Solution Exercice 2 - Série 2

Forme générale de l'instruction case :

```
case <exp> RA of
  e11,RB e12, ..., e1n1 :RC <inst1>;RD
  e21, e22, ..., e2n2 : <inst2>;
  .
  .
  .
  ek1, ek2, ..., eknk : <instk>
otherwise RD <instk+1>
end RE
```

Forme intermédiaire choisie:

	{ quadruplets (exp) } (:= , q_c-1 , T) La valeur de <exp> rangé dans la variable T	R_A
	(BE , $inst_1$, T , e_{11})	R_B
	(BE , $inst_1$, T , e_{12})	R_B
	. . .	
	(BNE , $ligne_2$, T , e_{1n1})	R_C
$inst_1$	{ quadruplets (<$inst_1$>) } (BR , fin, ,)	R_D
$ligne_2$	(BE , $inst_2$, T , e_{21})	R_B
	. . .	
$ligne_k$	(BE , $inst_k$, T , e_{k1})	R_B
	. . .	
	(BNE , otherwise, T , e_{knk})	R_C
$inst_k$	{ quadruplets (<$inst_k$>) } (BR , fin, ,)	R_D
otherwise	{ quadruplets (<$inst_{k+1}$>) }	R_E
fin		

Initialement :
Pile_BE := pile vide
Pile_BR := pile vide

Routine A

```
Début
    quad [ qc ] := ( := , qc - 1 , , T )
    qc ++
    type_exp := type(<exp>)
fin
```

Routine B (e)

```
Début
    Si type_exp et type(e) sont compatibles
    Alors
      quad [ qc ] := ( BE , , T, e )
      qc ++
      empiler(Pile_BE, qc)
fin
```

Routine C (e)

```
Début
    Si type_exp et type(e) sont compatibles
    Alors
      quad [ qc ] := ( BNE , , T, e )
      sauve_BNE := qc
      qc ++
      Tant que non pile vide (Pile_BE) faire
          a:= dépiler(Pile_BE)
          quad [ a , 2 ] := qc
      Fait
   FSi
fin
```

Routine D

```
Début
  quad [ qc ] := ( BR , , , )
  empiler(Pile_BR, qc)
  qc ++
  quad [ sauve_BNE , 2 ] := qc
fin
```

Routine E

```
Début
  Tant que non pile vide (Pile_BR) faire
     a:= dépiler(Pile_BR)
     quad [ a , 2 ] := qc
  Fait
  quad [ sauve_BNE , 2 ] := qc
fin
```
