

Como fazer o código

- 1) Colocar parêntese para indicar a precedência, mesmo quando trivial:
- 2) Não usar igualdade com variáveis de ponto flutuante:
- 3) Não fazer repetição de código, use uma função para substituir a repetição:
- 4) O valor da variável é constante?
- 5) Os switch possuem um default?
- 6) Algumas das variáveis pode não conter valores negativos?
- 7) Suas funções estão grandes?
- 8) É possível refatorar o código?
- 9) O nome das variáveis fazem sentido?
- 10) Existem variáveis ou funções que não estão sendo utilizadas?
- 11) Existem mensagens de erro sendo printadas na tela?
- 12) Todas as condições de erro estão sendo levadas em consideração?
- 13) Todos os loops tem uma condição de saída?
- 14) Use $a += b$ ao invés de $a = a + b$
- 15) O ponteiro é retornado pela função quando na verdade deveria ser o valor?

OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK

Assertivas

- 1) Ao terminar o escopo de um if é aconselhável comentar que está saindo:
- 2) As unidades de medida estão comentadas?
- 3) As entradas da função estão comentadas?
- 4) A utilidade da função está comentada?
- 5) As saídas/retorno das funções estão comentadas?
- 6) Há uso de assertivas de entrada e saída?
- 7) Comentar as condições booleanas complicadas
- 8) Se for utilizados flag comente ao lado delas sua função e o significados dos seus valores
- 9) Seu código utiliza-se de invariantes?
- 10) Seus código tem funções com exceções?
- 11) Suas condições de paradas de loop estão comentadas?
- 12) As variáveis das estruturas estão comentadas?
- 13) A funcionalidade do módulo está descrita?
- 14) Seu módulo está assinado?
- 15) As partes do código de difícil interpretação estão comentadas?

OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK

Como testar

- 1) Evitar repetição de testes:
- 2) Evitar ao máximo que um teste interfira em outro:
- 3) Automatizar os testes:
- 4) Utilize uma framework de teste;
- 5) Sempre que mudar algo significativo no código rode o teste novamente;
- 6) Todos os casos de erros estão cobertos pelos testes?
- 7) Os casos de teste estão especificados?
- 8) Os erros estão sendo tratados de maneira adequada?
- 9) Os casos estão bem identificados?
- 10) As entradas e retorno dos testes estão documentadas?
- 11) Os testes estão de fácil entendimento?
- 12) Os testes são finalizados quando há um erro de um artefato fundamental?
- 13) Os testes estão assinados?
- 14) Os testes estão fáceis de serem executados novamente?
- 15) Os testes possuem certa hierarquia?

OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK

Como debugar

- 1) Não utilize-se de printf ao longo do seu código
- 2) Use gdb
- 3) As variaveis estao sendo inicializadas corretamente?
- 4) As condições booleanas estão corretas?
- 5) A ordem dos argumentos nas funções estão corretos?
- 6) Alguma variável pode está dando overflow?
- 7) As variáveis tem nomes parecidos estão confundindo?
- 8) Tem alguma divisão de inteiros quando na verdade era esperado o resultado float?
- 9) Os arquivos estão sendo fechados depois de utilizados?
- 10) Cheque a parte a modificação mais recente;
- 11) Leia o código antes de tentar consertar erro
- 12) Explique seu código para outra pessoa
- 13) Desenhe um diagrama lógico
- 14) Use a estratégia *Divide and Conquer*
- 15) Debugue agora não depois

OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK