TD1 – éléments de corrigé (algorithmes)

Algorithme 1 : AfficherPrixTTC
Principe : Demander le prixHT à l'utilisateur, multiplier cette valeur par la constante de TVA (20%) et afficher le résultat dans la console.
Entrée : - TAUX_TVA : réel constant = 0.20
Local : - prixHT : réel
- prixTTC : réel
Sortie : void
Début
    Soit prixHT = SaisirNombreRéel()
    Soit prixTTC = prixHT * (1 + TAUX_TVA)
    Afficher(prixTTC)
Fin

Algorithme 2 : AfficherPrixTTCAvecRemise
Principe : Demander le prixHT à l'utilisateur, calculer le prixTTC (algo 1) compter le nombre de tranches de 100 euros dans prixHT (division entière) et retirer 8 euros fois ce nombre au prixTTC.
Entrée : - TAUX_TVA : réel constant = 0.20
- VALEUR_REMISE : entier constant = 8
- TRANCHE_REMISE : entier constant = 100
Local : - prixHT : réel
- prixTTC : réel
- nombreRemise : entier
- prixTTCAvecRemise : réel
Sortie : void
Début
| **Soit** prixHT = SaisirNombreRéel()
| **Soit** prixTTC = prixHT * (1 + TAUX_TVA)
| **Soit** nombreRemise = prixTTC / TRANCHE_REMISE // division entière !
| **Soit** prixTTCAvecRemise = prixTTC – nombreRemise * VALEUR_REMISE
| Afficher(prixTTCAvecRemise)
Fin

Algorithme 3 : CalculerSurfaceRectangle (variante 1 : on fait naïvement confiance à l'utilisateur)
Principe : Demander deux nombres longueur et largueur, les multiplier et afficher le résultat.
Entrée : /
Local : - longueur : réel
- largueur : réel
- surface : réel
Sortie : void
Début
| **Soit** longueur = SaisirNombreRéel()
| **Soit** largueur = SaisirNombreRéel()
| **Soit** surface = longueur * largueur
| Afficher(surface)
Fin

Algorithme 3 : CalculerSurfaceRectangle (variante 2 : préconditions sans vérification)
Principe : Demander deux nombres longueur et largueur, les multiplier et afficher le résultat.
Précondition : longueur ≥ 0 et largeur ≥ 0
Entrée : /
Local : - longueur : réel
- largueur : réel
- surface : réel
Sortie : void
Début
| **Soit** longueur = SaisirNombreRéel()
| **Soit** largueur = SaisirNombreRéel()
| **Soit** surface = longueur * largueur
| Afficher(surface)
Fin

Algorithme 3 : CalculerSurfaceRectangle (variante 3 : précondition + test et affichage d'une erreur)
Principe : Demander deux nombres longueur et largeur, les multiplier s'ils sont tous deux positifs et afficher le résultat, sinon afficher une erreur.

```
Entrée :        /
Local :         - longueur : réel
                - largeur : réel
                - surface : réel
Sortie :        void
Début
|       Soit longueur = SaisirNombreRéel()
|       Soit largeur = SaisirNombreRéel()
|       Si longueur ≥ 0 ET largeur ≥ 0 Alors
|       |       Soit surface = longueur * largeur
|       |       Afficher(surface)
|       Sinon
|       |       Afficher(erreur, la longueur et/ou la largeur ne sont pas valides !)
|       Fin Si
Fin
```

Algorithme 3 : CalculerSurfaceRectangle (variante 4 : utilisation de la valeur absolue des nombres)
Principe : Demander deux nombres longueur et largeur, multiplier leurs valeurs absolues et afficher le résultat.

```
Entrée :        /
Local :         - longueur : réel
                - largeur : réel
                - surface : réel
Sortie :        void
Début
|       Soit longueur = SaisirNombreRéel()
|       Soit largeur = SaisirNombreRéel()
|       Soit surface = ValeurAbsolue(longueur) * ValeurAbsolue(largeur)
|       Afficher(surface)
Fin
```

Algorithme 3 : CalculerSurfaceRectangle (variante 5 : saisie blindée avec une boucle)
Principe : Demander deux nombres longueur et largeur avec une saisie blindée, les multiplier et afficher le résultat.

```
Entrée :        /
Local :         - longueur : réel
                - largeur : réel
                - surface : réel
Sortie :        void
Début
|       Soit longueur = -1
|       Soit largeur = -1
|       Tant Que longueur < 0 Faire
|       |       Soit longueur = SaisirNombreRéel()
|       Fin Tant Que
|       Tant Que largeur < 0 Faire
|       |       Soit largeur = SaisirNombreRéel()
|       Fin Tant Que
|       Soit surface = longueur * largeur
|       Afficher(surface)
Fin
```

Algorithme 4 : CalculerIMC
Principe : Demander deux nombres taille et poids, calculer l'IMC = poids / taille² et afficher le résultat
Entrée : /
Local : - taille : réel
- poids : réel
- imc : réel
Sortie : void
Début
| **Soit** taille = SaisirNombreRéel()
| **Soit** poids = SaisirNombreRéel()
| **Soit** imc = poids / (taille * taille)
| Afficher(imc)
Fin

Algorithme 5 : CalculerIMCAvecCategories
Principe : Demander deux nombres taille et poids, calculer l'IMC = poids / taille². Déterminer la catégorie 1-5 en fonction de l'IMC et des bornes données (< 18.5, < 25, < 30, < 35, ≥ 40). Afficher l'IMC et la catégorie.
Entrée : /
Local : - taille : réel
- poids : réel
- imc : réel
- categorie : entier
- NORMAL : réel constant = 18.5
- SURPOIDS : réel constant = 25.0
- OBESITE_1 : réel constant = 30.0
- OBESITE_2 : réel constant = 35.0
- OBESITE_3 : réel constant = 40.0
Sortie : void
Début
| **Soit** taille = SaisirNombreRéel()
| **Soit** poids = SaisirNombreRéel()
| **Soit** imc = poids / (taille * taille)
| **Si** imc < NORMAL **Alors**
| | **Soit** categorie = -1
| **Sinon Si** imc < SURPOIDS **Alors**
| | **Soit** categorie = 1
| **Sinon Si** imc < OBESITE_1 **Alors**
| | **Soit** categorie = 2
| **Sinon Si** imc < OBESITE_2 **Alors**
| | **Soit** categorie = 3
| **Sinon Si** imc < OBESITE_3 **Alors**
| | **Soit** categorie = 4
| **Sinon**
| | **Soit** categorie = 5
| **Fin Si**
| Afficher(imc, categorie)
Fin

Algorithme 6 : VerifierValiditeDate
Principe : Demander trois nombres annee, mois et jour. Vérifier que l'année est supérieure à 1959. Calculer le nombre de jours du mois (après avoir déterminé si l'année est bissextile) et vérifier que le mois est entre 1 et 12 et que le jour est entre 1 et le nombre de jours du mois. Si oui la date est valide sinon invalide.

```
Entrée :        /
Local :         - jour : entier
                - mois : entier
                - annee : entier
                - estBissextile : booléen
                - nombreDeJoursDuMois : entier
Sortie :        void
Début
|       Soit estBissextile = faux
|       Soit nombreDeJoursDuMois = 31
|       Soit jour = SaisirNombreEntier()
|       Soit mois = SaisirNombreEntier()
|       Soit annee = SaisirNombreEntier()
|       Si (annee divisible par 4 ET non divisible par 100) OU annee divisible par 400 Alors
|       |       Soit estBissextile = vrai
|       Fin Si
|       Si mois == 4 OU mois == 6 OU mois == 9 OU mois == 11 Alors
|       |       Soit nombreDeJoursDuMois = 30
|       Sinon Si mois == 2 Alors
|       |       Si estBissextile == vrai Alors
|       |       |       Soit nombreDeJoursDuMois = 29
|       |       Sinon
|       |       |       Soit nombreDeJoursDuMois = 28
|       |       Fin Si
|       Fin Si
|       Si annee < 1959 Alors
|       |       Afficher(erreur année invalide !)
|       Sinon
|       |       Si mois < 1 OU mois > 12 Alors
|       |       |       Afficher(erreur, mois invalide !)
|       |       Sinon
|       |       |       Si jour < 1 OU jour > nombreDeJoursDuMois Alors
|       |       |       |       Afficher(erreur, jour invalide !)
|       |       |       Sinon
|       |       |       |       Afficher(date valide)
|       |       |       Fin Si
|       |       Fin Si
|       Fin Si
Fin
```