!! Comentariile se notează cu % !!

9.a

Model matematic:

Insert([1,2,3],5,7)

Insert([2,3],5,6)

Insert ([3],5,5)

Where l1…..ln is our list, e is the value we want to insert and p is the position on which we want to insert.

Also we will call it like where R will e the result.

means we have 3 inputs and one output.

Case 1: p = 0

Linia de cod care se ocupa cu acest caz:

– construieste ca rezultat o noua lista unde elemental o sa fie asezat in fata tuturor elementelor din lista data ca input

! - It's used to prevent backtracking for this specific clause. Once a solution is found, the cut ensures that Prolog doesn't try to find alternative solutions for this case, making the program more efficient. (Because when a clause is satisfied, Prolog will try to find additional solutions by backtracking and trying different combinations)

Case 2: we insert on any other position:

Cod afferent:

This is a recursive clause that decomposes the list into H (head) and T (tail).

In the first part: [H|T] is a list pattern that means H is the first element of the list and T is the rest of it, then we have E the element we want to insert and P the position. [H|R] constructs the result with H as the head of it and R the rest of the tail.

P1 is P – 1 = decrements the position by one.

Last part: is the recursive call, processes the tail and attempts to insert E at the updated position. This recursive call will effectively iterates through the list, decrementing the position (until reaches the base case).

9.b

Model mathematic:

gcd(A: number, B: number, R:number)

gcd(i,i,o)

We have 2 parameters a and b, the numbers for which we want to compute gcd, and r the result that will also be a number. And 2 inputs and one output.

What the program does?

Here we implement the fact that the gcd of 0 and B is B. (for ! see above)

Here is the fact that the gcd of A and 0 is A.

This is the third case, when a >= b. It calculates the new value of A and recursively calls gcd with the updated values. The result is stored in R.

Forth case, when a < b, computes the new value of b (b%a) and the recursively calls gcd for the updated values. Result also stored in R.

Using gcd now wecan compute the greatest common divisor of all the elements from a list.

Model mathematic:

Case 1, n = 1:

Resolved in where is stated that the greatest common divisor of a list containing a single element is the element itself.

Case 2, n > 1:

Here [H|T] means that H is the first element of the list and T the rest of it, R1 is the gcd of T (here is the recursive call) and after, in R is stored de gcd of all the elements.