

Lista01Q02i

June 19, 2025

Lista 01

Victor Paiva Paulo Neto

Questões : 1, 2(e), 2(a), 2(g), 2(c), 2(i)

Escreva um programa e envie ao arquivo fonte ou um link do mesmo, o qual tenha como ponto de partida uma funcao $a(n)$ que define o termo geral de uma sequência numerica $a_n = a(n)$ (use n como a variavel do programa), que sejam fornecidos dois números naturais n_{min} e n_{max} que definem um intervalo $[n_{min}, n_{max}]$ para os indices “plotados” e que tenha duas opcoes exclusivas de execucao:

```
[10]: import matplotlib.pyplot as plt
import math
```

```
[6]: #funcao que gera a sequencia numerica de termo geral  $i(n) = 2(i)$ 
def i(n):
    return (n**2) / (n + 1)
```

```
[3]: #definição de  $n_{min}$  e  $n_{max}$ 
nmin = int(input("Digite o nmin: "))
nmax = int(input("Digite nmax: "))
print(f"nmin: {nmin}, nmax: {nmax}")
```

nmin: 10, nmax: 200

```
[1]: #separa nos casos em que o limite é e não é conhecido
existeLimite = (input("eh conhecido se a sequencia converge para um limite L?_
↪(a se nao, b se sim): "))
print(f"opcao escolhida: {existeLimite}")
```

opcao escolhida:

```
[11]: valoresN = []
valoresAn = []
print(f"\n{'n':>5} {'i(n)':>10}")
print("-" * 15)

for n in range( nmin, nmax +1):
    iden = i(n)
```

```

print(f"{n:>5} {i(n):>10.6f}")
valoresN.append(n)
valoresAn.append(iden)

```

n	i(n)
10	9.090909
11	10.083333
12	11.076923
13	12.071429
14	13.066667
15	14.062500
16	15.058824
17	16.055556
18	17.052632
19	18.050000
20	19.047619
21	20.045455
22	21.043478
23	22.041667
24	23.040000
25	24.038462
26	25.037037
27	26.035714
28	27.034483
29	28.033333
30	29.032258
31	30.031250
32	31.030303
33	32.029412
34	33.028571
35	34.027778
36	35.027027
37	36.026316
38	37.025641
39	38.025000
40	39.024390
41	40.023810
42	41.023256
43	42.022727
44	43.022222
45	44.021739
46	45.021277
47	46.020833
48	47.020408
49	48.020000
50	49.019608

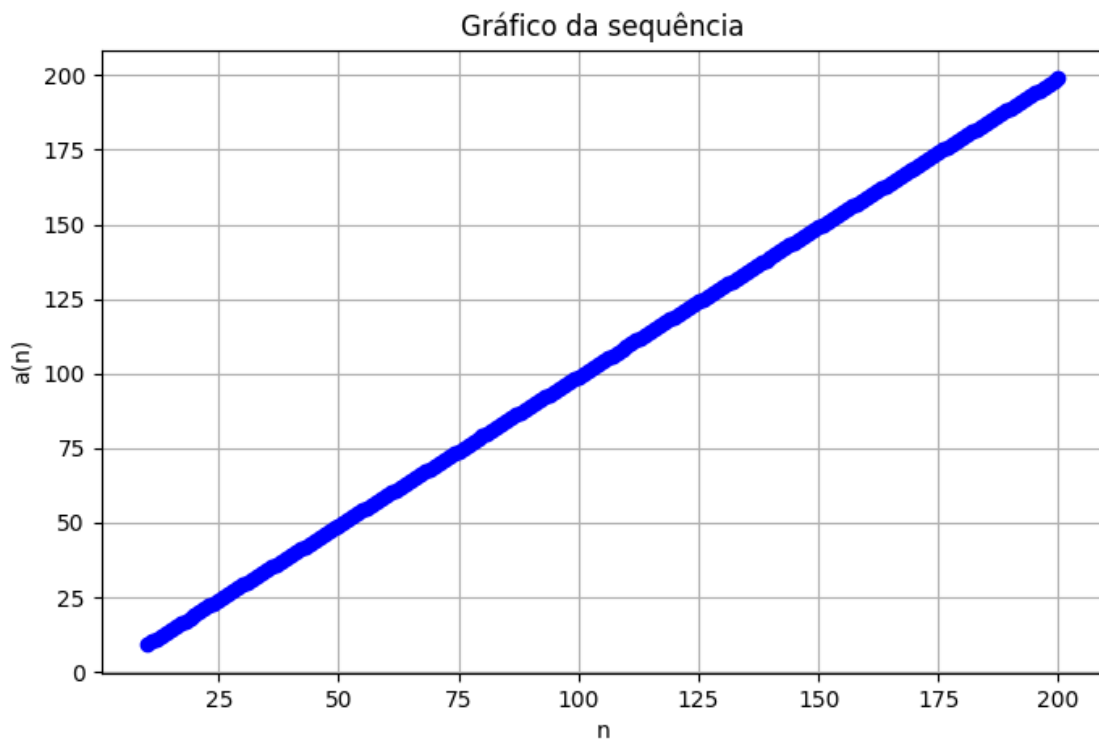
51 50.019231
52 51.018868
53 52.018519
54 53.018182
55 54.017857
56 55.017544
57 56.017241
58 57.016949
59 58.016667
60 59.016393
61 60.016129
62 61.015873
63 62.015625
64 63.015385
65 64.015152
66 65.014925
67 66.014706
68 67.014493
69 68.014286
70 69.014085
71 70.013889
72 71.013699
73 72.013514
74 73.013333
75 74.013158
76 75.012987
77 76.012821
78 77.012658
79 78.012500
80 79.012346
81 80.012195
82 81.012048
83 82.011905
84 83.011765
85 84.011628
86 85.011494
87 86.011364
88 87.011236
89 88.011111
90 89.010989
91 90.010870
92 91.010753
93 92.010638
94 93.010526
95 94.010417
96 95.010309
97 96.010204
98 97.010101

99 98.010000
100 99.009901
101 100.009804
102 101.009709
103 102.009615
104 103.009524
105 104.009434
106 105.009346
107 106.009259
108 107.009174
109 108.009091
110 109.009009
111 110.008929
112 111.008850
113 112.008772
114 113.008696
115 114.008621
116 115.008547
117 116.008475
118 117.008403
119 118.008333
120 119.008264
121 120.008197
122 121.008130
123 122.008065
124 123.008000
125 124.007937
126 125.007874
127 126.007812
128 127.007752
129 128.007692
130 129.007634
131 130.007576
132 131.007519
133 132.007463
134 133.007407
135 134.007353
136 135.007299
137 136.007246
138 137.007194
139 138.007143
140 139.007092
141 140.007042
142 141.006993
143 142.006944
144 143.006897
145 144.006849
146 145.006803

147 146.006757
148 147.006711
149 148.006667
150 149.006623
151 150.006579
152 151.006536
153 152.006494
154 153.006452
155 154.006410
156 155.006369
157 156.006329
158 157.006289
159 158.006250
160 159.006211
161 160.006173
162 161.006135
163 162.006098
164 163.006061
165 164.006024
166 165.005988
167 166.005952
168 167.005917
169 168.005882
170 169.005848
171 170.005814
172 171.005780
173 172.005747
174 173.005714
175 174.005682
176 175.005650
177 176.005618
178 177.005587
179 178.005556
180 179.005525
181 180.005495
182 181.005464
183 182.005435
184 183.005405
185 184.005376
186 185.005348
187 186.005319
188 187.005291
189 188.005263
190 189.005236
191 190.005208
192 191.005181
193 192.005155
194 193.005128

```
195 194.005102
196 195.005076
197 196.005051
198 197.005025
199 198.005000
200 199.004975
```

```
[12]: plt.figure(figsize=(8, 5))
plt.plot(valoresN, valoresAn, 'bo-', label='a(n)')
plt.xlabel('n')
plt.ylabel('a(n)')
plt.title('Gráfico da sequência')
plt.grid(True)
```



Limite:

Como demonstra o grafico, a sequencia não possui limite.

```
[13]: #funcao que verifica se o modulo da diferenca entre o termo geral e o limite é
      ↪ menor ou igual a epsilon
def N_epsilon_is_true(x, L, epsilon):
    if abs(x - L) <= epsilon:
        return True
    else:
```

```
return False
```

```
[14]: if existeLimite == "b":
    L = float(input("Digite o valor do limite L: "))
    epsilon = float(input("Digite a tolerância epsilon: "))
    N_epsilon = float(input("Digite o valor de N(epsilon): "))

    Ne = N_epsilon_is_true(a(n), L, epsilon)

    while Ne == False:
        L = float(input("Digite o valor do limite L: "))
        epsilon = float(input("Digite a tolerância epsilon: "))
        N_epsilon = int(input("Digite o valor de N(epsilon): "))
        Ne = N_epsilon_is_true(a(n), L, epsilon)

    plt.axhline(y=L, color='green', linestyle='--', label='y = L')
    plt.axhline(y=L+epsilon, color='red', linestyle=':', label='y = L + ')
    plt.axhline(y=L-epsilon, color='red', linestyle=':', label='y = L - ')

    plt.legend()
    plt.show()
```