

1 - ALGORITHME CalculatriceExpression DEBUT AFFICHER "=====
CALCULATRICE =====" AFFICHER "Exemple: tapez '2 + 3' puis Entrée"

```
TANT QUE VRAI FAIRE
    // PHASE 1 : SAISIE ET PRÉ-TRAITEMENT
    SAISIR expression ← input(">>> ")
    expression ← SUPPRIMER_ESPACES(expression)

    // Vérification commandes de sortie
    SI expression EN ('quit', 'exit', 'q') ALORS
        QUITTER BOUCLE
    FIN SI

    // Vérification expression vide
    SI expression EST VIDE ALORS
        CONTINUER
    FIN SI

    // PHASE 2: ÉVALUATION DE L'EXPRESSION
    resultat ← evaluer_expression(expression)

    // PHASE 3 : AFFICHAGE
    AFFICHER "= {resultat}"
FIN TANT QUE
```

FIN

// SOUS-ALGORITHME evaluer_expression ALGORITHME

evaluer_expression(expression) DEBUT // Étape 1 : Séparation des éléments
elements ← DIVISER(expression, " ")

```
// Étape 2 : Validation du format
SI LONGUEUR(elements) ≠ 3 ALORS
    RETOURNER "Erreur: Format attendu 'nombre opérateur nombre'"
FIN SI

nombre1_str, operateur, nombre2_str ← elements

// Étape 3 : Conversion numérique
ESSAYER
    nombre1 ← CONVERTIR(nombre1_str, float)
    nombre2 ← CONVERTIR(nombre2_str, float)
SI ERREUR ValueError ALORS
    RETOURNER "Erreur: Les deux premiers éléments doivent être des nombres"
FIN ESSAYER

// Étape 4 : Exécution du calcul
SELON operateur
```

```
CAS "+": RETOURNER nombre1 + nombre2
CAS "-": RETOURNER nombre1 - nombre2
CAS "*": RETOURNER nombre1 * nombre2
CAS "/":
    SI nombre2 = 0 ALORS
        RETOURNER "Erreur: Division par zéro"
    SINON
        RETOURNER nombre1 / nombre2
    FIN SI
AUTRE:
    RETOURNER "Erreur: Opérateur '" + operateur + "' non supporté"
FIN SELON
```

FIN