

```

Algorithmme dialogue_factorielle
(* calcule la factorielle des nombres entrés *)
(* jusqu'à ce qu'on entre un nb négatif      *)

var n : entier (* on calcule n! *)
    i : entier (* compteur de boucle *)
    f : entier (* pour le calcul *)
début
    écrire("factorielle de :")
    lire(n)
    tant que n ≥ 0 faire
        f ← 1
        pour i de 2 à n faire
            f ← f × i
        fin pour
        écrire("résultat ", f)
        écrire("factorielle de (<0 pour terminer) :")
        lire(n)
    fin tant que
    écrire("au revoir")
fin

```

FIG. 1 – Version initiale

```

Algorithmme dialogue_factorielle
(* calcule la factorielle des nombres entrés *)
(* jusqu'à ce qu'on entre un nb négatif      *)

var n : entier (* on calcule n! *)
    f : entier (* résultat *)

procédure factorielle(n:entier; var fn:entier)
(* place n! dans fn *)
var i:entier
début
    fn ← 1
    pour i de 2 à n faire
        fn ← fn × i
    fin pour
fin

début
    écrire("factorielle de :")
    lire(n)
    tant que n ≥ 0 faire
        factorielle(n, f)
        écrire("résultat ", f)
        écrire("factorielle de (<0 pour terminer) :")
        lire(n)
    fin tant que
    écrire("au revoir")
fin

```

FIG. 2 – Version avec procédure

```

Algorithme dialogue_factorielle
(* calcule la factorielle des nombres entrés *)
(* jusqu'à ce qu'on entre un nb négatif      *)

var n : entier (* on calcule n! *)
    f : entier (* résultat *)

fonction factorielle(n:entier):entier
(* retourne valeur de n! *)
var i:entier
    f:entier
début
    f ← 1
    pour i de 2 à n faire
        f ← f × i
    fin pour
    retour(f)
fin

début
    écrire("factorielle de :")
    lire(n)
    tant que n ≥ 0 faire
        f ← factorielle(n)
        écrire("résultat ", f)
        écrire("factorielle de (<0 pour terminer) :")
        lire(n)
    fin tant que
    écrire("au revoir")
fin

```

FIG. 3 – Version avec fonction