

Basic Arithmetic in Prolog

Introduction

This program will allow the user to store one value per variable and perform basic arithmetic functions on those variables. The 5 arithmetic functions within the program are addition, subtraction, multiplication, division, and exponents.

The purpose of building this program was to build familiarization with global variables and basic arithmetic in Prolog.

Syntax

At the end of every Prolog statement, be sure to enter a period. The period represents the end of a statement. If the user wants to perform multiple tasks at once, commas may be used to separate tasks, but this is not recommended.

The following list will demonstrate how to perform each action within the program.

- `consult('File_Name')`. Loads program into SWIPL.
- `declare(VN, V)`. Gives value `V` to variable with name `VN`.
- `bind(VN, V)`. Gives value `V` to variable with name `VN`.
- `valueOf(VN, V)`. If value `V` is stored in variable name `VN`, return true.
- `undeclear(VN)`. Remove variable name `VN`'s value.
- `bindings`. Returns all variable names along with their respective value.
- `inc(VN)`. Add one to the value stored in variable name `VN`.
- `dec(VN)`. Subtract one from the value stored in variable name `VN`.
- `add(V1, V2, OPN)`. Take the values stored in variable `V1` and variable `V2` and store the sum into variable `OPN`.
- `sub(V1, V2, OPN)`. Take the values stored in variable `V1` and variable `V2` and store the difference into variable `OPN`.
- `mul(V1, V2, OPN)`. Take the values stored in variable `V1` and variable `V2` and store the product into variable `OPN`.
- `div(V1, V2, OPN)`. Take the values stored in variable `V1` and variable `V2` and store the quotient into variable `OPN`.
- `pow(V1, V2, OPN)`. Take the values stored in variable `V1` and variable `V2` and store the power into variable `OPN`.
- `halt`. Close the program.

Instructions

The first step in the program is to actually start the program. With the appropriate prolog tool, such as SWIPL, use `consult()` to open `gv2.pro`. Be sure to enter the entire file path. If the file path is correct, `true` should be returned.

```
consult('~documents/.../gv2.pro').
```

Now, the user needs to declare values for variables.

```
declare(x, 10).
```

```
declare(y, 3).
```

The user is now free to perform any of the above rules using the variables declared. Some examples are given below.

```
add(x, y, sum).
```

```
sub(x, y, difference).
```

```
pow(y, x, power).
```