# Validation :

## 1.Recherche caractère :

### Pseudocode :

**DÉBUT**

//mot [20] (chaine de caractère) : mot entré par l’utilisateur

//lettre (caractère) : lettre recherchée dans le mot

//lettre\_tampon (caractère)

// i (entier) : variable d’implémentation

Demander à l’utilisateur le mot

Lire mot [20] sur le clavier

Demander à l’utilisateur la lettre

Lire lettre sur le clavier

i :=0

Faire

lettre\_tampon :=mot[i]

implémenter i

Tant que lettre\_tampon ≠ nulle && lettre ≠ lettre

Si lettre\_tampon== nulle

Afficher message « -1 »

Sinon

Afficher position lettre i

**FIN**

Diagramme :

## 2.Calcul d’un sinus à l’aide d’une série:

### Pseudocode :

PI : = 3,141926535

**DÉBUT**

//x (réel) : Valeur à appliquer le sinus

Demander valeur de x

Lire x du clavier

Afficher sommation(x)

Return 0;

**FIN**

**FONCTION** somation (x) : résultat

// x (réel) : valeur à calculer le sinus

// résultat (réel) : réponse du sinus

**DÉBUT**

// j (entier) : Valeur d’implémentation

j := 0

résultat :=0

pour i := 1 à i==20 implémenter i :=i+2

résultat : = résultat + ( exposant (x, i)/factoriel(i))\*exposant(-1, j);

j ++;

Retourner résultat

**FIN**

**FONCTION** exposant (x, i) : résultat

// x (réel) : valeur à multiplier

// i (entier) : exposant

// résultat (réel) : résultat de l’exposant

**DÉBUT**

// j (entier)

Pour j : = 0 à j ==i implémenter j++

Si j ==0

résultat :=1

sinon

résultat=résultat\*x

Retourner résultat

**FIN**

**FONCTION** factoriel(i) : résultat

// i (entier) : valeur à factoriser

//résultat (réel) : résultat de la factorisation

**DÉBUT**

// n (entier) : valeur d’implémentation

résultat :=1

pour n :=1 à n==i implémenter n++

résultat :=résultat\*n

Retourner résultat

**FIN**

Diagramme :

## 3.Calculde l’addition de matrices:

### Pseudocode :

**DÉBUT**

//dimensionx(entier) : nombre de ranger des matrices

//dimensiony(entier) : nombre de colonne des matrices

//range(entier) : numéro de ranger

//colonne(entier) : numéro de colonne

Demander valeur dimensionx

Lire dimensionx sur clavier

Demander valeur dimensiony

Lire dimensiony sur clavier

// m1[dimensionx] [dimensiony](entier) : première matrice

// m2[dimensionx] [dimensiony](entier) : deuxième matrice

// m3[dimensionx] [dimensiony](entier) : matrice sommative

Pour range :=0 à range==dimensionx implémenter range ++

Pour colonne :=0 ; colonne == dimensiony implémenter colonne ++

Demander valeur m1[range ][colonne]

Lire m1[range ][colonne] sur clavier

Pour range :=0 à range==dimensionx implémenter range ++

Pour colonne :=0 ; colonne == dimensiony implémenter colonne ++

Demander valeur m2[range ][colonne]

Lire m2[range ][colonne] sur clavier

Pour range :=0 à range==dimensionx implémenter range ++

Pour colonne :=0 ; colonne == dimensiony implémenter colonne ++

m3[range][colonne] :=m1[range][colonne]\*m2[range][colonne]

afficherM(dimensionx,dimensiony,m1[dimensionx][ dimensiony])

afficherM(dimensionx,dimensiony,m2[dimensionx][ dimensiony]))

afficherM(dimensionx,dimensiony,m3[dimensionx][ dimensiony]))

retourner 0;

**FIN**

**FONCTION** afficherM (dimensionx,dimensiony,matrice[dimensionx][ dimensiony])

// dimensionx (réel) : nombre de ranger de la matrice

// dimensiony (réel) : nombre de colonne de la matrice

// matrice[dimensionx][ dimensiony](réel) : matrice à afficher

**DÉBUT**

//range(entier) : numéro de ranger

//colonne(entier) : numéro de colonne

Pour range :=0 à range==dimensionx implémenter range ++

Pour colonne :=0 ; colonne == dimensiony implémenter colonne ++

Afficher m3[ranger][ colonne]

**FIN**

Diagramme :