Assignment #2 on Prolog

Date Due: November 3, 2025

Total: 100 marks

Include for all programs both rules/code and execution. Don't use code from the internet. Write your own code.

Please use only the type of constructions that you see the slides/videos/examples. Don't use built-in predicates that will solve the problem for you[1]; that will defeat the purpose of the assignment.

Don't use predicates that emulate procedural programming constructions like for loops(any of them: logical or counter control loops) or conditionals(all kind). Use just the BNF syntax present in the slides – no other extensions. You need to think in Prolog, not in a procedural programming language. By using procedural programming constructions you'll get 10% of the mark or 0.

The content of the slides is enough to complete the assignment. You must send the code(text) together with instructions of how to run the programs (the `Readme` file), and a text capturing the execution (the `Run.txt` file) (no binary files!)[2]. Just use the general format required for all assignment submissions, as it is described in the slides.

1. (10 marks) Write the corresponding program for the following predicate:

   ```
   translate(L1,L2).
   ```

   to translate a list words naming digits between 0 and 9 into a list of corresponding digits.

   For example:

   ```
   translate([one,two,nine],[1,2,9]).
   ```

2. (15 marks) Write a predicate to interactively guess an even number from 1000 to 3000 in at most k tries. The number k is read from standard input. In case the user types an odd number or the number is out of bounds, the number of tries is reduced by an additional try.

[1] Some versions of Prolog may have that predicates built in

[2] pictures/screenshots are binary files

After each try, the number of remaining tries should be displayed. Outputs can be obtained with predicates `write(X)` and `nl`, and an input with the predicate `read(X)` (integer input will be followed by a “.”). The secret number should be hardcoded.

For example:

```
game.
6.
You have 6 tries to guess a number between 1000 and 3000
6.
You have 4 tries to guess a number between 1000 and 3000
2000.
The number is higher, 3 tries remaining.
2500.
You guessed right.
yes.
```

3. (20 marks) Write a predicate to interactively guess an element from two lists (stored internally) in at most k tries, where k is also an input (integer) value. The program should precalculate the maximum and minimum number from both lists and should tell the user if the number is out of bounds. Therefore, you input k followed by at most k tries. The predicate will be true if you guess a number from either lists and false otherwise.

For example:

```
game.
Number of tries:
4.
You have 4 tries to guess a number from the two lists.
60.
You have 3 tries to guess a number from the two lists.
3001.
The number is out of bounds, 2 tries remaining.
2502.
You guessed right.
```

4. (20 marks) Write a predicate that computes the sum of all even numbers contained in a list `L` and are between two other natural numbers. The list is hardcoded internally, but the two numbers are read from standard input.

5. (25 marks) Write a predicate that computes as a parameter a sublist of a list stored internally (harcoded), starting with the k-th element. The number k is an integer that is read from standard input. If k is negative the program will return the last $-k$ elements of the list. If $-k$ is grater than the length of the list it will return the whole list. If k is positive and greater than the length of the list it will return the empty list.

6. (25 marks) Write a predicate to interactively compute the price for a menu. You should have at least 5 questions for menu choices, accept quantities, then compute the total amount and display it. In case the quantity required for a certain item exceeds the amount available, it should be adjusted to the maximum available and the user warned about the adjustment being made.