

Informatica B

Esercitazione 10 (Soluzioni)

29 novembre 2021

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

Funzioni MATLAB

10.1 La *radice numerica* di un numero intero è ottenuta sommando le sue cifre, poi sommando le cifre del risultato, eccetera, fino a quando non si ottiene una singola cifra.

Esempio: $65536 \rightarrow 6 + 5 + 5 + 3 + 6 = 25 \rightarrow 2 + 5 = 7$.

Scrivere una **funzione** che calcoli la radice numerica di un numero intero. Quindi, scrivere uno **script** che, dato un array di interi, calcola la radice numerica di ogni suo elemento.

```
1 function s = somma_cifre(n)
2     %Somma delle cifre di un intero positivo
3     s = 0;
4
5     while(n > 0)
6         s = s + mod(n, 10);
7         n = floor(n/10);
8     end
9 end

1 function c = radnum(n)
2 %Radice numerica
3
4 %Elimina segno e parte decimale
5 c = abs(floor(n));
6
7 while(c > 9)
8     c = somma_cifre(c);
9 end
```

```

1 %Es. 10.1
2 %Radice numerica
3
4 close all
5 clear
6 clc
7
8 vett = input('Inserire array di interi\n');
9
10 for n = vett
11     disp(radnum(n));
12 end

```

10.2 Scrivere una **funzione** che, dati *due* numeri, restituisca il minimo, il massimo e la media. Quindi, scrivere una funzione che faccia lo stesso, ma per un numero qualsiasi di numeri.

```

1 function [mi, ma, me] = stats2(a, b)
2 %Minimo, massimo, media
3     if a > b
4         ma = a;
5         mi = b;
6     else
7         ma = b;
8         mi = a;
9     end
10
11     me = (a + b)/2;
12 end

```

```

1 function [mi, ma, me] = statsn(vett)
2 %Minimo, massimo, media
3     mi = Inf;
4     ma = -Inf;
5     somma = 0;
6
7     for x = vett
8         %massimo?
9         if x > ma
10            ma = x;
11        end
12
13        %minimo?
14        if x < mi
15            mi = x;
16        end

```

```
17
18     %somma
19     somma = somma + x;
20 end
21
22 %media
23 me = somma/length(vett);
24
25 end
```

10.3 Scrivere una **funzione** che, dato una coppia di numeri (assumere positivi) calcoli il valore della superficie sotto la funzione x^3 in quell'intervallo, usando il metodo dei trapezi. La funzione accetta in input anche il numero di trapezi per definire la accuratezza del calcolo.

Aiuto: Il metodo dei trapezi suddivide l'intervallo $[a, b]$ in n parti uguali di ampiezza h , definendo n trapezi, con superficie $h \frac{f(x_i) + f(x_{i+1})}{2}$. Il metodo dei trapezi approssima la superficie della funzione come la somma delle superfici di questi trapezi.

```

1 function superficie = integrale_trapezi(a, b, n)
2     h = (b - a) / n;
3     x = [a:h:b];
4     y = x.^3;
5     superfici = h * (y(1:end-1) + y(2:end)) / 2;
6     superficie = sum(superfici);
7
8 end

```

10.4 Nel gioco del *Minisudoku* bisogna riempire una tabella 4×4 con le cifre da 1 a 4, in modo tale che nessuna riga, nessuna colonna e nessuno dei quattro riquadri 2×2 contenga cifre ripetute. Ad esempio:

3		4	
	1		3
2	3		
1			2

è una tabella incompleta, ma valida. Rappresentiamo una tabella di Minisudoku con una matrice 4×4 , dove gli zeri denotano caselle vuote. Scrivere una **funzione** che, data una tabella di Minisudoku (possibilmente incompleta) dica se questa è valida o no.

```

1 function flag = unico(vett)
2 %Vero se il vettore non ha elementi ripetuti
3     flag = true;
4
5     ii = 1;
6     for x = vett(2:end)
7         if (any(x == vett(1:ii)))
8             flag = false;
9             break;

```

```
10         end
11         ii = ii + 1;
12     end
13 end
```

```

1 function flag = sudoku4(tab)
2 %Vera se la matrice e' una tabella valida di Minisudoku
   (4x4)
3   flag = true;
4   %controlla elementi
5   if (~ (0 <= tab & tab <= 4))
6       flag = false;
7       return;
8   end
9
10  %controlla righe
11  for riga_t = tab'
12      %le righe sono le colonne della
13      %matrice trasposta, trasposte
14      riga = riga_t';
15      %ignora gli zeri
16      if (~ unico(riga(riga>0)))
17          flag = false;
18          break;
19      end
20  end
21
22  %controlla colonne
23  for col = tab
24      %unicopos vuole un vettore riga
25      col_t = col';
26      if (~ unico(col_t(col_t>0)))
27          flag = false;
28          break;
29      end
30  end
31
32  %controlla blocchi
33  for ii = [1, 3]
34      for jj = [1, 3]
35          bloc = tab(ii:ii+1, jj:jj+1);
36          %bloc(1:end) trasforma la matrice 2x2
37          %in una riga 1x4
38          bloc_flat = bloc(1:end);
39          if (~ unico(bloc_flat(bloc_flat>0)))
40              flag = false;
41              break;
42          end
43      end
44  end
45 end

```