U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Teste, Verificação e Validação de Software

Exercício Prático

João Bordalo – ei12040
João Pereira – ei12023

4 de outubro de 2016

1 GDC

a) Através da ferramenta *Pitclipse* identifique os mutantes presentes na função ***GDC*** da classe ***Algorithm*****.**

```
public int gcd(int x, int y) {
    int tmp;
    while (y != 0) {
        tmp = x % y;
        x = y;
        y = tmp;
    }
    return x;
}
```

Figura 1 - Função gdc

b) Encontre a melhor solução com o método de eliminação de mutantes.

c) Teste a sua solução com a ferramenta *Pitclipse* e verifique se é possível eliminar os mutantes.

2 Min

a) Através da ferramenta *Pitclipse* identifique os mutantes presentes na função ***Min*** da classe ***Algorithm*****.**

```
public int Min(int x, int y) {
    int v;

    if (x < y)
        v = x;
    else
        v = y;

    return v;
}
```

Figura 2 - Função min

b) Encontre a melhor solução com o método de eliminação de mutantes e verifique com a ferramenta *Pitclipse* se é possível eliminar os mutantes.

c) O que conclui dos mutantes da função ***Min*** da classe ***Algorithm*****?**

3 NumZero e NegateArray

a) Através da ferramenta *Pitclipse* identifique os mutantes presentes na função ***NumZero*** e ***NegateArray*** da classe ***Algorithm*** e encontre possíveis soluções de teste capazes de os eliminar.

```
public int numZero(int[] x) {
    int count = 0;

    for (int i = 0; i < x.length; i++)
        if (x[i] == 0)
            count++;

    return count;
}

public void negateArray(final float i, float a[]) {
    for(int k = 0; k < a.length; k++)
        a[k] = a[k] * (-i);
}
```

Figura 3 - Função numZero e negateArray

Bom trabalho!