1. (a) multiplication par additions successives.

```
Multiplier (a, b) {
  produit <- 0;
  Pour i <- 1 A a Faire
    produit <- produit + b;
  retourner produit;
}
```

```
Multiplier (a, b) {
  produit <- 0;
  Repeter
    produit <- produit + b;
    a <- a - 1;
  Jusqu'a a = 0
  retourner produit;
}
```

Problème posé par la version utilisant la boucle *Repeter* : cas a = 0.

(b) division par soustractions successives.

```
Diviser (a, b) {
  quotient <- 0;
  Tant que a >= b Faire {
     a <- a - b;
     quotient <- quotient + 1;
   }
   retourner quotient;
}
```

(c) puissance par multiplications successives.

```
Puissance (a, b) {
  puiss <- 1;
  Pour i <- 1 A b Faire
     puiss <- puiss * a;
  retourner puiss;
}
```

(d) pgcd par soustractions successives.

```
PGCD (a, b) {
  Tant que a <> b Faire
    Si a > b Alors
       a <- a - b;
    Sinon
       b <- b - a;
  retourner a;
}
```

2. $\sum_{i=1}^{i=n} i$ et $\sum_{i=1}^{i=n} x^i$.

```
Somme_1_n (n) {
  somme <- 0;
  Pour i <- 1 A n Faire
    somme <- somme + i;
  retourner somme;
}
```

```
Somme_puiss (x, n) {
  somme <- 0;
  puiss_x <- 1;
  Pour i <- 1 A n Faire {
    puiss_x <- puiss_x * x;
    somme <- somme + puiss_x;
  }
  retourner somme;
}
```

1. Moyenne des éléments d'un tableau.

```
Moyenne (T, N) {
  somme <- 0;
  Pour i <- 1 a N Faire
    somme <- somme + T[i];
  moyenne <- somme / N;
  retourner moyenne;
}
```

   Minimum des éléments d'un tableau et sa position.

```
Minimum (T, N) {
  min <- T[1];
  Pour i <- 2 a N Faire
    Si T[i] < min  Alors {
      min <- T[i];
      pos_min <- i;
    }
  retourner (min, pos_min);
}
```

2. Calcul du nombre d'occurences d'un élément donné.

```
Nb_occurences (T, N, X) {
  nb_occ <- 0;
  Pour i <- 1 a N Faire
   Si T[i] = X Alors
      nb_occ <- nb_occ + 1;
   retourner (nb_occ);
}
```

3. Teste si un tableau est trié.

```
Est_trie (T, N) {
  i <- 1;
  Tant que i < N ET T[i] <= T[i+1] Faire
     i <- i + 1;
  est_trie <- (i = N);
  retourner est_trie;
}
```

4. Teste si deux tableaux sont identiques.

```
Sont_identiques (T1, T2, N) {
  i <- 1;
  Tant que (i <= N) ET (T1[i] = T2[i]) Faire
   i <- i + 1;
```

```
    sont_identiques <- (i = N + 1);
    retourner sont_identiques;
  }
```

5. Produit scalaire de deux vecteurs u et v de dimension n.

```
Produit_scalaire (u, v, n) {
  prod_scalaire <- 0;
  Pour i <- 1 a n Faire
    prod_scalaire <- prod_scalaire + u[i] * v[i];
  retourner prod_scalaire;
}
```

6. Produit de deux matrices carrées $A = (a_{ij})$ et $B = (b_{ij})$ de dimension n.

```
Produit_matriciel (a, b, n) {
  Pour i <- 1 a n Faire
    Pour j de 1 a n Faire {
      c[i][j] <- 0;
      Pour k de 1 a n Faire
         c[i][j] <- c[i][j] + a[i][k] * b[k][j];
    }
  retourner c;
}
```

7. Décalage vers la gauche des élements d'un tableau.

```
Decalage_gauche (T, N) {
  tmp <- T[1];
  Pour i <- 1 a N-1 Faire
    T[i] <- T[i+1];
  T[N] <- tmp;
}
```