

# Matemática Concreta - Listas de exercícios

Helder Mateus dos Reis Matos - 201904940036

11 de outubro de 2020

## Lista 1

1. Escolha uma linguagem de programação, implemente as funções de recorrência e exiba os seis primeiros termos de cada sequência. Inclua o código fonte das funções na resposta.

(a)  $a_1 = 5$  e  $a_n = a_{n-1} + 3, \forall n > 1$

```
1 #include <iostream> // std::cin e std::cout
2 #include <vector>    // std::vector
3
4 std::vector<float> recorrencia_1a (float a_1, int n){
5     std::vector<float> elementos = {a_1};
6
7     while (n > 1) {
8         elementos.push_back(elementos.back() + 3);
9         --n;
10    }
11
12    return elementos;
13 }
14
15 int main (){
16     int a_1, n;
17     std::vector<float> resposta;
18
19     std::cout << "Insira o valor de a_1: ";
20     std::cin >> a_1;
21     std::cout << "Insira o valor de n: ";
22     std::cin >> n;
23     resposta = recorrencia_1a(a_1, n);
24
25     std::cout << n << " primeiros termos da sequencia: ";
26     for (auto i: resposta) {
27         std::cout << i << " ";
28     }
29 }
```

(b)  $b_1 = 2$  e  $b_n = b_{n-1}^2, \forall n > 1$

```
1 #include <iostream> // std::cin and std::cout
2 #include <vector>    // std::vector
3 #include <cmath>     // pow()
4
5 std::vector<long double> recorrancia_1b(float b_1, int n) {
6     std::vector<long double> elementos = {b_1};
7
8     while (n > 1) {
9         elementos.push_back(pow(elementos.back(), 2));
10        --n;
11    }
12
13    return elementos;
14 }
15
16 int main() {
17     int n;
18     float b_1;
19     std::vector<long double> resposta;
20
21     std::cout << "Insira o valor de b_1: ";
22     std::cin >> b_1;
23     std::cout << "Insira o valor de n: ";
24     std::cin >> n;
25     resposta = recorrancia_1b(b_1, n);
26
27     std::cout << n << " primeiros termos da sequencia: ";
28     for (auto i: resposta) {
29         std::cout << i << " ";
30     }
31 }
```

(c)  $c_1 = 0$  e  $c_n = 2c_{n-1} + n, \forall n > 1$

```
1 #include <iostream> //std::cin e std::cout
2 #include <vector>    //std::vector
3
4 std::vector<float> recorrancia_1c(float c_1, int n) {
5     std::vector<float> elementos = {c_1};
6
7     for (int i = 2; i <= n; ++i) {
8         elementos.push_back(2 * elementos.back() + i);
9         --n;
10    }
11
12    return elementos;
13 }
14
15 int main() {
16     float c_1;
17     int n;
18     std::vector<float> resposta;
19
20     std::cout << "Insira o valor de c_1: ";
21     std::cin >> c_1;
22     std::cout << "Insira o valor de n: ";
23     std::cin >> n;
24     resposta = recorrancia_1c(c_1, n);
25
26     std::cout << n << " primeiros termos da sequencia: ";
27     for (auto i: resposta) {
28         std::cout << i << " ";
29     }
30 }
```

2. Escolha uma linguagem de programação e escreva um programa para receber uma sequência numérica e informar se a sequência é um P.A ou não. Caso seja uma

P.A, o programa deve informar se a P.A é crescente, constante ou decrescente.

```
1 #include <iostream> // std::cin e std::cout
2 #include <vector>    // std::vector
3
4 void categorizar_pa(std::vector<float> sequencia, int n) {
5     float razao = sequencia[1] - sequencia[0];
6     bool pa = false;
7
8     for (int i = 2; i < n; ++i) {
9         if (sequencia[i] - sequencia[i-1] == razao) {
10             pa = true;
11         }
12         else {
13             pa = false;
14             break;
15         }
16     }
17     if (pa == true) {
18         if (razao > 0) {
19             std::cout << "A sequencia eh P.A. crescente";
20         }
21         else if (razao < 0) {
22             std::cout << "A sequencia eh P.A. decrescente";
23         }
24         else {
25             std::cout << "A sequencia eh P.A. constante";
26         }
27     }
28     else {
29         std::cout << "A sequencia nao eh P.A.";
30     }
31 }
32
33 int main() {
34     int n;
35     float x;
36     std::vector<float> sequencia;
37
38     std::cout << "Insira o tamanho da sequencia: ";
39     std::cin >> n;
40     std::cout << "Insira os " << n << " elementos: ";
41     for (int i = 0; i < n; ++i) {
42         std::cin >> x;
43         sequencia.push_back(x);
44     }
45
46     categorizar_pa(sequencia, n);
47 }
```

3. Sabendo que o primeiro termo é igual a 3 e a razão é igual a 5, calcule o 17º termo de uma P.A.

$$a_n = a_1 + (n - 1)r$$

$$a_{17} = 3 + (17 - 1)5$$

$$a_{17} = 3 + 16 \cdot 5$$

$$a_{17} = 83$$

4. Sabendo que o primeiro termo é igual a -8 e o vigésimo igual a 30, calcule a razão da P.A.

$$a_n = a_1 + (n - 1)r$$

$$30 = -8 + (20 - 1)r$$

$$30 = -8 + 19r$$

$$r = \frac{30 + 8}{19}$$

$$r = \frac{38}{19}$$

$$r = 2$$

5. Escolha uma linguagem de programação e escreva um programa para receber os extremos de uma P.A, o valor de K e calcule a interpolação dessa P.A.

```

1  #include <iostream> // std::cin e std::cout
2  #include <vector>   // std::vector
3
4  std::vector<float> interpor_pa(float a_1, float a_n, int k) {
5      float razao;
6      std::vector<float> elementos;
7
8      razao = (a_n - a_1)/(k+1);
9      while (a_1 <= a_n) {
10         elementos.push_back(a_1);
11         a_1 += razao;
12     }
13
14     return elementos;
15 }
16
17 int main() {
18     float a_1, a_n;
19     int k;
20     std::vector<float> resposta;
21
22     std::cout << "Insira o primeiro termo da P.A.: ";
23     std::cin >> a_1;
24     std::cout << "Insira o ultimo termo da P.A.: ";
25     std::cin >> a_n;
26     std::cout << "Insira o valor de k: ";
27     std::cin >> k;
28     resposta = interpor_pa(a_1, a_n, k);
29
30     std::cout << "P.A. interpolada: ";
31     for (auto i: resposta) {
32         std::cout << i << " ";
33     }
34 }

```

6. Calcule a P.A em que a soma dos n primeiros termos é igual a  $n^2 + 2n$ .

$$S_n = n^2 + 2n$$

$$S_1 = 1^2 + 2 \cdot 1 = 3 \quad \therefore a_1 = 3$$

$$S_2 = 2^2 + 2 \cdot 2 = 8 \quad \therefore a_2 = S_2 - S_1 = 8 - 3 = 5$$

$$S_3 = 3^2 + 2 \cdot 3 = 15 \quad \therefore a_3 = S_3 - S_2 = 15 - 8 = 7$$

$$S_4 = 4^2 + 2 \cdot 4 = 24 \quad \therefore a_4 = S_4 - S_3 = 24 - 15 = 9$$

$$S_5 = 5^2 + 2 \cdot 5 = 35 \quad \therefore a_5 = S_5 - S_4 = 35 - 24 = 11$$

7. Escolha uma linguagem de programação e escreva um programa para receber uma sequência numérica e informar se a sequência é um P.G ou não. Caso seja uma P.G, o programa deve informar se a P.G é crescente, constante, decrescente, alternante ou estacionária.

```

1 #include <cctype>
2 #include <iostream> // std::cin e std::cout
3 #include <vector>    // std::vector
4 #include <cmath>     // std::floor
5 #include <string>    // std::string
6
7 float round(float num) {
8     return std::floor(num*100000.0) / 100000.0;
9 }
10
11 std::vector<float> split_parse(std::string input) {
12     std::vector<float> vec;
13     std::string numero = "";
14     float num1;
15     float num2;
16     bool divisao = false;
17
18     for (auto c: input) {
19         if (std::isdigit(c)) {
20             numero += c;
21         }
22         else if (c == ',' && divisao == false) {
23             vec.push_back(stoi(numero));
24             numero = "";
25         }
26         else if (c == '/') {
27             divisao = true;
28             num1 = stoi(numero);
29             numero = "";
30         }
31         else if (c == ',' && divisao == true) {
32             num2 = stoi(numero);
33             vec.push_back(num1*1.0/num2);
34             numero = "";
35             divisao = false;
36         }
37     }
38     if (divisao == true) {
39         num2 = stoi(numero);
40         vec.push_back(num1*1.0/num2);
41     }
42     else {
43         vec.push_back(stoi(numero));
44     }
45
46     return vec;
47 }
48
49 void categorizar_pg(std::vector<float> sequencia, int n) {
50     float q = round(sequencia[1] / sequencia[0]);
51     bool pg = false;
52
53     for (int i = 2; i < n; ++i) {
54         if ((round(sequencia[i] / sequencia[i-1]) == q) || (q == 0 && sequencia[i-1] == 0)) {
55             pg = true;
56         }
57         else {
58             pg = false;
59             break;
60         }
61     }
62
63     if (pg == true) {
64         if ((q > 1 && sequencia[0] > 0) || (q > 0 && q < 1 && sequencia[0] < 0)) {
65             std::cout << "A sequencia eh P.G. crescente";
66         }
67         else if ((q > 0 && q < 1 && sequencia[0] > 0) || (q > 1 && sequencia[0] < 0)) {
68             std::cout << "A sequencia eh P.G. decrescente";
69         }
70     }
71     else if (q == 1 && sequencia[0] != 0) {

```

```

71         std::cout << "A sequencia eh P.G. constante";
72     }
73     else if (q < 0) {
74         std::cout << "A sequencia eh P.G. alternante";
75     }
76     else if (q == 0) {
77         std::cout << "A sequencia eh P.G. estacionaria";
78     }
79 }
80 else {
81     std::cout << "A sequencia nao eh P.G.";
82 }
83 }
84
85 int main() {
86     int n;
87     float x;
88     std::vector<float> sequencia;
89     std::string input;
90
91     std::cout << "Insira o tamanho da sequencia: ";
92     std::cin >> n;
93     std::cout << "Insira os " << n << " primeiros termos da sequencia: ";
94     std::cin.ignore();
95     std::getline(std::cin, input);
96     sequencia = split_parse(input);
97
98     categorizar_pg(sequencia, n);
99 }

```

8. Escolha uma linguagem de programação e escreva um programa para receber os extremos de uma P.G, o valor de K e calcule a interpolação dessa P.G.

```

1  #include <iostream>
2  #include <vector>
3  #include <cmath>
4
5  std::vector<int> interpolacao(int a_1, int a_n, int k) {
6      std::vector<int> elementos;
7      int q;
8
9      q = pow((a_n/a_1), (1.0/(k+1)));
10     while (a_1 <= a_n){
11         elementos.push_back(a_1);
12         a_1 *= q;
13     }
14
15     return elementos;
16 }
17
18 int main() {
19     int a_1, a_n, k;
20     std::vector<int> resposta;
21
22     std::cin >> a_1 >> a_n >> k;
23     resposta = interpolacao(a_1, a_n, k);
24     for (auto i : resposta) {
25         std::cout << i << " ";
26     }
27 }

```

9. Escolha uma linguagem de programação e escreva um programa para receber uma sequência numérica. Se a sequência numérica for uma P.G, informe a produto e a soma dos termos dessa P.G. Caso contrário, informe que a sequência não é uma P.G.

```

1  #include <iostream>

```

```

2 #include <vector>
3 #include <cmath>
4
5 void soma_produto_pg(std::vector<float> sequencia, int n) {
6     float q = sequencia[1] / sequencia[0];
7     bool pg = false;
8     float soma, prod;
9
10    for (int i = 2; i < n; ++i){
11        if ((sequencia[i] / sequencia[i-1] == q) || std::isnan(sequencia[i] / sequencia[i-1])) {
12            pg = true;
13        }
14        else {
15            pg = false;
16            break;
17        }
18    }
19
20    if (pg == true){
21        prod = pow(sequencia[0], n) * pow(q, n*(n-1)/2.0);
22        soma = (sequencia[0] * pow(q, n) - sequencia[0]) / (q-1);
23
24        std::cout << "A sequencia eh uma P.G.\n";
25        std::cout << "Sua soma eh: " << soma << "\n";
26        std::cout << "Seu produto eh: " << prod << "\n";
27    }
28    else std::cout << "A sequencia nao eh uma P.G.";
29 }
30
31 int main() {
32     int n;
33     float x;
34     std::vector<float> sequencia;
35
36     std::cin >> n;
37     for (int i = 0; i < n; ++i){
38         std::cin >> x;
39         sequencia.push_back(x);
40     }
41     soma_produto_pg(sequencia, n);
42 }

```

10. Determine o valor de  $n$  tal que  $\sum_{i=3}^n 2^i = 4088$