Nota: 4,5 (quatro e meio ) Atraso: Grupo de 1:

Grupo:

Gabriel Amaral Fuchs (1521531) - gafuchs@gmail.com

João Pedro Garcia (1621161) - jpmldg@gmail.com

Marcus V. B. Siqueira (1621381) - marcusviniciusb.siqueira@hotmail.com

Todos os arquivos identificam os autores (não: -1) OK

Relatórios (não: -1) OK

1 – (2 ponto) Correção do módulo fornecido, usando script manual OK

2 – (3 pontos) Correção do módulo e do gerador de casos de teste fornecidos -3

Copiaram o tst-smt.cpp de uma versão anterior, sim. Ver próxima.

3 – (1 ponto) Completar o gerador de casos de teste fornecidos -1

Copiaram o tst-smt.cpp de uma versão anterior, sim.

Código exatamente igual. Inclusive as mensagens de erro, a formatação, o nome das variáveis locais.

A única alteração com relação ao original é CountRetrieve = -1; Indicando que não entenderam o objetivo do contador.

Usei diff para fazer a comparação do original e o do enunciado, e inspeção visual do código original e do fragmento adicionado.

4 – (2 pontos) Gerar clone de SMT Cancelada

5 – (2 pontos) Medir eficácia usando mutantes -1,5

A ideia com relação aos operadores de mutação está OK.

A implementação deixa a desejar, o código precisa ser compilável

Existem duas soluções

- uma é compilar n programas, cada um contendo uma mutação

- outra é escrever um if( idMutação ){ código mutado} else { código original}

Não vieram os executáveis para os mutantes. No caso, não dá para saber se compila corretamente cada um dos mutantes, nem sequer se o teste mata todos eles.