

Informatica B

Esercitazione 9 (Soluzioni)

24 novembre 2021

Scrivere **script MATLAB** che risolvano i seguenti problemi:

Accesso Logico

9.1 Data una matrice di numeri (assumere almeno due colonne), stampare le righe per cui l'elemento al secondo indice è maggiore di 2.

```
1 % Es. 9.1
2 % Righe
3
4 close all
5 clear
6 clc
7
8 x = input('Inserire una matrice\n');
9
10 % trovare le righe per cui il secondo elemento è maggiore
    di 2
11 seconda_colonna = x(:, 2);
12 righe_valide = seconda_colonna > 2;
13
14 %prendere solo le righe valide
15 y = x(righe_valide, :);
16
17 disp(y);
```

9.2 Data una matrice quadrata in input, estrarre solo gli elementi delle due diagonali.

Aiuto: la funzione `eye(n)` crea una matrice quadrata identità di dimensioni $n \times n$.

```
1 % Es. 9.2
```

```

2 % Diagonali
3
4 close all
5 clear
6 clc
7 M = input("Inserire una matrice quadrata");
8 % prendere la dimensione della matrice
9 a = length(M);
10 % creare una matrice di identita' della stessa dimensione
11 % quindi con valori 1 solo nella diagonale principale
12 d1 = logical(eye(a));
13 % invertire le collone (o le righe) matrice per ricavare
    gli indici per la
14 % seconda diagonale
15 d2 = d1(:,end:-1:1);
16 diagonali=M(d1 | d2);
17
18 disp("I valori nelle diagonali sono:")
19 disp(diagonali)

```

Strutture di Controllo

9.3 Scrivere un programma matlab che legge numeri dall'utente fino a che l'utente inserisce un numero negativo o 0. Dopo la lettura, il programma stampa una matrice con tutti 0 apparte la diagonale dove si trovano i numeri inseriti.

```

1 % Es. 9.3
2 % Matrice Diagonale
3
4 close all
5 clear
6 clc
7
8 matrice = [];
9 ii = 0;
10 flag = true;
11 while flag
12     numero = input("Inserire un numero");
13     flag = numero > 0;
14     if flag
15         ii = ii + 1;
16         % inserire il numero nella diagonale
17         % la matrice si intende in modo automatico
18         matrice(ii , ii) = numero;
19     end
20 end

```

```
21
22 disp(matrice)
```

9.4 Dato un intero $n > 0$, salvare in un array e stampare la successione a_i così definita:

- $a_1 = x$;
- Se $a_i = 1$, la successione termina;
- Se a_i è pari, $a_{i+1} = a_i/2$;
- Se a_i è dispari, $a_{i+1} = 3a_i + 1$.

Attenzione: il programma potrebbe non terminare!

https://it.wikipedia.org/wiki/Congettura_di_Collatz

```
1 % Es. 9.4
2 % Successione di Collatz
3
4 close all
5 clear
6 clc
7
8
9 n = 0;
10 ii = 1;
11 %La successione potrebbe non terminare:
12 % fissiamo una lunghezza massima
13 max_len = 1000;
14
15 %Input
16 while(n <= 0 || floor(n) ~= n)
17     n = input('Inserire un intero positivo\n');
18 end
19
20 %Successione
21 succ = n;
22
23 while(n ~= 1 && ii < max_len)
24     if(mod(n, 2) == 0)
25         n = n / 2;
26     else
27         n = 3 * n + 1;
28     end
29     ii = ii + 1;
30     succ(ii) = n;
```

```

31 end
32
33 %Questa successione termina con 1 (quando termina)
34 % se non e' cosi', qualcosa e' andato storto:
35 % la successione non e' terminata entro max_len passi
36 % oppure c'e' stato un overflow
37 if (succ(end)==1)
38     disp('succ ');
39 else
40     fprintf('Errore!\n');
41 end

```

9.5 Dato un array di numeri, dire se rispetta la seguente condizione: "La media dei primi n numeri cresce al crescere di n ".

E' sufficiente controllare che ogni elemento (a partire dal secondo) sia maggiore della media dei precedenti:

```

1 % Es. 9.5
2 % Medie crescenti
3
4 close all
5 clear
6 clc
7
8 v = input('Inserire array\n');
9 ii = 1;
10 somma = v(1);
11 media = 0;
12 flag = true;
13
14 for x = v(2 : end)
15     media = somma / ii;
16
17     if(x <= media)
18         flag = false;
19         break;
20     end
21
22     somma = somma + x;
23     ii = ii + 1;
24 end
25
26 if(flag == true)
27     fprintf('SI\n');
28 else
29     fprintf('NO\n');

```

30 `end`

9.6 Scrivere un programma che legge una matrice quadrata e aggiunge uno a tutti gli elementi della sottodiagonale superiore, e rimuove uno agli elementi della sottodiagonale inferiore. Per sottodiagonale superiore si intendono gli elementi una riga sopra quelli della diagonale principale, e viceversa per la sottodiagonale inferiore.

```
1 % Es. 9.6
2 % Sottodiagonali
3
4 close all
5 clear
6 clc
7
8 mat = input('Inserire una matrice\n');
9
10 disp("Matrice originale:")
11 disp(mat)
12 sgiad_superiore = zeros(size(mat));
13 sgiad_inferiore = zeros(size(mat));
14 for ii = 2:size(mat)
15     mat(ii-1,ii) = mat(ii-1, ii) + 1;
16     mat(ii, ii-1) = mat(ii, ii-1) - 1;
17 end
18
19 disp("Matrice Modificata:")
20 disp(mat)
```

9.7 (Bonus) Data una matrice, dire se rispetta la seguente condizione: "Ogni riga può essere ottenuta moltiplicando la prima riga per uno scalare (possibilmente diverso)".

Esempio: $[1 \ 2 \ 3; \ 2 \ 4 \ 6; \ 1.5 \ 3 \ 4.5]$ soddisfa la condizione.

Suggerimenti:

- Con `for v = A` posso iterare le colonne di una matrice. E' possibile iterare le righe facendo opportune trasposizioni;
- `if(any(v))` è verificato se **almeno un** elemento di `v` è vero;