1. Transformation parenthésée en postfixée.

```
Parenthesee_vers_postfixee (expr_par, n_par) {
   P = Pile_vide (); //P est une pile de caracteres
   n_post <- 1;
   Pour i de 1 A n_par Faire {
      c <- expr_par[i];
      Selon que {
         c dans '0'..'9' :
            expr_post[n_post] <- c;
            n_post <- n_post + 1;

         c = '(' :
            Empiler(P, c);

         c = ')':
            Tant que (Valeur_sommet(P) <> '(') Faire {
               expr_post[n_post] <- Valeur_sommet (P);
               n_post <- n_post + 1;
               Depiler (P);
            }
            Depiler (P);

         c dans ('+', '-', '*', '/') :
            Tant que (Non Pile_vide? (P) ET
                      Valeur_sommet (P) <> '(' ET
                      Priorite (Valeur_sommet (P)) >= Priorite (c))) {
               expr_post[n_post] <- Valeur_sommet (P);
               n_post <- n_post + 1;
               Depiler (P);
            }
            Empiler (P, c);
       }
   }
   Tant que (Non Pile_vide? (P)) Faire {
      expr_post[n_post] <- Valeur_sommet (P);
      n_post <- n_post + 1;
      Depiler (P);
   }

   retourner (expr_post, n_post);
}
```

2. Inversion d'une file.

```
Inverser_file (F) {
   P = Pile_vide ();
   Tant que Non File_vide? (F) Faire {
      Empiler (P, Valeur_premier(F));
      Defiler (F);
    }
   Tant que Non Pile_vide? (P) Faire {
      Enfiler (F, Valeur_sommet (P));
      Depiler (P);
    }
}
```