

MSc in Software Engineering and Database Technologies

CT621 Artificial Intelligence

**Student Name:** Cristina Borges

**NUIG ID Number:** 20234955

**Course:** CT621 Artificial Intelligence

**Workshop No:** Week 3

**Assignments:** Week 3

**Date of Submission:** 06/06/2021

Contents

1.0 Assignment 2

1.1 Question 1 3

Program Summary 3

Issues encountered and solutions 3

Conclusion 4

1.2 Question 2 5

Program Summary 5

Issues encountered and solutions 5

Conclusion 6

Appendix 1 7

# Assignment

**Question 1:**

Write a Prolog predicate that takes two arguments. The first argument is a list and the predicate should return the list reversed as the second argument. For example:

**reverse ( [a,b,c,d] , X).**

**X = [d,c,b,a]**

How flexible is your code? What would your code return when given the following query?

**reverse (X, [a,b,c,d]).**

**Question 2:**

Write a recursive Prolog predicate that accepts two numbers X, Y as input. It should output to screen a list of all numbers less than or equal to X that are evenly divisible by Y.

**Note:** there may be no numbers that meet this criteria. Assume X and Y are both positive integers.

## 1.1 Question 1

## Program Summary

I have used the SWI Prolog IDE to write a short program that allows a user to enter a list which is then mirrored/reversed and combined into a new empty list which is then printed to the console.

From my reading, these pieces of code should take a general form of:

1. Deal with empty list
2. Specify head & tail & where Head is dealt with
3. Specify where tail is dealt with.(if relevant).
4. Additional steps needed

The code I have prepared is as follows

mirror\_list([],[]).

mirror\_list([H|T],A) :-

mirror\_list(T,B),

concatenate (B,[H], A).

concatenate([], H, H).

concatenate ([H|T],A,[H|B]):-

concatenate (T,A,B).

Program can be explained as follows:

First line specifies that if an empty list is input, an empty list is returned,

Second line defines that the reversal takes place in element A.

Third line specifies that the rest (Tail) is reversed in element B.

Fourth line, specifies that we combine the reversed list stored in element B with the first element stored in Head – referred to Concatenation of Lists in Prolog – subsequently found further reading on append which may have been easier to write

Line 5 is for the case of an empty list where if an empty list is entered, an empty list is returned.

Line 6 and 7 is the combination of the elements in element A for display in element B

## Issues encountered and solutions

I ran a number of test cases and the errors encountered included:

1. Error due to missing full stop, bracket or variable when inputting query. Examples include:
   1. Missing full stop
   2. Missing variable X
   3. Incorrect query command
2. Using Heads & Tail (H & T) notation along with other variables was confusing – if I was starting again I’d just use A, B, C and D.

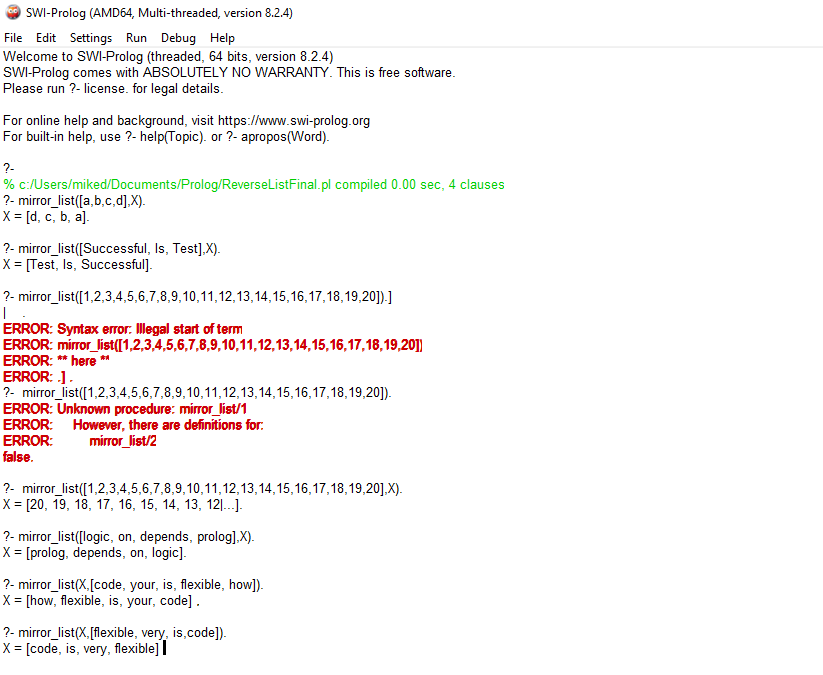
The following table summarises the tests ran:

|  |  |  |  |
| --- | --- | --- | --- |
| Query input | Output | Correct Y/N | Comment |
| Mirror\_list([a,b,c,d],X). | d, c, b, a | Yes | None |
| Mirror\_list([Successful, Is, Test],X). | Test Is Successful | Yes | None |
| Mirror\_list([1,2,3….18,19,20],X). | 20, 19, 18….3, 2, 1 | Yes | None |
| Mirror\_list([1,2,3….18,19,20]). | Error | No | Missing “X” |
| Mirror\_list([1,2,3….18,19,20]X). | Error | No, | Missing comma |
| Mirrir\_list(([a,b,c,d],X). | Error | No | Typo in command |
| Mirror\_list(code, your, is, flexible, how) | how, flexible, is, your, code | Yes | None |

Code returns result when command is done as:

Mirror\_list([list],X). or Mirror\_list(X,[List]).

This suggests the code has sufficient flexibility.



## Conclusion

Program works as expected.

## 1.2 Question 2

## Program Summary

The objective is to write a program where the user enters a number (X) and the program returns a list of numbers between 0 and X that are divisible by another number (Y).

From my reading, these pieces of code should take a general form of:

1. Deal with empty list
2. Specify head & tail & where Head is dealt with
3. Specify where tail is dealt with.(if relevant).
4. Additional steps needed

I firstly write a short program which generates a list.

Num\_list(0,[0]).

Num\_list(X,[H|T]) :-

H is X,

N is X-1,

Num\_list(N,T).

Program can be explained as:

First line deals with the case with zero

Second line defines that number entered by user is X is to return a list between Head and Tail

Third line specifies that Head is equal to X

Fourth line specifies that N is equal to X-1

Fifth line returns a list between X-1 and Tail.

The second program requires the user to enter a list and a divisor so it will then return a new list of numbers divisible by that number.

A simple division program could be written as:

Divisible (X,H):-

**H mod X=.=0**

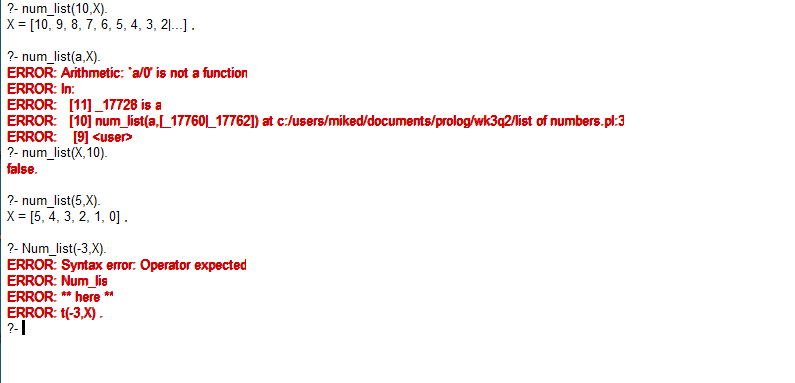
**Query to use would be**

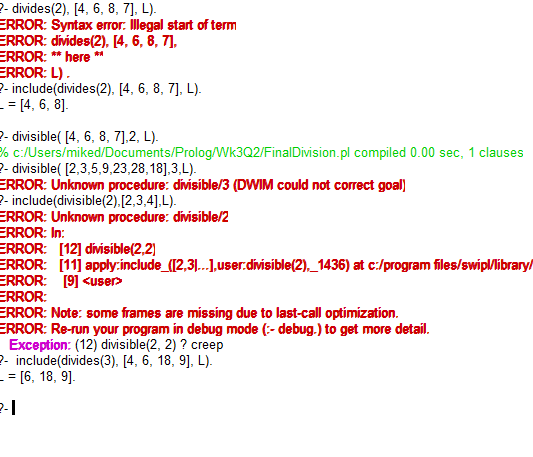
?- include(division(Y),[Num\_List], Result.

## Issues encountered and solutions

While I was able to create each stage of the program individually, I was unable to combine the two. The two changes I need to make are:

1. Division function should look at the result generated by Num\_list(A,X) instead of requiring user to enter list.
2. Query should be something like:
   1. ?- include(division(Y),[Num\_list(A,X), List).





## Conclusion

Unable to finalise program so cannot confirm it works as expected.

## Appendix 1

**Assignment 1**

**References**

SWI Prolog Predicate append/2

Available at: <https://www.swi-prolog.org/pldoc/man?predicate=append/2>

Accessed 5th June 2021

SWI Prolog Predicate append/3

Available at: [https://www.swi-prolog.org/pldoc/man?predicate=append/3](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.swi-2Dprolog.org_pldoc_man-3Fpredicate-3Dappend_3&d=DwMFaQ&c=3NBXXUKukgVIjVXwt0Rin6h0GAxIKZespWWvcJx4w9c&r=PyhisvEo9RBKGEQtQ1eYI0TixWoSLkcudJP2T2Qw1sANtEbc-bl34GM05l5SMjzK&m=UI9-k5hyhCHw1IOTYr1awF4OY7oG47YYKg3Gp1dteNE&s=k-X2RjUMRaa00oQ-F31ReKf1smvYyE88JwdMW3OTZmQ&e=)

Accessed 5th June 2021

Concatenation of Strings in Prolog

Available at: <https://stackoverflow.com/questions/33726463/concatenation-of-strings-in-prolog>

Accessed 5th June 2021

Concatenation of Lists in Prolog

Available at: <https://stackoverflow.com/questions/9348640/concatenation-of-lists-in-prolog>

Accessed 5th June 2021

reverse(+List, ?Reversed)

Available at: [http://eclipseclp.org/doc/bips/lib/lists/reverse-2.html](https://urldefense.proofpoint.com/v2/url?u=http-3A__eclipseclp.org_doc_bips_lib_lists_reverse-2D2.html&d=DwMFaQ&c=3NBXXUKukgVIjVXwt0Rin6h0GAxIKZespWWvcJx4w9c&r=PyhisvEo9RBKGEQtQ1eYI0TixWoSLkcudJP2T2Qw1sANtEbc-bl34GM05l5SMjzK&m=UI9-k5hyhCHw1IOTYr1awF4OY7oG47YYKg3Gp1dteNE&s=3brL7JssY9JnPBeY5c580uH2tX3HWtCd_2wTZQZ1Kts&e=)

Accessed 5th June 2021

Numbers in a list smaller than a given number

Available at: <https://stackoverflow.com/questions/32260202/numbers-in-a-list-smaller-than-a-given-number>

Accessed 5th June 2021