

Lista de Exercícios - Erros de Ponto Flutuante

Prof. Andréa Maria Pedrosa Valli

Data de Entrega: dia 28 de março de 2019.

Envio: Envie apenas os arquivos `.cpp` (ou `.c`) e os arquivos de saidas. Envie por email com o assunto “Lista0”. Lista de exercícios **opcional** para os alunos dos cursos de matemática e física.

1. Compile o código `loop.cpp`, execute e explique o que acontece e porque. Coloque as suas conclusões como comentários no código. Como podemos implementar o teste ($x \neq 1.0$) no comando **while** para que o programa funcione adequadamente? O loop deve terminar quando x é igual a 1.0, ou seja, não pode modificar o teste do loop para terminar quando $x < 1.0$, por exemplo. Qual é a saída do código corrigido?
2. Compile e execute o código `def_vetor.cpp`. Apesar do código executar corretamente, existe um erro na definição do vetor que pode trazer problemas de invasão de memória e no qual o compilador não consegue detectar. Conserte o código para definir uma constante para estabelecer o tamanho do vetor. Esta constante passa a ser usada nas referências ao tamanho do vetor. Caso seja necessário trocar o tamanho do vetor basta alterar o valor da constante no código principal.
3. Faça um outro código de nome `def_tam_vetor.cpp` para definir o vetor do exercício anterior, em tempo de execução, de tamanho n fornecido pelo usuário através do teclado. Não esqueça de liberar a memória alocada antes de terminar o programa.
4. Para o código `def_vetor.cpp` consertado, modifique o programa para imprimir, em cada linha de um arquivo de nome “saida.txt”, os índices do vetor e os seus valores.
5. Faça um código para definir e testar uma função de nome “adicao” que calcula a soma de dois valores do tipo **double** e retorna a adição também do tipo **double**. A saída do seu código deve ser feita no programa principal e não dentro da função “adicao”. Mostre na tela do computador os valores somados e o resultado da soma, usando a função definida no seu código.
6. Faça um código para definir e testar um procedimento (**void**) de nome “duplica” que possui dois argumentos do tipo **double** e duplica estes valores. Os argumentos devem ser passados por referência. Construa um programa para testar o seu procedimento, mostrando na tela os valores iniciais e depois os valores duplicados, utilizando o seu procedimento. A saída do código tem que ser feita no programa principal, ou seja, não pode ser feita dentro do procedimento “duplica”.