# The Pizza-House

### Problem Description

The Pizza-House chalkboard that lists various types of pizza with their ingredients and prices was wiped out by the rain. You are asked to re-write the board using the following information:

There are five types of pizza with completely different toppings and prices. Each type of pizza has two toppings: one from the list ham, mussels, prawns, salami and tuna; the other from the list avocado, corn, olives, pineapple and tomato. The five prices are $5, $6, $7, $8 and $9.

Other clues are: Hawaiian pizza has mussels and costs over $6. Marco-Polo pizza has tomato, but no ham, which is on the pizza that costs $8. Pepperoni pizza costs $7, and Super has no pineapple. The price of the pizza that has tuna and corn is not $6, and the pizza that costs $5 has olives but no salami. The price of the pizza that has pineapple is not $9. The last type of pizza is called Ninja-Pizza.

You need to determine which pizzas have which toppings (there is exactly one correct solution). You will define the predicate as **chalkboard/1**. See the section below for an example of how the predicate should be defined and called.

### Hints/Suggestions

Try representing a pizza as a structure of the form [Name, Topping1, Topping2, Price], and the chalkboard as a list of five such structures.

You then need to select two toppings and a price for each pizza, keeping in mind the constraints given above. To do this, you might find the **permutation/2** predicate from the Prolog lists library useful in preparing the pizzas.

One approach would be to make five pizzas and then test the constraints; however, you will find that the performance of your program is significantly improved if you check the constraints as you go along, rather than at the end.

Here is an example of what the left hand side of the chalkboard predicate looks like in my solution:

chalkboard([[HPN, HP1, HP2, HPP],

[MPN, MP1, MP2, MPP],

[NPN, NP1, NP2, NPP],

[PPN, PP1, PP2, PPP],

[SPN, SP1, SP2, SPP]]):-

?-chalkboard(X).

X = [[hawaiian, …,…,…],[marcopolo,…,…,…],[ninja,…,…,…],[pepperoni,…,…,…],[super,…,…,…]]

Note: Your program will provide the contents of the “…”s .