

Esame di Programmazione Informatica

31 gennaio 2025

Esercizio 1 (16/30)

Si consideri la somma s dei quadrati di due interi successivi: $s = n^2 + (n + 1)^2$.

Scrivere una funzione MATLAB che prenda n come argomento in ingresso, e restituisca come argomento in uscita un booleano (`logical` in MATLAB) che sarà `true` se s è un quadrato perfetto, ossia se s è il quadrato di un altro numero naturale, `false` altrimenti.

Ad esempio, per $n = 3$ la funzione dovrà restituire `true` poichè $3^2 + 4^2 = 25 = 5^2$. Per $n = 2$ la funzione dovrà restituire `false` poichè $2^2 + 3^2 = 13$ che non è un quadrato perfetto.

Mostrare poi come utilizzare la precedente funzione per determinare, con un ciclo `for`, per quali interi n compresi tra 1 e 1000 la corrispondente somma s è un quadrato perfetto.

Soluzione: calcoliamo direttamente la radice quadrata di s , e verifichiamo quindi che tale radice sia intera mediante sottrazione della sua parte intera:

```
                                sommaQuadrati.m
function quadratoPerfetto = sommaQuadrati(n)
    s = n^2 + (n+1)^2 ;
    r = sqrt(s) ;
    parteNonIntera = r - floor(r) ;
    quadratoPerfetto = parteNonIntera == 0 ;
end
```

Utilizzo per determinare i valori di n tra 1 e 1000:

```
for n = 1 : 1000
    if sommaQuadrati(n)
        disp(n) ;
    end
end
```

ottenendo i seguenti valori: $n = 3, 20, 119, 696$.

Esercizio 2 (17/30)

Si consideri un poligono definito da n vertici consecutivi $\mathbf{x}_i$, ognuno dei quali è identificato dalle sue coordinate cartesiane $\mathbf{x}_i = (x_i, y_i)$. Scrivere una funzione MATLAB che prenda come argomenti in ingresso i due vettori $\mathbf{x} = [x_1, \dots, x_n]$ e $\mathbf{y} = [y_1, \dots, y_n]$ rispettivamente delle coordinate x e y degli n vertici, e restituisca come argomento in uscita il perimetro del poligono. (Suggerimento: si scorrono tutti i vertici del poligono sommando le lunghezze dei lati.)

Mostrare poi come utilizzare la precedente funzione per calcolare il perimetro del seguente poligono che approssima un'ellisse con semiassi lunghi 2 e 1:

```
t = linspace(0, 2*pi, 100) ;  
x = 2*cos(t) ;  
y = sin(t) ;
```

Soluzione: utilizziamo un ciclo **for** per scorrere tutti gli n vertici e calcolare le lunghezze dei lati. E' opportuno aggiungere in coda ai vettori $\mathbf{x}$ e $\mathbf{y}$ le coordinate del primo vertice di modo da semplificare il calcolo della lunghezza dell'ultimo lato:

perimetro.m

```
function L = perimetro(x, y)  
    n = length(x) ;  
    x(end+1) = x(1) ;  
    y(end+1) = y(1) ;  
    L = 0 ;  
    for i = 1 : n  
        dx = x(i+1) - x(i) ;  
        dy = y(i+1) - y(i) ;  
        dL = sqrt(dx^2 + dy^2) ;  
        L = L + dL ;  
    end  
end
```