i;
        j = j+2;
        t = max(v);
    end

```

```
    j = 1;  
    i = i+1;  
end
```

Riscrivere il codice facendo uso di uno o piu' cicli `for`.

Soluzione

Una possibile soluzione è la seguente:

```
n=100;  
i = 1; j = 1;  
maxt = 30000;  
v = [];  
x = 1:10;  
t=0;  
  
for i=1:n  
    for j = 1:2:i-1  
        v(i) = j+i  
        t = max(v);  
    end  
    if(t > maxt)  
        break;  
    end  
end  
end
```

10/07/2013 – fila B

Esercizio 1

È dato il frammento di codice Matlab

```
n = 5;
v = [1 3 4 3 2];
m = 2;
for i=1:n
    v(i) = v(i)*2;
    s = sum(v);
    m = min(min(v), s+m);
    v(i) = m+v(i);
end
disp(v);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

L'interprete Matlab stampa

```
octave:10> disp(v);
    4    8   10    8    8
```

Esercizio 2

Discutere l'utilizzo dell'operatore `:`. Riportare un esempio tipico di indicizzazione di array attraverso il suo utilizzo. Supponendo che **a** sia una matrice $n \times m$, dire cosa accade utilizzando il seguente comando `a(:,n:-1:1)`.

Soluzione

L'operatore `:` viene utilizzato in Octave/Matlab per definire un range di valori. Un tipico esempio del suo utilizzo per indicizzare un array **b** è il seguente `b(1:n)` dove **b** è un array di almeno n elementi. Nel caso in cui **a** sia una matrice $n \times m$ il comando `a(:,n:-1:1)` ritorna una matrice quadrata $n \times n$ in cui le colonne sono le prime n colonne di **a** ma riportate in senso inverso (assumendo che $n \leq m$).

Esercizio 3

In una prova di esami di informatica viene richiesto di implementare un programma che simuli il contachilometri di un veicolo. Uno studente propone il seguente codice

```
b=10; i=0; j=0; k=0;
v = zeros(1,3);
while(k<b)
    while(j < b)
```

```

        while(i<b)
            v(3) = i;
            disp(v);
            i=i+1;
        end
        j = j+1;
        v(2) = j; v(3) = 0;
    end
    k=k+1;
    v(1) = k;
    v(2) = 0; v(3) = 0;
end

```

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

Soluzione

Il problema è nella mancata reinizializzazione delle variabili i e j alla fine dei due cicli while piu' esterni, per cui dopo la configurazione di cifre 009 il ciclo piu' interno non viene piu' eseguito e non vengono stampate ulteriori sequenze di cifre. Una possibile soluzione è la seguente:

```

b=10;
i=0;
j=0;
k=0;
v = zeros(1,3);
while(k<b)
    while(j < b)
        while(i<b)
            v(3) = i;
            disp(v);
            i=i+1;
        end
        j = j+1;
        v(2) = j;
        v(3) = 0;
        i=0;
    end
    k=k+1;
    v(1) = k;
    v(2) = 0;
    v(3) = 0;
    j = 0;
end

```

Esercizio 4

È data la seguente funzione Matlab

```

function [x n]=simple(a,x,b)
    n=length(x);
    for i=1:n
        for j=1:2:n
            x(i) = a(i)/b(i);
        end
        n = length(x) - 1;
    end
end

```

Si stimi il numero di operazioni aritmetiche eseguite assumendo che a,b ed x siano arrays della stessa lunghezza multipla di 2.

Soluzione

Nel ciclo `for` interno viene eseguita una operazione aritmetica per ogni iterazione; il ciclo viene eseguito $n/2$ volte. Nel ciclo esterno viene eseguito il ciclo interno e una operazione aritmetica aggiuntiva, verranno quindi eseguite in totale $n^2/2 + n$ operazioni aritmetiche. Da notare che il calcolo della variabile n all'interno del ciclo non influenza il numero di iterazioni eseguite, in quanto n viene diminuito di 1, passando da pari a dispari, ed il ciclo interno viene appunto eseguito sui dispari, quindi rimane invariato. La modifica di n non altera (secondo la semantica del ciclo `for`) il ciclo esterno.

Esercizio 5

È dato il frammento di codice Matlab

```

n = 100;
j = 0;
k = 1;
for i = 1:4:n
    for j=1:n
        if (k>1000 || j==n)
            k = k-2;
            break;
        else
            k = k+j .* 2;
        end
    end
end

```

Riscrivere il codice facendo uso di uno o piu' cicli `while` e NON usando blocchi condizionali `if` o `switch`.

Soluzione

Una possibile soluzione è la seguente:

```

n = 100;

```

```
j = 0;
k = 1;
i = 1;

while (i<=n)
    j = 1;
    while (j<=n && k<=1000)
        k = k+j.*2;
        j=j+1;
    end
    k = k-2;
    i = i+4;
end
```