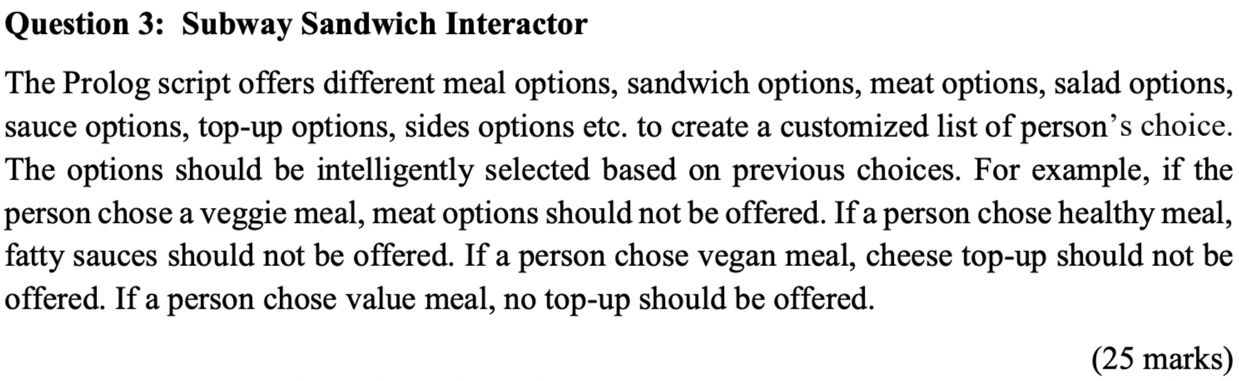
CZ3005: Artificial Intelligence
Assignment 2 – Implementing a Talking Box with Prolog

Bachhas Nikita
Matriculation Number: U1921630F
Student Email: BACH0003@e.ntu.edu.sg
Lab Group: TS5
Lab Supervisor: Mahardhika Pratama (mpratama@ntu.edu.sg)

Nanyang Technological University
Singapore, April 2021

Question 3: Subway Sandwich Interactor



Question 3: Subway Sandwich Interactor

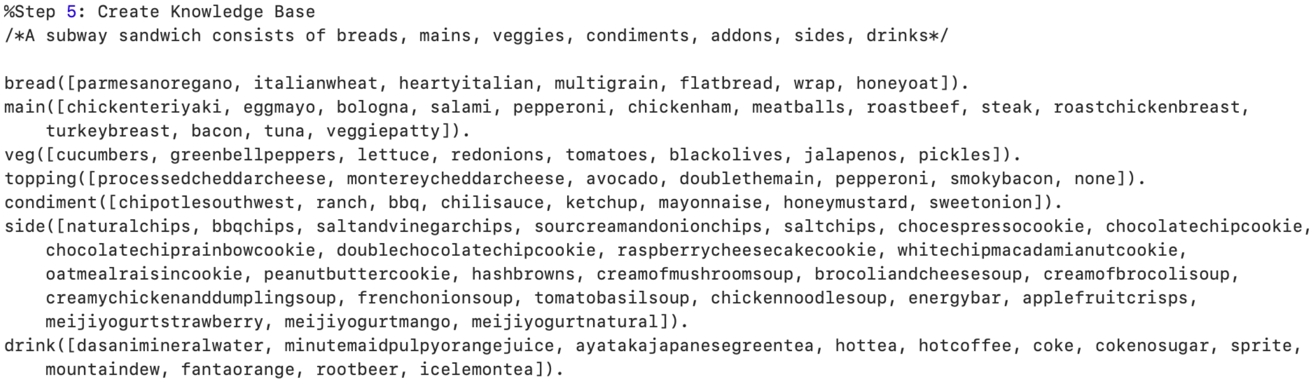
The Prolog script offers different meal options, sandwich options, meat options, salad options, sauce options, top-up options, sides options etc. to create a customized list of person's choice. The options should be intelligently selected based on previous choices. For example, if the person chose a veggie meal, meat options should not be offered. If a person chose healthy meal, fatty sauces should not be offered. If a person chose vegan meal, cheese top-up should not be offered. If a person chose value meal, no top-up should be offered.

(25 marks)

Step 1: Creating a Prolog Knowledge base of all ingredients from Subway

A typical meal from Subway would consists of

a. Bread
b. Main Ingredient such as a B.L.T or Chicken Teriyaki
c. An assortment of vegetables to add on to the sandwich
d. Extra toppings
e. Condiments such as different sauces
f. A Side
g. A drink

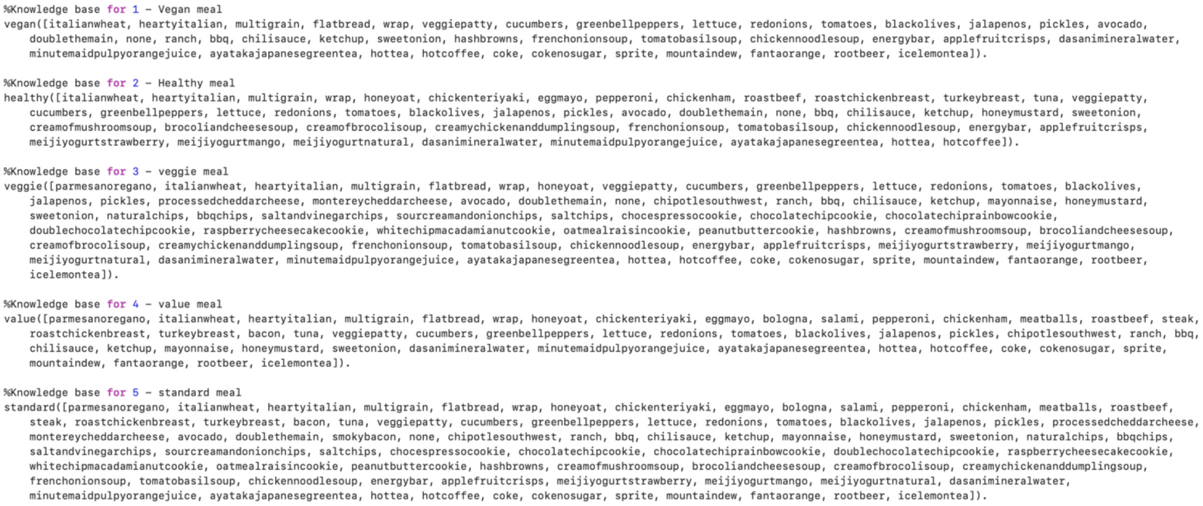


```
%Step 5: Create Knowledge Base
/*A subway sandwich consists of breads, mains, veggies, condiments, addons, sides, drinks*/

bread([parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat]).
main([chickenteriyaki, eggmayo, bologna, salami, pepperoni, chickenham, meatballs, roastbeef, steak, roastchickenbreast,
    turkeybreast, bacon, tuna, veggiepatty]).
veg([cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles]).
topping([processedcheddarcheese, montereycheddarcheese, avocado, doublethemain, pepperoni, smokybacon, none]).
condiment([chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard, sweetonion]).
side([naturalchips, bbqchips, saltandvinegarchips, sourcreamandonionchips, saltchips, chocespressocookie, chocolatechipcookie,
    chocolatechiprainbowcookie, doublechocolatechipcookie, raspberrycheesecakecookie, whitechipmacadamianutcookie,
    oatmealraisincookie, peanutbuttercookie, hashbrowns, creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup,
    creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps,
    meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural]).
drink([dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite,
    mountaindew, fantaorange, rootbeer, icelemontea]).
```

We also need to have different meal options for people:

a. Vegan – consisting of no products related to animals
b. Veggie – consisting of no meat, but still has dairy and honey products
c. Value – Doesn't allow for extra toppings or sides
d. Standard – The regular meal for users with no special preference (presents all available options)

Hence, the lists below show the ingredients allowed for each meal options:



```
%Knowledge base for 1 - Vegan meal
vegan([italianwheat, heartyitalian, multigrain, flatbread, wrap, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, avocado,
    doublethemain, none, ranch, bbq, chilisauce, ketchup, sweetonion, hashbrowns, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, dasanimineralwater,
    minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea]).

%Knowledge base for 2 - Healthy meal
healthy([italianwheat, heartyitalian, multigrain, wrap, honeyoat, chickenteriyaki, eggmayo, pepperoni, chickenham, roastbeef, roastchickenbreast, turkeybreast, tuna, veggiepatty,
    cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, avocado, doublethemain, none, bbq, chilisauce, ketchup, honeymustard, sweetonion,
    creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup, creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps,
    meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee]).

%Knowledge base for 3 - veggie meal
veggie([parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives,
    jalapenos, pickles, processedcheddarcheese, montereycheddarcheese, avocado, doublethemain, none, chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard,
    sweetonion, naturalchips, bbqchips, saltandvinegarchips, sourcreamandonionchips, saltchips, chocespressocookie, chocolatechipcookie, chocolatechiprainbowcookie,
    doublechocolatechipcookie, raspberrycheesecakecookie, whitechipmacadamianutcookie, oatmealraisincookie, peanutbuttercookie, hashbrowns, creamofmushroomsoup, brocoliandcheesesoup,
    creamofbrocolisoup, creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, meijiyogurtstrawberry, meijiyogurtmango,
    meijiyogurtnatural, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer,
    icelemontea]).

%Knowledge base for 4 - value meal
value([parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, chickenteriyaki, eggmayo, bologna, salami, pepperoni, chickenham, meatballs, roastbeef, steak,
    roastchickenbreast, turkeybreast, bacon, tuna, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, chipotlesouthwest, ranch, bbq,
    chilisauce, ketchup, mayonnaise, honeymustard, sweetonion, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite,
    mountaindew, fantaorange, rootbeer, icelemontea]).

%Knowledge base for 5 - standard meal
standard([parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, chickenteriyaki, eggmayo, bologna, salami, pepperoni, chickenham, meatballs, roastbeef,
    steak, roastchickenbreast, turkeybreast, bacon, tuna, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, processedcheddarcheese,
    montereycheddarcheese, avocado, doublethemain, smokybacon, none, chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard, sweetonion, naturalchips, bbqchips,
    saltandvinegarchips, sourcreamandonionchips, saltchips, chocespressocookie, chocolatechipcookie, chocolatechiprainbowcookie, doublechocolatechipcookie, raspberrycheesecakecookie,
    whitechipmacadamianutcookie, oatmealraisincookie, peanutbuttercookie, hashbrowns, creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup, creamychickenanddumplingsoup,
    frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural, dasanimineralwater,
    minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea]).
```

Ingredients for a Vegan Meal: italianwheat, heartyitalian, multigrain, flatbread, wrap, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, avocado, doublethemain, none, ranch, bbq, chilisauce, ketchup, sweetonion, hashbrowns, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea

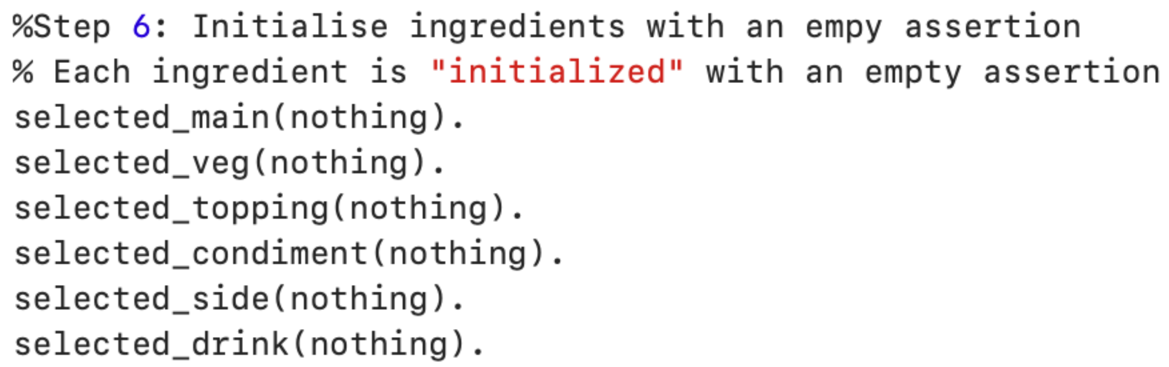
Ingredients for a Healthy Meal: italianwheat, heartyitalian, multigrain, wrap, honeyoat, chickenteriyaki, eggmayo, pepperoni, chickenham, roastbeef, roastchickenbreast, turkeybreast, tuna, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, avocado, doublethemain, none, bbq, chilisauce, ketchup, honeymustard, sweetonion, creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup, creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee

Ingredients for a Veggie Meal: parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, processedcheddarcheese, montereycheddarcheese, avocado, doublethemain, none, chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard, sweetonion, naturalchips, bbqchips, saltandvinegarchips, sourcreamandonionchips, saltchips, chocespressocookie, chocolatechipcookie, chocolatechiprainbowcookie, doublechocolatechipcookie, raspberrycheesecakecookie, whitechipmacadamianutcookie, oatmealraisincookie, peanutbuttercookie, hashbrowns, creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup, creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea

Ingredients for a Value Meal: parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, chickenteriyaki, eggmayo, bologna, salami, pepperoni, chickenham, meatballs, roastbeef, steak, roastchickenbreast, turkeybreast, bacon, tuna, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard, sweetonion, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea

Ingredients for a Standard Meal: parmesanoregano, italianwheat, heartyitalian, multigrain, flatbread, wrap, honeyoat, chickenteriyaki, eggmayo, bologna, salami, pepperoni, chickenham, meatballs, roastbeef, steak, roastchickenbreast, turkeybreast, bacon, tuna, veggiepatty, cucumbers, greenbellpeppers, lettuce, redonions, tomatoes, blackolives, jalapenos, pickles, processedcheddarcheese, montereycheddarcheese, avocado, doublethemain, smokybacon, none, chipotlesouthwest, ranch, bbq, chilisauce, ketchup, mayonnaise, honeymustard, sweetonion, naturalchips, bbqchips, saltandvinegarchips, sourcreamandonionchips, saltchips, chocespressocookie, chocolatechipcookie, chocolatechiprainbowcookie, doublechocolatechipcookie, raspberrycheesecakecookie, whitechipmacadamianutcookie, oatmealraisincookie, peanutbuttercookie, hashbrowns, creamofmushroomsoup, brocoliandcheesesoup, creamofbrocolisoup, creamychickenanddumplingsoup, frenchonionsoup, tomatobasilsoup, chickennoodlesoup, energybar, applefruitcrisps, meijiyogurtstrawberry, meijiyogurtmango, meijiyogurtnatural, dasanimineralwater, minutemaidpulpyorangejuice, ayatakajapanesegreentea, hottea, hotcoffee, coke, cokenosugar, sprite, mountaindew, fantaorange, rootbeer, icelemontea

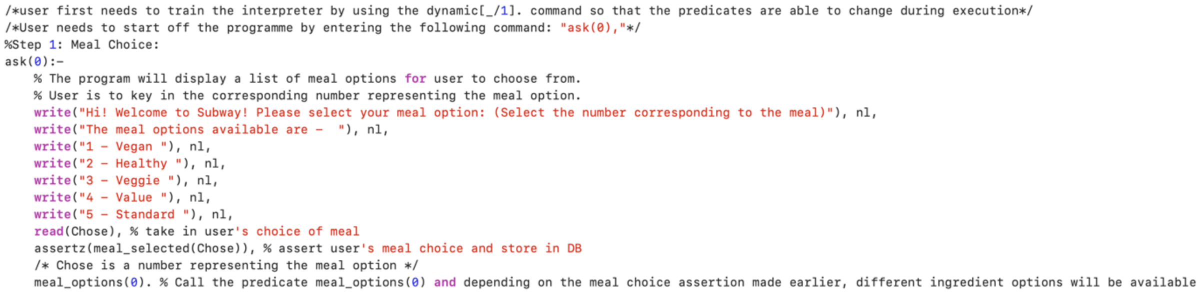
Step 2: Initialising the ingredients



```
%Step 6: Initialise ingredients with an empy assertion
% Each ingredient is "initialized" with an empty assertion
selected_main(nothing).
selected_veg(nothing).
selected_topping(nothing).
selected_condiment(nothing).
selected_side(nothing).
selected_drink(nothing).
```

The main ingredients which form the sandwich first have to be initialised with an empty assertion so that they can later store the user's choice.

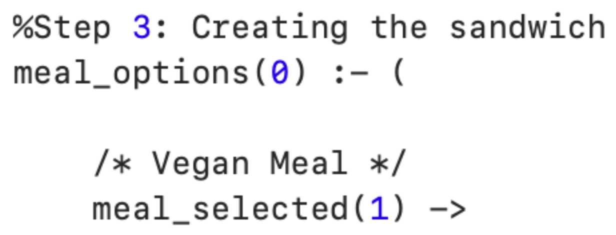
Step 3: Presenting user's with the different meal options



```
/*user first needs to train the interpreter by using the dynamic[_/1]. command so that the predicates are able to change during execution*/
/*User needs to start off the programme by entering the following command: "ask(0),"*/
%Step 1: Meal Choice:
ask(0):-
    % The program will display a list of meal options for user to choose from.
    % User is to key in the corresponding number representing the meal option.
    write("Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)"), nl,
    write("The meal options available are -  "), nl,
    write("1 - Vegan "), nl,
    write("2 - Healthy "), nl,
    write("3 - Veggie "), nl,
    write("4 - Value "), nl,
    write("5 - Standard "), nl,
    read(Chose), % take in user's choice of meal
    assertz(meal_selected(Chose)), % assert user's meal choice and store in DB
    /* Chose is a number representing the meal option */
    meal_options(0). % Call the predicate meal_options(0) and depending on the meal choice assertion made earlier, different ingredient options will be available
```

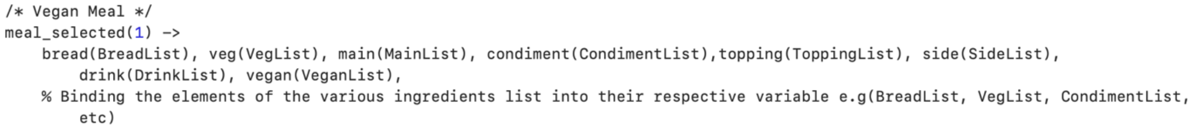
This step first introduces the user to all the meal options that they can choose from: Vegan, Healthy, Veggie, Value and Standard.
After user enters the number corresponding to the meal option they are interested in, reads the choice as "Chose" and asserts into the database as meal_selected(Chose).
The meal_selected(Chose) predicate will be needed later in the meal_options(0) predicate.
The predicate meal_options(0) is then called upon and based on the number keyed in earlier, the program moves on to that specific portion of the code.

Step 4: Creating predicate meal_options(0) to allow users to choose their ingredients according to the meal type that they choose



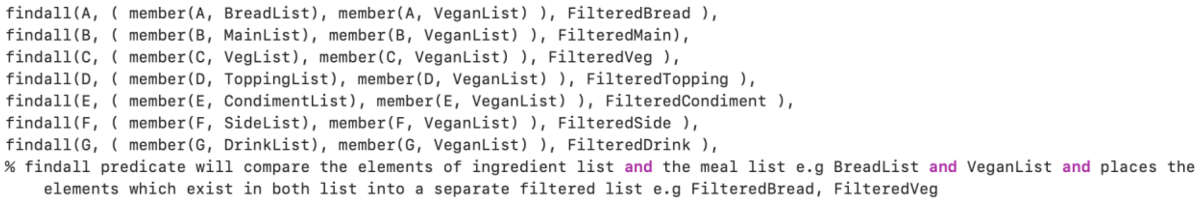
```
%Step 3: Creating the sandwich
meal_options(0) :- (

     /* Vegan Meal */
     meal_selected(1) ->
```

The predicate meal_options(0) ingredients options related to the different meal options.
For example if user selected the vegan meal – i.e. keys in the number 1 – the program will jump to meal_selected(1) within the meal_options(0) predicate.



```
/* Vegan Meal */
meal_selected(1) ->
    bread(BreadList), veg(VegList), main(MainList), condiment(CondimentList),topping(ToppingList), side(SideList),
        drink(DrinkList), vegan(VeganList),
    % Binding the elements of the various ingredients list into their respective variable e.g(BreadList, VegList, CondimentList,
        etc)
```

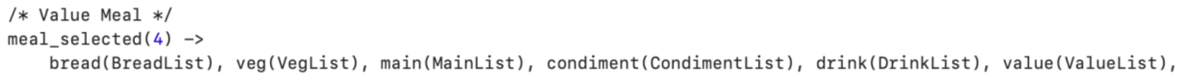
The ingredients from the knowledge base are first inserted into the different lists. For example, the ingredients from the breads knowledge base into the BreadList while the ingredients from the vegan knowledge base are insert into the VeganList.



```
findall(A, ( member(A, BreadList), member(A, VeganList) ), FilteredBread ),
findall(B, ( member(B, MainList), member(B, VeganList) ), FilteredMain),
findall(C, ( member(C, VegList), member(C, VeganList) ), FilteredVeg ),
findall(D, ( member(D, ToppingList), member(D, VeganList) ), FilteredTopping ),
findall(E, ( member(E, CondimentList), member(E, VeganList) ), FilteredCondiment ),
findall(F, ( member(F, SideList), member(F, VeganList) ), FilteredSide ),
findall(G, ( member(G, DrinkList), member(G, VeganList) ), FilteredDrink ),
% findall predicate will compare the elements of ingredient list and the meal list e.g BreadList and VeganList and places the
    elements which exist in both list into a separate filtered list e.g FilteredBread, FilteredVeg
```

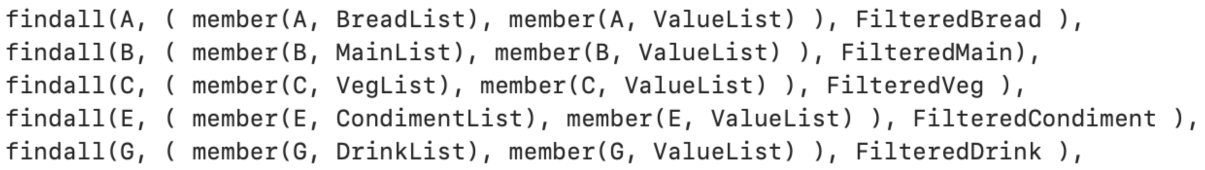
Then, the findall predicate is used to compare each element of the different lists with the allowed ingredients in the meal options list and stores them into a separate list, which is called the Fileteredingredient list. For example, when the VeganList is compared against the MainList will result in only the veggiepatty being stored in the FilteredMain list since the rest of the elements of the MainList are animal products.

Different meal options will have different set of ingredients needed for the meal. For example, the Value meal will not have any topping ingredients or side ingredients. Hence, the created lists from the knowledge base will be as such:

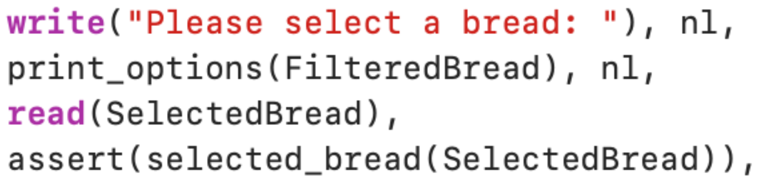


```
/* Value Meal */
meal_selected(4) ->
    bread(BreadList), veg(VegList), main(MainList), condiment(CondimentList), drink(DrinkList), value(ValueList),
```

And its findall predicate portion of the code will be as such:

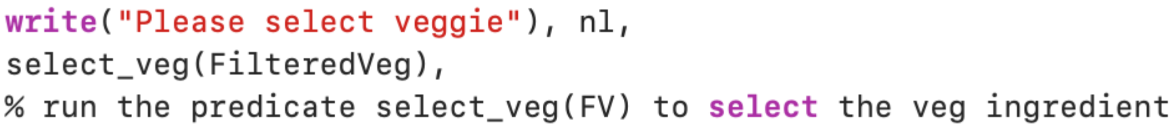


```
findall(A, ( member(A, BreadList), member(A, ValueList) ), FilteredBread ),
findall(B, ( member(B, MainList), member(B, ValueList) ), FilteredMain),
findall(C, ( member(C, VegList), member(C, ValueList) ), FilteredVeg ),
findall(E, ( member(E, CondimentList), member(E, ValueList) ), FilteredCondiment ),
findall(G, ( member(G, DrinkList), member(G, ValueList) ), FilteredDrink ),
```



```
write("Please select a bread: "), nl,
print_options(FilteredBread), nl,
read(SelectedBread),
assert(selected_bread(SelectedBread)),
```

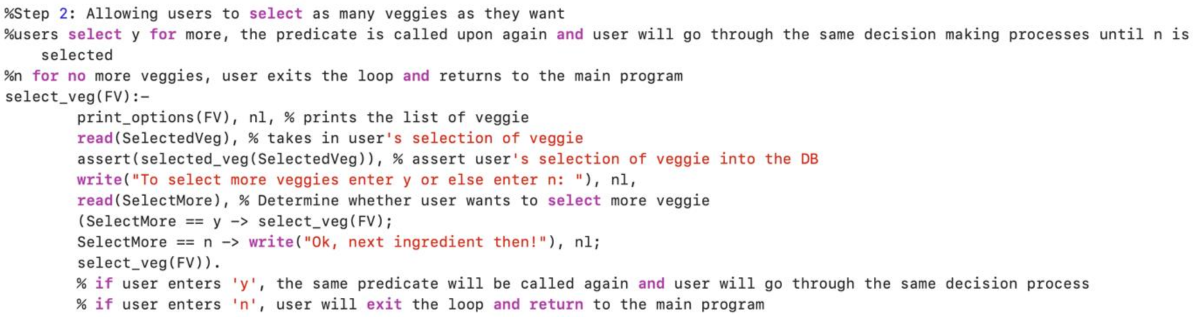
Then, based on the meal option selected, all the filtered ingredients are presented to the user so that the user can make a choice from it. User's choice is read as Fileteredingredient and asserted into the database as selected_*ingredient.* This is done for all the ingredients present in that meal option except for the veg ingredient.



```
write("Please select veggie"), nl,
select_veg(FilteredVeg),
% run the predicate select_veg(FV) to select the veg ingredient
```

Since the user can select various different types of vegetables to add to his/her sandwich, a separate predicate, select_veg(FV) is called upon using the FileteredVeg list. The predicate select_veg(FV) is described in further detail below.

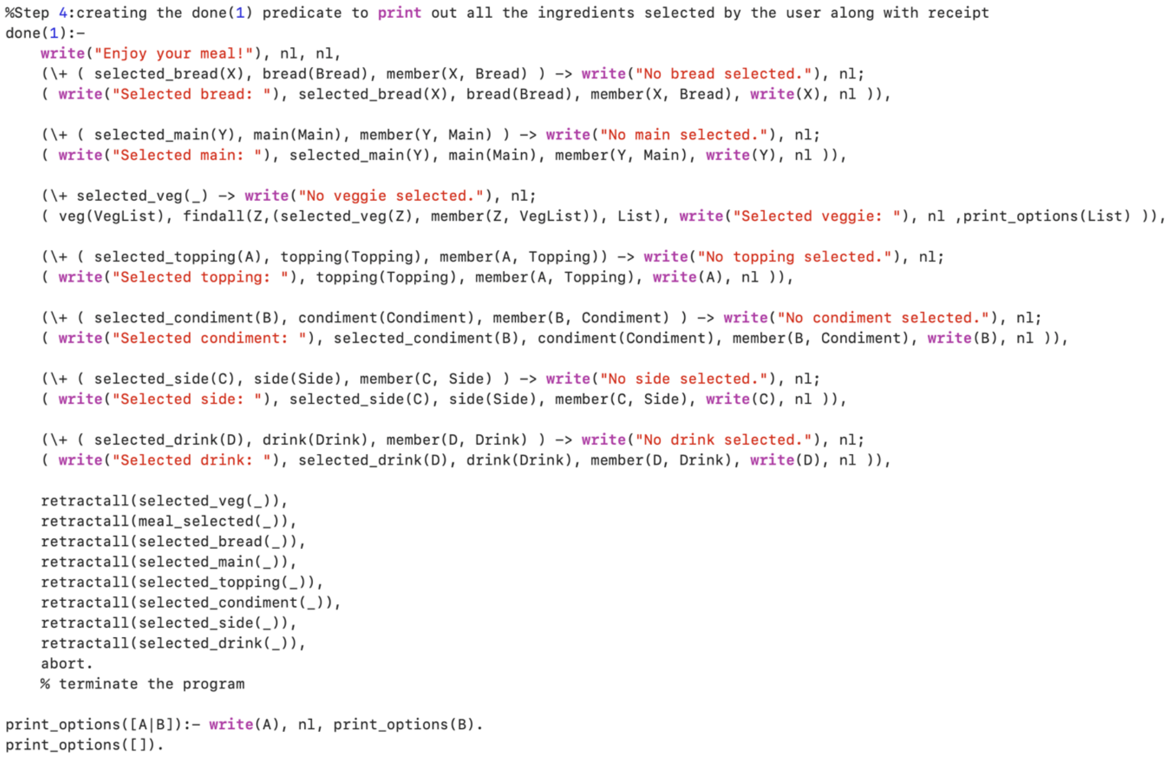
Step 5: Creating predicate select_veg(FV) to allow users to choose multiple vegetables for their sandwich



```
%Step 2: Allowing users to select as many veggies as they want
%users select y for more, the predicate is called upon again and user will go through the same decision making processes until n is
    selected
%n for no more veggies, user exits the loop and returns to the main program
select_veg(FV):-
        print_options(FV), nl, % prints the list of veggie
        read(SelectedVeg), % takes in user's selection of veggie
        assert(selected_veg(SelectedVeg)), % assert user's selection of veggie into the DB
        write("To select more veggies enter y or else enter n: "), nl,
        read(SelectMore), % Determine whether user wants to select more veggie
        (SelectMore == y -> select_veg(FV);
        SelectMore == n -> write("Ok, next ingredient then!"), nl;
        select_veg(FV)).
        % if user enters 'y', the same predicate will be called again and user will go through the same decision process
        % if user enters 'n', user will exit the loop and return to the main program
```

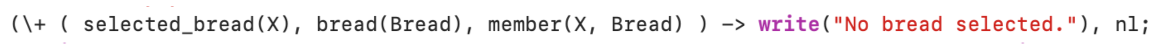
Since the user can select multiple types of vegetables for a Subway sandwich, another predicate is created here to allow users to decide their list of vegetables.
First using “print_options(FV), the program presents the users with the filtered lists of vegetables available for their meal options. This filtered list is created in the meal_options(0) predicate.
Then, the user’s choice is read as SelectedVeg and inserted into the database and selected_veg stores the choice.
The user is then prompted for more vegetables. If the user enters y, then the predicate is called upon again, this time without the vegetable that has already been selected, and if the user enters n, the user exits the loop to return to the main program in meal_options(0).
If neither y nor n is enter, then the predicate is called upon again to prompt the user for an appropriate response.

Step 3: Creating predicate done(1) to allow program to display user’s choices



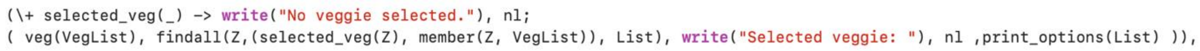
```
%Step 4:creating the done(1) predicate to print out all the ingredients selected by the user along with receipt
done(1):-
    write("Enjoy your meal!"), nl, nl,
    (\+ ( selected_bread(X), bread(Bread), member(X, Bread) ) -> write("No bread selected."), nl;
    ( write("Selected bread: "), selected_bread(X), bread(Bread), member(X, Bread), write(X), nl )),

    (\+ ( selected_main(Y), main(Main), member(Y, Main) ) -> write("No main selected."), nl;
    ( write("Selected main: "), selected_main(Y), main(Main), member(Y, Main), write(Y), nl )),

    (\+ selected_veg(_) -> write("No veggie selected."), nl;
    ( veg(VegList), findall(Z,(selected_veg(Z), member(Z, VegList)), List), write("Selected veggie: "), nl ,print_options(List) )),

    (\+ ( selected_topping(A), topping(Topping), member(A, Topping)) -> write("No topping selected."), nl;
    ( write("Selected topping: "), topping(Topping), member(A, Topping), write(A), nl )),

    (\+ ( selected_condiment(B), condiment(Condiment), member(B, Condiment) ) -> write("No condiment selected."), nl;
    ( write("Selected condiment: "), selected_condiment(B), condiment(Condiment), member(B, Condiment), write(B), nl )),

    (\+ ( selected_side(C), side(Side), member(C, Side) ) -> write("No side selected."), nl;
    ( write("Selected side: "), selected_side(C), side(Side), member(C, Side), write(C), nl )),

    (\+ ( selected_drink(D), drink(Drink), member(D, Drink) ) -> write("No drink selected."), nl;
    ( write("Selected drink: "), selected_drink(D), drink(Drink), member(D, Drink), write(D), nl )),

    retractall(selected_veg(_)),
    retractall(meal_selected(_)),
    retractall(selected_bread(_)),
    retractall(selected_main(_)),
    retractall(selected_topping(_)),
    retractall(selected_condiment(_)),
    retractall(selected_side(_)),
    retractall(selected_drink(_)),
    abort.
    % terminate the program

print_options([A|B]):- write(A), nl, print_options(B).
print_options([]).
```

When the user has not selected anything for a particular ingredient, then the program will display the message “No *ingredient* selected.” using this portion of the code:



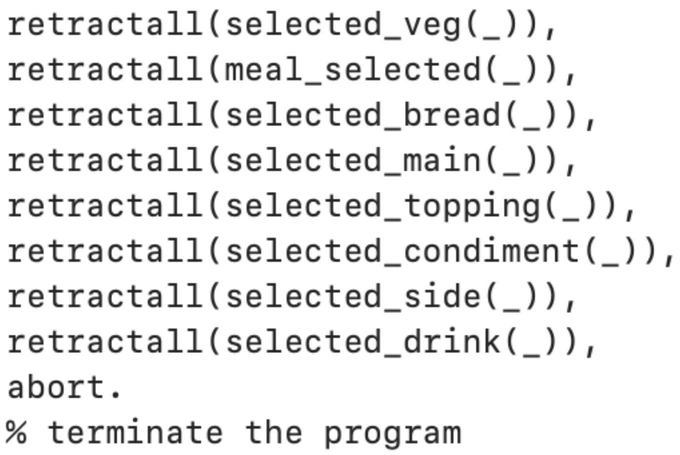
```
(\+ ( selected_bread(X), bread(Bread), member(X, Bread) ) -> write("No bread selected."), nl;
```

This is done using the not provable operator, \+, where the program check if the condition has been met or not. When the logic fails the condition, the OR potion of the predicate will run. For example, if the condition selected_bread(X), bread(Bread), member(X, Bread) – i.e is the bread selected is a member of the bread knowledge base – is true then the OR part will run, which consist of printing out the selected ingredient. This is repeated for all the ingredients.



```
(\+ selected_veg(_) -> write("No veggie selected."), nl;
( veg(VegList), findall(Z,(selected_veg(Z), member(Z, VegList)), List), write("Selected veggie: "), nl ,print_options(List) )),
```

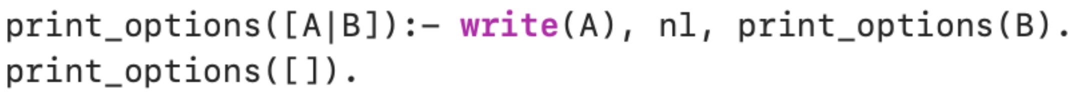
Since the user can select multiple vegetables, the findall predicate is used to add all the selected_veg assertions and the print_options(List) is used print out the vegetables selected by the users as a singular list.



```
retractall(selected_veg(_)),
retractall(meal_selected(_)),
retractall(selected_bread(_)),
retractall(selected_main(_)),
retractall(selected_topping(_)),
retractall(selected_condiment(_)),
retractall(selected_side(_)),
retractall(selected_drink(_)),
abort.
% terminate the program
```

The retractall predicate is called upon to remove all previously asserted choices in the database, so that if the user reruns the program, it will start with empty assertion and not display the old choices. If this step is not carried, past choice still remain in the database and when the done(1) predicate runs, it will print the old choices rather than the new ones.

The abort command is used to terminate the program once the user has selected his/her choices and the program has displayed the summary of the ingredients.
Should the user wish to rerun the program, he/she will have use the ask(0). command again.

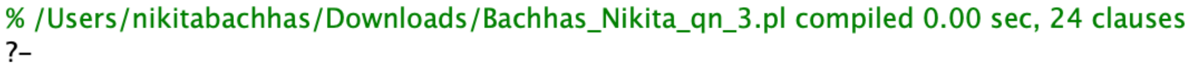


```
print_options([A|B]):- write(A), nl, print_options(B).
print_options([]).
```

The print_options([A|B]) predicate is used to print out all the elements in a list. Being a recursive predicate, it will start printing from the head element and call itself with the list from the tail element. The predicate will continue until the list is empty. This predicate is used while printing out the list of vegetables selected by the users.

Step 6: Instructions to Interact with the Prolog program

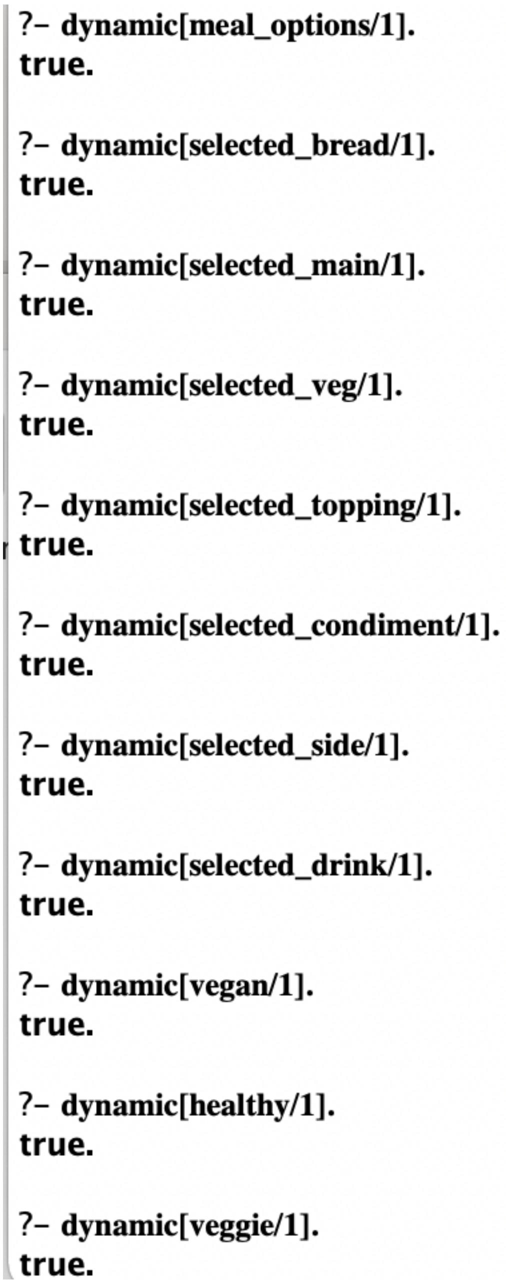
a. Consult the file Bachhas_Nikita_qn_3.pl:



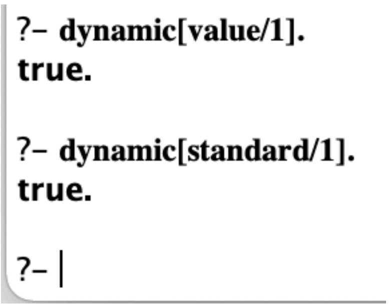
```
% /Users/nikitabachhas/Downloads/Bachhas_Nikita_qn_3.pl compiled 0.00 sec, 24 clauses
?-
```

b. Enter the following commands one-by-one first:
dynamic[meal_options/1].
dynamic[selected_bread/1].
dynamic[selected_main/1].
dynamic[selected_veg/1].

```
dynamic[selected_topping/1].
dynamic[selected_condiment/1].
dynamic[selected_side/1].
dynamic[selected_drink/1].
dynamic[vegan/1].
dynamic[healthy/1].
dynamic[veggie/1].
dynamic[value/1].
dynamic[standard/1].
```

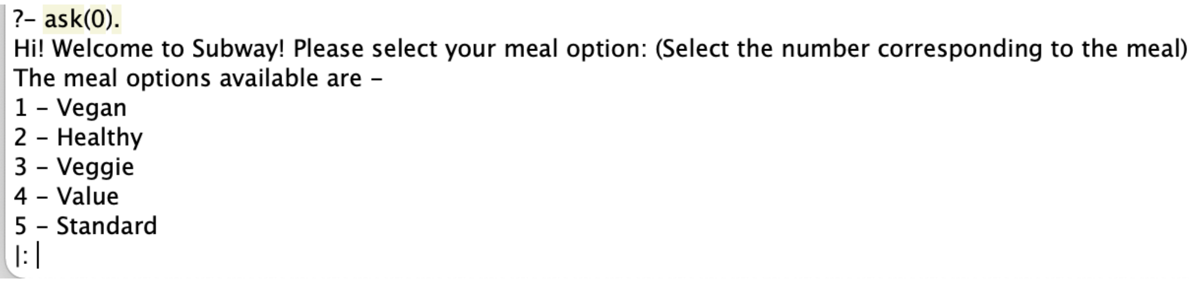


```
?- dynamic[meal_options/1].
true.

?- dynamic[selected_bread/1].
true.

?- dynamic[selected_main/1].
true.

?- dynamic[selected_veg/1].
true.

?- dynamic[selected_topping/1].
true.

?- dynamic[selected_condiment/1].
true.

?- dynamic[selected_side/1].
true.

?- dynamic[selected_drink/1].
true.

?- dynamic[vegan/1].
true.

?- dynamic[healthy/1].
true.

?- dynamic[veggie/1].
true.
```



```
?- dynamic[value/1].
true.

?- dynamic[standard/1].
true.

?- |
```

c. Initiate the program with the following command:
ask(0).

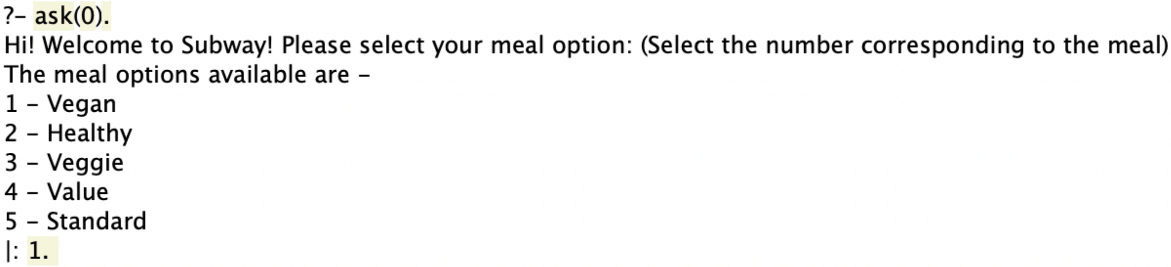


```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are -
1 - Vegan
2 - Healthy
3 - Veggie
4 - Value
5 - Standard
|: |
```

Step 7: Output of the Prolog Program

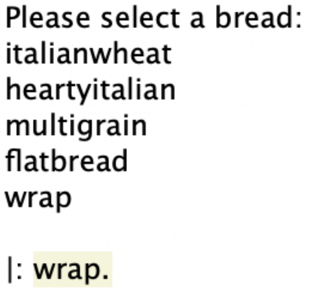
a. Vegan Meal

- All ingredients shown will be ingredients from the vegan list
- User has to enter “1.” to select the vegan meal option:



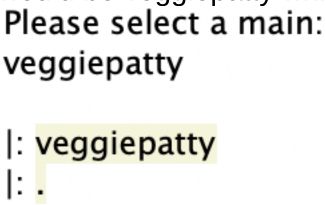
```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are -
1 - Vegan
2 - Healthy
3 - Veggie
4 - Value
5 - Standard
|: 1.
```

- Then, we select the bread:



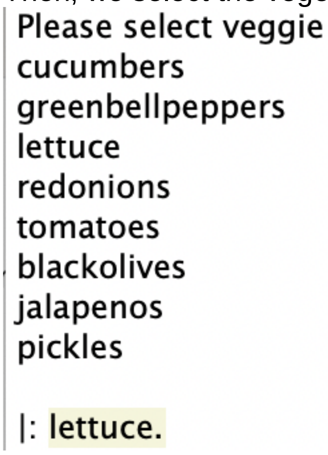
```
Please select a bread:
italianwheat
heartyitalian
multigrain
flatbread
wrap

|: wrap.
```

- Then, we select the main. Since this is a vegan meal, the only main available would be veggiepatty which is free of any animal product:



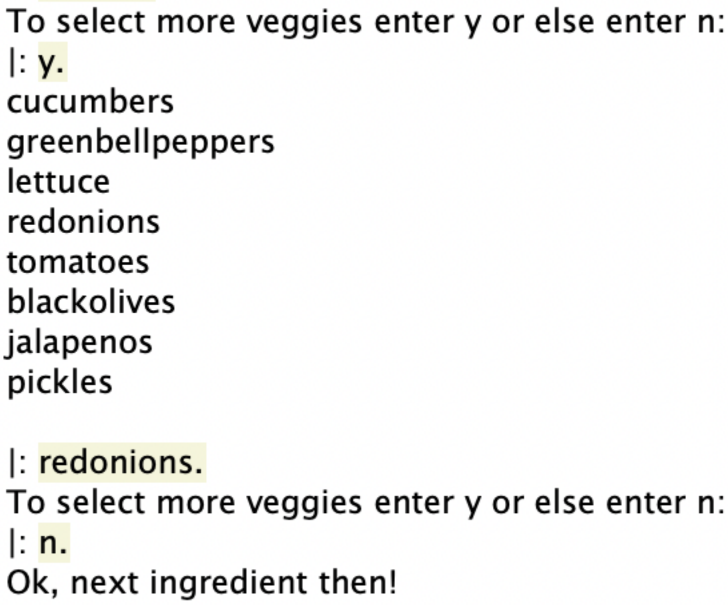
```
Please select a main:
veggiepatty

|: veggiepatty
|: .
```

- Then, we select the vegetables for the sandwich:



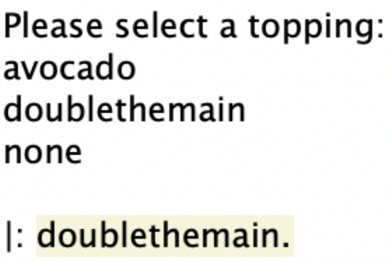
```
Please select veggie
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: lettuce.
```

- The user is then prompted if he/she wants more veggie. Here the user can enter y for yes or n for no:

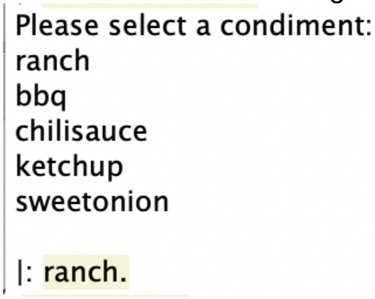


```
To select more veggies enter y or else enter n:
|: y.
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: redonions.
To select more veggies enter y or else enter n:
|: n.
Ok, next ingredient then!
```

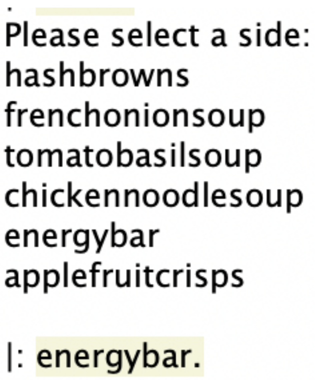
- Here, we initially selected y to add more vegetables. After selecting the addition ingredient, then the user is prompted again for more vegetables. When we enter n, the program moves back to the main program meal_options(0) but still remains under the vegan meal.
- The program will then ask the users for the rest of the ingredients:



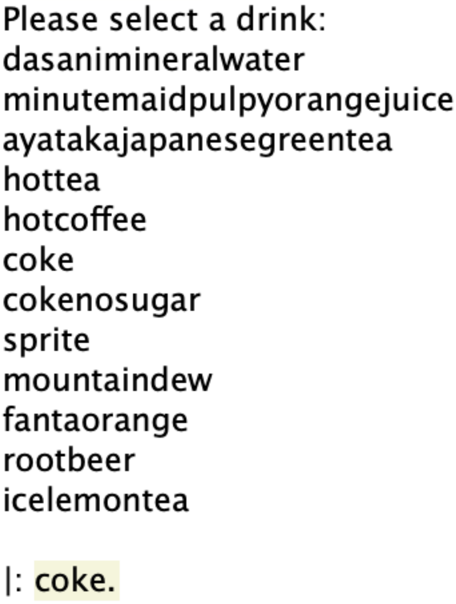
```
Please select a topping:
avocado
doublethemain
none

|: doublethemain.
```



```
Please select a condiment:
ranch
bbq
chilisauce
ketchup
sweetonion

|: ranch.
```

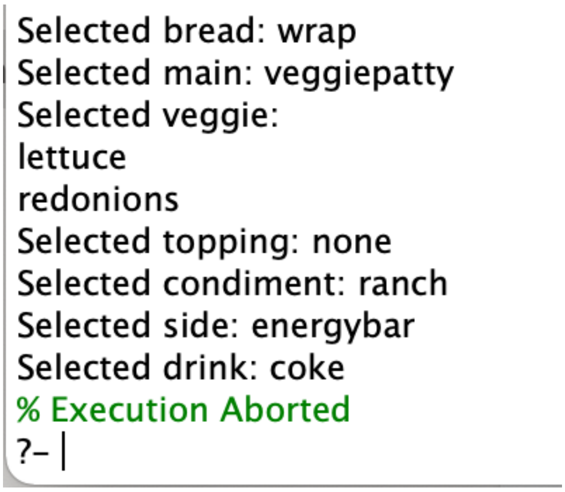


```
Please select a side:
hashbrowns
frenchonionsoup
tomatobasilsoup
chickennoodlesoup
energybar
applefruitcrisps

|: energybar.
```



```
Please select a drink:
dasanimineralwater
minutemaidpulpyorangejuice
ayatakajapanesegreentea
hottea
hotcoffee
coke
cokenosugar
sprite
mountaindew
fantaorange
rootbeer
icelemontea

|: coke.
```

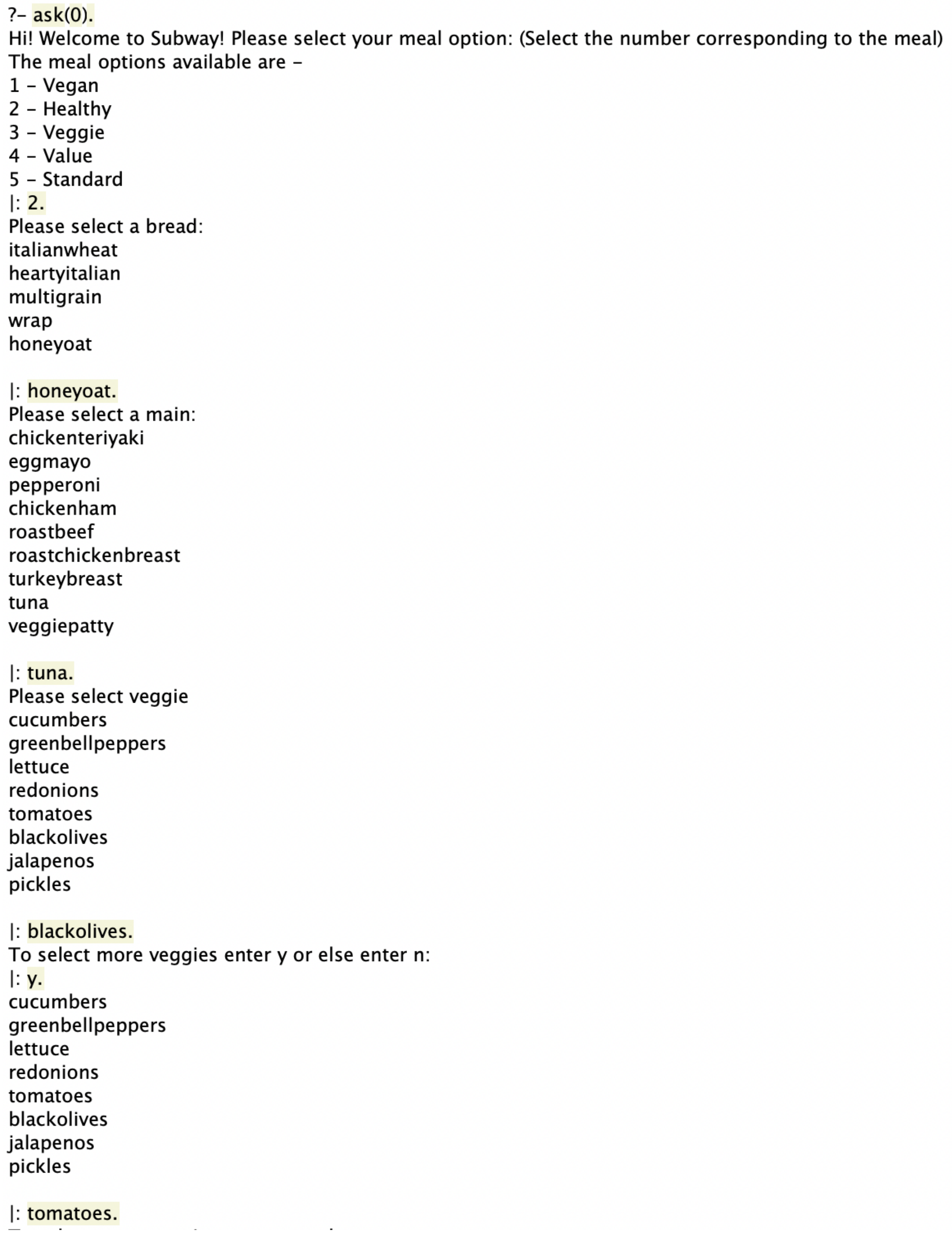
- After selecting all the ingredients, the program prints out the final summary for the sandwich:



```
Selected bread: wrap
Selected main: veggiepatty
Selected veggie:
lettuce
redonions
Selected topping: none
Selected condiment: ranch
Selected side: energybar
Selected drink: coke
% Execution Aborted
?- |
```

- Execution is aborted after the program finished running

b. Healthy Meal

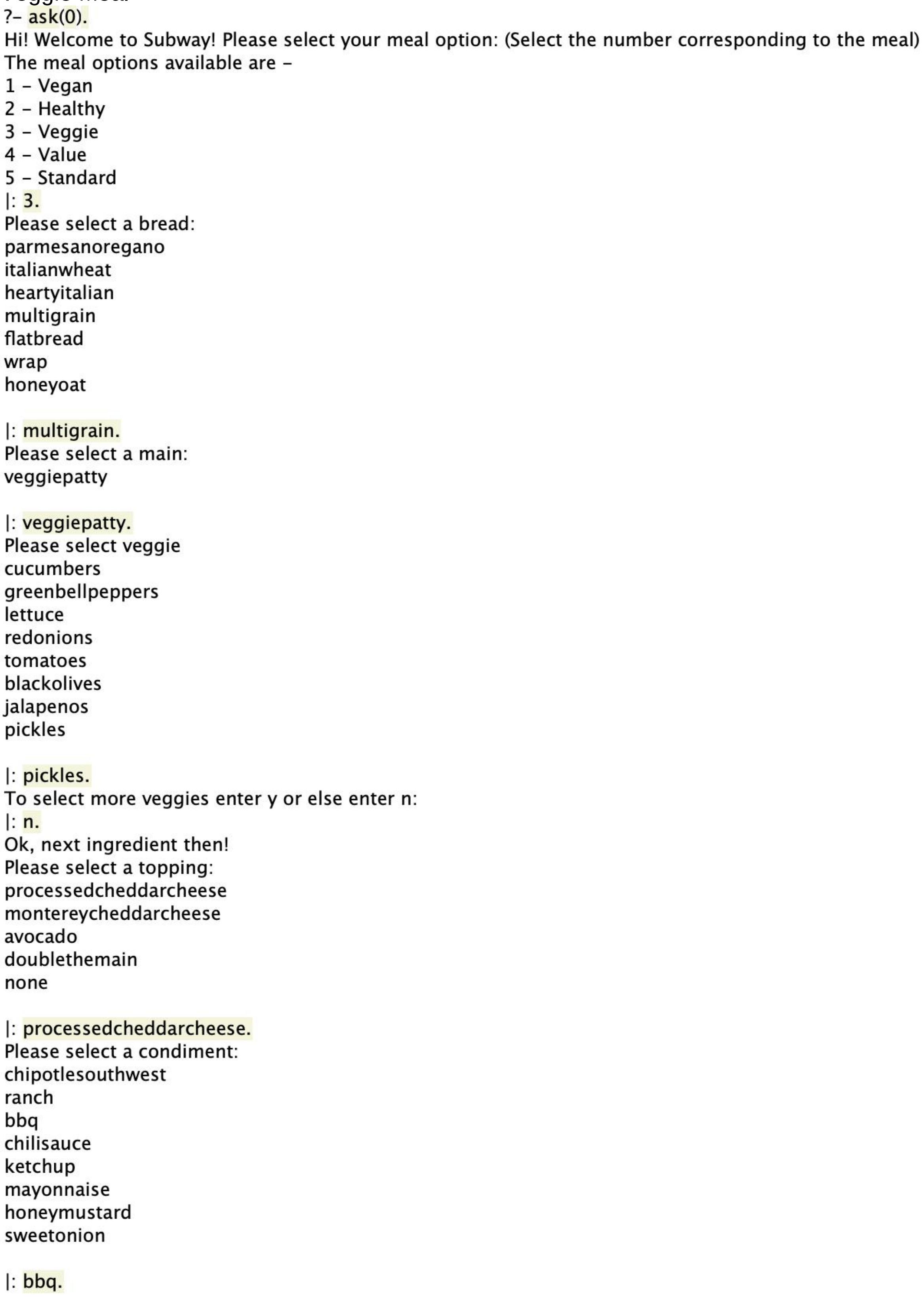


```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are –
1 – Vegan
2 – Healthy
3 – Veggie
4 – Value
5 – Standard
|: 2.
Please select a bread:
italianwheat
heartyitalian
multigrain
wrap
honeyoat

|: honeyoat.
Please select a main:
chickenteriyaki
eggmayo
pepperoni
chickenham
roastbeef
roastchickenbreast
turkeybreast
tuna
veggiepatty

|: tuna.
Please select veggie
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: blackolives.
To select more veggies enter y or else enter n:
|: y.
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: tomatoes.
```

```
To select more veggies enter y or else enter n:
|: y.
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: greenbellpeppers.
To select more veggies enter y or else enter n:
|: n.
Ok, next ingredient then!
Please select a topping:
avocado
doublethemain
none

|: avocado.
Please select a condiment:
bbq
chilisauce
ketchup
honeymustard
sweetonion

|: honeymustard.
Please select a side:
creamofmushroomsoup
brocoliandcheesesoup
creamofbrocolisoup
creamychickenanddumplingsoup
frenchonionsoup
tomatobasilsoup
chickennoodlesoup
energybar
applefruitcrisps
meijiyogurtstrawberry
meijiyogurtmango
meijiyogurtnatural

|: meijiyogurtmango.
Please select a drink:
dasanimineralwater
minutemaidpulpyorangejuice
ayatakajapanesegreentea
hottea
hotcoffee

|: hottea.
```

```
Enjoy your meal!

Selected bread: honeyoat
Selected main: tuna
Selected veggie:
blackolives
tomatoes
greenbellpeppers
Selected topping: avocado
Selected condiment: honeymustard
Selected side: meijiyogurtmango
Selected drink: hottea
% Execution Aborted
?-
```

c. Veggie Meal



```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are -
1 - Vegan
2 - Healthy
3 - Veggie
4 - Value
5 - Standard
|: 3.
Please select a bread:
parmesanoregano
italianwheat
heartyitalian
multigrain
flatbread
wrap
honeyoat

|: multigrain.
Please select a main:
veggiepatty

|: veggiepatty.
Please select veggie
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: pickles.
To select more veggies enter y or else enter n:
|: n.
Ok, next ingredient then!
Please select a topping:
processedcheddarcheese
montereycheddarcheese
avocado
doublethemain
none

|: processedcheddarcheese.
Please select a condiment:
chipotlesouthwest
ranch
bbq
chilisauce
ketchup
mayonnaise
honeymustard
sweetonion

|: bbq.
```

```
Please select a side:
naturalchips
bbqchips
saltandvinegarchips
sourcreamandonionchips
saltchips
chocespressocookie
chocolatechipcookie
chocolatechiprainbowcookie
doublechocolatechipcookie
raspberrycheesecakecookie
whitechipmacadamianutcookie
oatmealraisincookie
peanutbuttercookie
hashbrowns
creamofmushroomsoup
brocoliandcheesesoup
creamofbrocolisoup
creamychickenanddumplingsoup
frenchonionsoup
tomatobasilsoup
chickennoodlesoup
energybar
applefruitcrisps
meijiyogurtstrawberry
meijiyogurtmango
meijiyogurtnatural

|: creamofbrocolisoup.
Please select a drink:
dasanimineralwater
minutemaidpulpyorangejuice
ayatakajapanesegreentea
hottea
hotcoffee
coke
cokenosugar
sprite
mountaindew
fantaorange
rootbeer
icelemontea
|: icelemontea.
Enjoy your meal!

Selected bread: multigrain
Selected main: veggiepatty
Selected veggie:
pickles
Selected topping: processedcheddarcheese
Selected condiment: bbq
Selected side: creamofbrocolisoup
Selected drink: icelemontea
% Execution Aborted
```

d. Value Meal

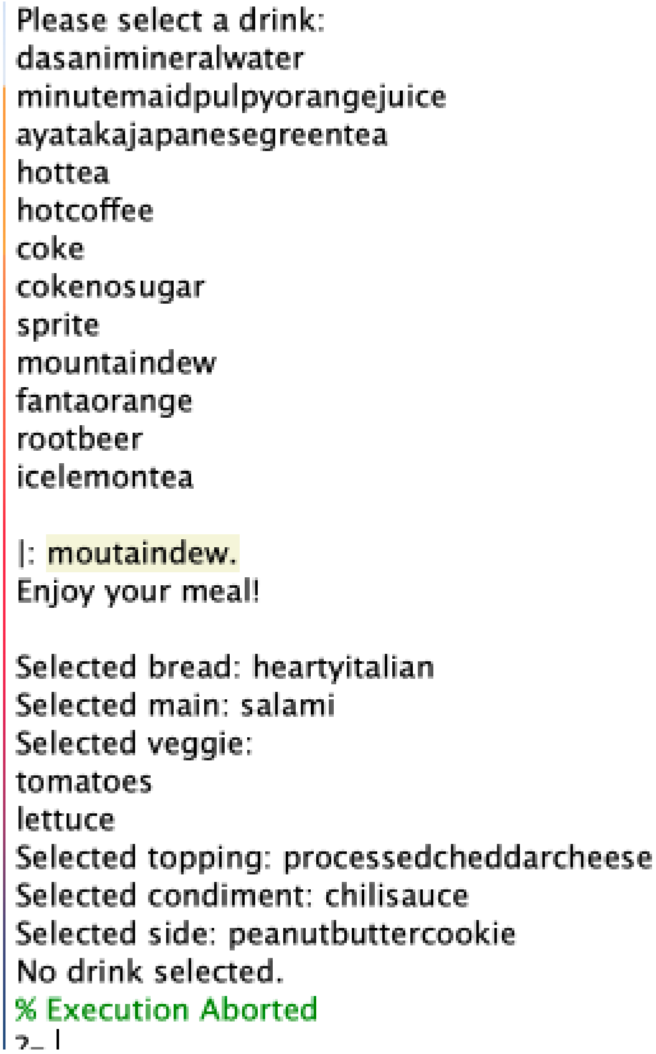
```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are -
1 - Vegan
2 - Healthy
3 - Veggie
4 - Value
5 - Standard
|: 4.
Please select a bread:
parmesanoregano
italianwheat
heartyitalian
multigrain
flatbread
wrap
honeyoat

|: italianwheat.
Please select a main:
chickenteriyaki
eggmayo
bologna
pepperoni
chickenham
meatballs
roastchickenbreast
bacon
tuna
veggiepatty

|: chickenham.
Please select veggie
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: cucumbers.
```

```
To select more veggies enter y or else enter n:
|: n.
Ok, next ingredient then!
Please select a condiment:
chipotlesouthwest
ranch
bbq
chilisauce
ketchup
mayonnaise
honeymustard
sweetonion

|: sweetonion.
Please select a drink:
dasanimineralwater
coke
sprite
icelemontea
|: sprite.
Enjoy your meal!

Selected bread: italianwheat
Selected main: chickenham
Selected veggie:
cucumbers
No topping selected.
Selected condiment: sweetonion
No side selected.
Selected drink: sprite
% Execution Aborted
```

e. Standard Meal

```
?- ask(0).
Hi! Welcome to Subway! Please select your meal option: (Select the number corresponding to the meal)
The meal options available are -
1 - Vegan
2 - Healthy
3 - Veggie
4 - Value
5 - Standard
|: 5.
Please select a bread:
parmesanoregano
italianwheat
heartyitalian
multigrain
flatbread
wrap
honeyoat

|: heartyitalian.
Please select a main:
chickenteriyaki
eggmayo
bologna
salami
pepperoni
chickenham
meatballs
roastbeef
steak
roastchickenbreast
turkeybreast
bacon
tuna
veggiepatty

|: salami.
Please select veggie
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: tomatoes.
To select more veggies enter y or else enter n:
|: y.
cucumbers
greenbellpeppers
lettuce
redonions
tomatoes
blackolives
jalapenos
pickles

|: lettuce.
```

```
To select more veggies enter y or else enter n:
|: n.
Ok, next ingredient then!
Please select a topping:
processedcheddarcheese
montereycheddarcheese
avocado
doublethemain
smokybacon
none

|: processedcheddarcheese.
Please select a condiment:
chipotlesouthwest
ranch
bbq
chilisauce
ketchup
mayonnaise
honeymustard
sweetonion

|: chilisauce.
Please select a side:
naturalchips
bbqchips
saltandvinegarchips
sourcreamandonionchips
saltchips
chocespressocookie
chocolatechipcookie
chocolatechiprainbowcookie
doublechocolatechipcookie
raspberrycheesecakecookie
whitechipmacadamianutcookie
oatmealraisincookie
peanutbuttercookie
hashbrowns
creamofmushroomsoup
brocoliandcheesesoup
creamofbrocolisoup
creamychickenanddumplingsoup
frenchonionsoup
tomatobasilsoup
chickennoodlesoup
energybar
applefruitcrisps
meijiyogurtstrawberry
meijiyogurtmango
meijiyogurtnatural

|: peanutbuttercookie.
```



```
Please select a drink:
dasanimineralwater
minutemaidpulpyorangejuice
ayatakajapanesegreentea
hottea
hotcoffee
coke
cokenosugar
sprite
mountaindew
fantaorange
rootbeer
icelemontea

|: moutaindew.
Enjoy your meal!

Selected bread: heartyitalian
Selected main: salami
Selected veggie:
tomatoes
lettuce
Selected topping: processedcheddarcheese
Selected condiment: chilisauce
Selected side: peanutbuttercookie
No drink selected.
% Execution Aborted
?-
```

- As you can see here, since the name of the ingredient wasn’t spelled out correctly, the drink wasn’t selected.
- Also, you can see that after the program is executed each time, the retractall predicate ensures that none of the old ingredients are displayed when the user reruns the program.