Exercice 1 :

Exercice 2 : Prolog

Partie 1 Définissez un prédicat *neighbor(X, Y)*

```
neighbor(X, Y) :-
   e(X, Y);
   e(Y, X).
```

Partie 2 Définissez le prédicat *reachable(A, B)*

```
reachable(X, X).
reachable(X, Y) :-
   e(X, Z),
   reachable(Z, Y).
```

Partie 3 Définissez un deriner prédicat *path(X, Y, Route)*

```
path(X, X, [X]).
path(X, Y, [X|Rest]) :-
   e(X, Z),
   path(Z, Y, Rest).
```

Exercice 3 : Refactoriser des boucles While

Vieux Code	Nouveau Code
`fun1(xs)`	`xs.reverse`
`fun2(xs)(x => 2 * x)`	`xs.map(2 * _)`
`fun3(xs)(_ + 7)`	`xs.map(_+ 7).foldLeft(0)(_+_)` ou `xs.map(_+ 7).sum`
`fun4(xs,ys)`	`xs.zip(ys).reverse`
`fun5(xs)(_ % 2 == 1))`	`xs.partition(_ % 2 == 1)`
`fun6(xs, 3)`	`xs.take(3).reverse`
`fun2(xs)(ys => fun1(ys))`	`xs.map(_.reverse)`
`fun5(xs)(p)._1`	`xs.filter(p)`