**WEA150022 ONG JIA AUN**

**Tutorial 13**

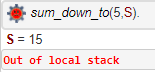
1. **The predicate defined below causes the sum of integers from 1 to N to be calculated and returns the answer as the value of SUM. It appears to be correct, but it has a serious flaw.** sum\_down\_to(1,1).

sum\_down\_to(N, SUM):- N1 is N-1,

sum\_down\_to(N1, SUM1),

SUM is SUM1+N.

1. **Demonstrate that this definition is incorrect by entering a query such as ?- sum\_down\_to(5, S). and then use backtracking to force Prolog to find more than one solution.**



After getting the first solution S=15,

Prolog will backtrack to get the second solution by using going for the second rule again and this time N = 1, thus N1 = 0. The first rule will never fire, hence the query will never stop, leading to stack overflow.

1. **Correct the program using two different techniques to resolve the above issue.**

**Solution 1:**

sum\_down\_to(N, SUM):- N1 is N-1, **N1>0,**

sum\_down\_to(N1, SUM1),

SUM is SUM1+N.

Reason:  
By having a checking to make sure that N1 is always larger than 0, will ensure the program will not run out of stack.

**Solution 2:**

sum\_down\_to(1,1)**:- !**.

Reason:

By cutting the prolog query once firing the first rule will prevent it from going to the second rule, hence solving the flaw above. This is a better solution as it makes the program perform better without changing its use.