



Node coverage

RT= {1,2,3,4,5,6,7,8,9,10}

· Caminos de test:

1) [1,2]

2) [1, 3, 4]

3) [1,3,5,6,10]

4) [1,3,5,6,7,8,9]

3) 4º return:

List (String > List1 = new Array List (String>(); List < String > list 2 = new Army List < String > (); assume False (list1. equals (list2));

1) 1 gr return :

List < String > List = new Array List < String>();

List 1. add ("100");

assume True (list1. equals(list1));

2) 2º retum:

List < String> list | = new Amoublet < String>();

int list2 = 8

List 1. add ("Loo"):

assume False (list ! equals (list 2));

4) 3st return:

List < String > list 1 = now Armyllst < 9ting >1); list< String > list2 = new Array list < String >1);

List 1. add (" 100");

list 2. add ("box");

assume False (list. equals (list2));

EP coverage

RT= {(1,2),(1,3),(3,4),(3,5),(5,6), (67),(6,10),(7,8),(8,9),(8,6)} · Cominos de test:

1) [1,2] -> Ignal que 1) NC

2) [1,3,4] - Ignel que 2) NC

3) [1,3,5,6,7,8,9] - Zend gue 4)NC

4) [1,3,5,6,7,8,6,10]

4) 4º return:

List <String> list! = new Arraylist <String>(); List <String> list? = new Arraylist < String>(); listlado ("foo");

list2. add ("foo"); assume True (list1. equals(list2));

15 Prime path

RT = { [1,2], [1,3,4], [1,3,5,6,7,8,9], [6,7,8,6], [1,3,5,6,10], [8,6,7,8], [8,6,7,8], [6,7,8,9]}

· Caminos de test:

1) [1,2] - Igual que 1) NC

2) [1,3,4] - Ignal que 2) NC

3) [1,3,5,6,7,8,6,7,8,6,7,8,9]

9 [1,3,5,6,7,8,6,10] → Zgood que 4) EPC

3) 3er return dondo dos melta al while:

list < String > list ! = new Array List < String > !!; List < String > ! list 2 = new Array List < String > !!;

list 1. add ("100");

list 1. add ("bar");

list ? add ("word");

list 2. add ("100");

hist 2 add ("bar");

list 2 add ("line");

assume False (list 1. equals (list2));