

Dec 05, 13 19:00	08-04-fibonacci.m	Page 1/2
<pre> %% % Scrivere un programma che calcoli la sequenza di Fibonacci di lunghezza 100, % poi richieda di inserire un numero e valuti se il numero è di Fibonacci. % Altrimenti restituisce il numero di Fibonacci più vicino. % % % La successione di Fibonacci è definita così: % % <math>F(n) = F(n-1) + F(n-2)</math> , <math>n &gt; 1</math> % <math>F(0) = 0</math> % <math>F(1) = 1</math>  % fare un programma che salva in un vettore i primi % 100 numeri di fibonacci  N = 30;  F = [0 , 1];  % diverse soluzioni sono proposte  % ii = 3; % while(ii &lt;= N) %     F(ii) = F(ii - 1) + F(ii - 2); %     ii = ii + 1; % end  for ii = [3 : 1: 100]     F(ii) = F(ii - 1) + F(ii - 2); end  % while(length(F) &lt; N) %     F = [F , F(end) + F(end - 1)]; % end  % figure(8) , plot(F)  % richiede un input all'utente da command line. Qualsiasi input "alla matlab" ve rrà associato a p p = input(['inserire un numero ']);  % guardo se un elemento in F coincide con p. Faccio la somma del vettore logico p er vedere se c'è almeno un valore uguale % a 1 if (sum(F == p) == 1)     disp([num2str(p) ' è di fibonacci']); else     % prendo la sequenza dei numeri di Fibonacci maggiori di p     mag_b = F &gt;= p;     mag = F(mag_b);      % il più piccolo maggiorante ( -&gt; candidato ad essere il più vicino a p), è     % il primo elemento dei aggiornati     M = mag(1);      % prendo la sequenza dei numeri di Fibonacci minori di p     minori_b = F &lt; p;     minori = F(minori_b);      % il più grande minorante( -&gt; candidato ad essere il più vicino a p), è il     % l'ultmo elemento dei minoranti     m = minori(end); </pre>		

Dec 05, 13 19:00	08-04-fibonacci.m	Page 2/2
<pre> % ho implicitamente assunto che le due sequenze sono ordinate in maniera % crescente  % trovo tra M e m il più vicino, ovviamente posso assumere <math>m &lt; p &lt; M</math> if(M - p &lt; p - m)     P = M; else     P = m; end  disp(['il numero fb+vc a ' , num2str(p) , ' è ' , num2str(P)]) end </pre>		