

David Alejandro Velázquez Valdéz A01632648

Felix David De Haro Soto A01637589

Diego Ortiz Mariscal A01552000

Actividad 5.1 Programación Lógica

2. Write a predicate `deconsonant/1` to print any element of a list that isn't a consonant (i.e. we want to print out the vowels `fa,e,i,o,u`). It should always succeed provided it is given a list as its argument (we assume that the input list only contains vowels and consonants).

The screenshot shows the SWISH Prolog IDE. On the left, the code editor contains the following Prolog code:

```
1 deconsonant([]).
2 deconsonant([_:_]).
3 vowel(a).
4 write(a),
5 deconsonant(_).
6 deconsonant([_:_]).
7 deconsonant(_).
8 vowel(o).
9 vowel(e).
10 vowel(i).
11 vowel(o).
12 vowel(u).
13
14
```

On the right, the execution console shows the results of several queries:

- `deconsonant([a, e, c, d])`: Singleton variables: [A], true. Next: 10 100 1,000 Stop.
- `deconsonant([r, b, c, d])`: Singleton variables: [A], true.
- `deconsonant([a, b, l, d])`: Singleton variables: [A], true.
- `deconsonant([a, e, i, o])`: Singleton variables: [A], true.

7. Write a predicate `fact/2` which takes a natural number as first argument and returns the factorial of the number.

`fact(0, 1).`

`fact(X, R) :- X > 0, X1 is X-1, fact(X1, R2), R is R2*X.`

The screenshot shows the SWISH Prolog IDE. On the left, the code editor contains the following Prolog code:

```
1 fact(0, 1).
2 fact(X, R) :- X > 0, X1 is X-1, fact(X1, R2), R is R2*X.
```

On the right, the execution console shows the results of several queries:

- `fact(1, X)`: X = 1, false.
- `fact(2, X)`: X = 2, false.
- `fact(3, X)`: X = 6, false.
- `fact(4, X)`: X = 24, false.
- `fact(5, X)`: X = 120, false.

At the bottom, there is a query prompt: `?- fact(5, X)`.

16. Define `sum/2` to take a list of integers as input and return the output as their sum.

`sum([], 0).`

`sum([Head|Tail], R) :-`

sum(Tail, SumTail),
R is Head + SumTail.

The screenshot displays the SWISH Prolog IDE interface. The top menu bar includes 'File', 'Edit', 'Examples', and 'Help'. The top right corner shows a search bar and a user count of '240 users online'. The main editor area on the left contains the following Prolog code:

```
1 sum([], 0).  
2 sum([Head|Tail], R) :-  
3   sum(Tail, SumTail),  
4   R is Head + SumTail.  
5
```

The right-hand pane features a large owl illustration. Below it, the execution results are shown for three queries:

- Query: `sum([5, 10, 20], X)`
Result: `X = 35`
- Query: `sum([1, 2, 3, 4, 5, 6], X)`
Result: `X = 21`
- Query: `?- sum([1, 2, 3, 4, 5, 6], X).`

At the bottom of the interface, there are tabs for 'Examples', 'History', and 'Solutions', along with a checkbox for 'table results' and a 'Run' button.

19. Write a predicate `split/4` that splits a list into two parts, the length of the first part is given.

`split(L,0,[],L).`

`split([Head|TailX],N,[Head|TailY],List2) :- N > 0, N1 is N - 1, split(TailX,N1,TailY,List2).`

The screenshot shows the SWISH Prolog IDE interface. On the left, the 'Program' window contains the following code:

```
1 split(L,0,[],L).
2 split([Head|TailX],N,[Head|TailY],List2) :- N > 0, N1 is N - 1, split(TailX,N1,TailY,List2).
```

On the right, the 'Results' window displays the execution of several test cases:

- Test 1: `split([1,2,3,4],3,X,Y)`
X = [1,2,3],
Y = [4]
false
- Test 2: `split([1,2,3,4,7,8,9,10],3,X,Y)`
X = [1,2,3],
Y = [4,7,8,9,10]
false
- Test 3: `split([1,2,3,4,7,8,9,10],6,X,Y)`
X = [1,2,3,4,7,8],
Y = [9,10]
false
- Test 4: `split([],0,X,Y)`
X = Y, Y = []
- Test 5: `split([],0,X,Y)`

The interface includes a menu bar (File, Edit, Examples, Help), a search bar, and a status bar at the bottom with 'Examples', 'History', 'Solutions', and 'table results' buttons.