

Lista de exercícios II: Funções e Recursividade; Ponteiro genérico e ponteiro de função; e Sobrecarga de Função

Notas:

1. Para cada questão de implementação em C++:
 - 1.1. Separe o código (*.cpp) das definições das funções, estruturas e constantes (*.h);
 - 1.2. Crie sempre um arquivo main.cpp e utilize a função principal para demonstrar o funcionamento da sua implementação. Sempre que possível, crie um arquivo de casos de teste (case.txt) e o utilize para testar exaustivamente sua implementação;
 - 1.3. Faça uso das boas práticas de programação que você já conhece.
 - 1.4. O código deverá ser compilado e devidamente testado.

Q01. Desenvolva funções recursivas como solução para os seguintes problemas e indique qual tipo de recursão está sendo utilizada em cada um dos algoritmos:

- a. Dado um valor N , calcular o valor da sequência:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$$

- b. Dado um valor n , calcular o valor da sequência:

$$\frac{2}{4} + \frac{5}{5} + \frac{10}{6} + \frac{17}{7} + \frac{26}{8} + \dots + \frac{(n^2+1)}{(n+3)}$$

Q02. Converta as funções anteriores para uma versão iterativa.

Q03. Escreva um programa que contenha uma função sobrecarregada chamada **min** para determinar o menor de dois parâmetros. Teste o seu programa usando pares de números inteiros, de caracteres e de números em ponto flutuante.

Q05. Escreva um programa que contenha a função sobrecarregada **somatório()** que atenda aos seguintes critérios:

- Caso receba apenas um vetor de inteiros \mathbf{v} , deverá retornar o resultado da expressão:
$$v_1 + v_2 + v_3 + \dots + v_n$$
- Caso receba, além do vetor de inteiros \mathbf{v} , um segundo parâmetro inteiro \mathbf{x} , deverá retornar o resultado da expressão:
$$\frac{v_1}{x} + \frac{v_2}{2x} + \frac{v_3}{3x} + \dots + \frac{v_n}{nx}$$
- Caso receba, além do vetor de inteiros \mathbf{v} e do parâmetro \mathbf{x} , um terceiro parâmetro inteiro \mathbf{y} , deverá retornar o resultado da expressão:
$$\frac{v_1}{x-y} + \frac{v_2}{2x-y} + \frac{v_3}{3x-y} + \dots + \frac{v_n}{nx-y}.$$

Considere **y** como tendo o valor padrão (default) de 0 (zero).

Q06. Escreva uma função chamada **quadrado** que mostre, na margem esquerda da tela, um quadrado de caracteres cujo lado é especificado por um parâmetro do inteiro lado. Sua função deve permitir que o usuário não especifique um caractere, o que nesse caso, utilizará o caractere * como padrão. Por exemplo, a chamada **quadrado(4)** exibe:

```
****
****
****
****
```

enquanto que a chamada **quadrado(4, 'a')** exibe:

```
aaaa
aaaa
aaaa
aaaa
```

Q07. Escreva uma função para manipulação de vetores de inteiros com a seguinte assinatura:

int faztudo_vetor (int coleção [], <tipo de operação> operacao);

Onde, o tipo de operação deve ser implementado como uma enumeração para indicar as operações:

opMax – retorna o maior valor presente na coleção

opMin – retorna o maior valor presente na coleção

opSum – retorna a soma de todos os valores presentes na coleção

opAvg – retorna o valor médio (a parte inteira da media) de todos os valores presentes na coleção.

opSiz – retorna a quantidade de elementos no vetor

Implemente cada operação como uma função. A chamada da função referente à operação deve ser realizada através de um ponteiro de função.

Q08. Sobrecarregue a função **faztudo_vetor** da Q07 para permitir a operação sobre um vetor de reais (double). Considere ainda que a operação padrão (default) deve ser opSiz.

Q09. Que valores serão impressos pelo programa listado abaixo? Para cada resultado impresso pelo programa, justifique a sua resposta, baseado nos conceitos discutidos em sala de aula.

```

1  #include <iostream>
2  #include <bitset>
3
4  using namespace std;
5
6  int somaA (int a, int b)
7  {
8      a++;
9      int result = a + b;
10     return result;
11 }
12
13 int somaB (int &a, int b)
14 {
15     a++;
16     int result = a + b;
17     return result;
18 }
19
20
21 void somaC (int a, int b, int *result)
22 {
23     a++;
24     (*result) += a + b;
25 }
26
27

```

```

27
28 void somaD (int a, int b, int &result)
29 {
30     a++;
31     result += a + b;
32 }
33
34 void somaE (int a, int b, int result)
35 {
36     a++;
37     result += a + b;
38 }
39
40 int main(int argc, char* argv[])
41 {
42     int arg1 = 5;
43     int arg2 = 6;
44
45     cout << somaA(arg1, arg2) << endl;
46     cout << somaB(arg1, arg2) << endl;
47
48     int resultado = 0;
49     somaC(arg1, arg2, &resultado);
50     cout << resultado << endl;
51
52     somaD(arg1, arg2, resultado);
53     cout << resultado << endl;
54
55     somaE(arg1, arg2, resultado);
56     cout << resultado << endl;
57
58     return 0;
59 }

```

FIM DA LISTA.