

TD1 suite – éléments de corrigé (algorithmes)

Algorithme 1 : MeilleureFormuleLocation

Principe : Demander la distance à parcourir (réel) et le nombre de jours de location (entier) puis calculer le prix pour les trois formules (en appliquant les bons coefficients). Comparer les résultats (dans 3 variables) et déterminer la meilleure formule (valeur minimale).

Entrée :

- ESSENCE_JOURNALIER : entier constant = 23
- ESSENCE_KILOMETRAGE : réel constant = 0.16
- DIESEL_JOURNALIER : entier constant = 25
- DIESEL_KILOMETRAGE : réel constant = 0.13
- ELECTRIQUE_JOURNALIER : entier constant = 40
- ELECTRIQUE_KILOMETRAGE : réel constant = 0.03

Local :

- duree : entier
- distance : entier
- valeurEs : réel
- valeurD : réel
- valeurEl : réel

Sortie : void

Début

```
Soit duree = SaisirNombreEntier()
Soit distance = SaisirNombreEntier()
Soit valeurEs = ESSENCE_JOURNALIER * duree + ESSENCE_KILOMETRAGE * distance
Soit valeurD = DIESEL_JOURNALIER * duree + DIESEL_KILOMETRAGE * distance
Soit valeurEl = ELECTRIQUE_JOURNALIER * duree + ELECTRIQUE_KILOMETRAGE *
distance
Si valeurEs ≤ valeurD ET valeurEs ≤ valeurEl Alors
|   Afficher(meilleure formule : essence)
Sinon Si valeurD ≤ valeurEs ET valeurD ≤ valeurEl Alors
|   Afficher(meilleure formule : diesel)
Sinon
|   Afficher(meilleure formule : électrique)
Fin Si
```

Fin

Algorithme 2 : CalculFraisDePort

Principe : Demander le prix unitaire et la quantité de produits. Calculer le prix total et vérifier que l'on soit sous le seuil des frais de port. Si oui, calculer le pourcentage à retirer et vérifier si l'on est en dessous du forfait ou au-dessus. Si en dessous, alors afficher le minimum forfaitaire, sinon afficher le pourcentage calculé.

Entrée :

- SEUIL_FDP : réel constant = 150.0
- POURCENTAGE_FPD : réel constant = 0.15
- MINIMUM_FORFAITAIRE : réel constant = 15.0

Local :

- prixUnitaire : réel
- quantite : réel
- pourcentageFDP : réel
- fraisDePort : réel

Sortie : void

Début

```
Soit prixUnitaire = SaisirNombreRéel()
Soit quantite = SaisirNombreEntier()
Si quantite * prixUnitaire < SEUIL_FDP Alors
|   Soit fraisDePort = POURCENTAGE_FPD * quantite * prixUnitaire
|   Si fraisDePort < MINIMUM_FORFAITAIRE Alors
|   |   Soit fraisDePort = MINIMUM_FORFAITAIRE
|   Fin Si
Fin Si
Afficher(fraisDePort)
```

Fin

Algorithme 3 : CalculTempsTrajetTrain

Principe : Demander deux nombres longueur et largeur, les multiplier et afficher le résultat.

Entrée : /

Local :

- heureDepart : entier
- heureArrive : entier
- minuteDepart : entier
- minuteArrivee : entier
- deltaHeures : entier
- deltaMinutes : entier

Sortie : void

Début

```
Soit heureDepart = SaisirNombreEntier()
Soit minuteDepart = SaisirNombreEntier()
Soit heureArrivee = SaisirNombreEntier()
Soit minuteArrivee = SaisirNombreEntier()
Soit deltaHeures = heureArrivee - heureDepart - 1
Soit deltaMinutes = minuteArrivee + (60 - minuteDepart)
Soit deltaHeures = deltaHeures + deltaMinutes / 60
Soit deltaMinutes = deltaMinutes % 60
Afficher(deltaHeures, deltaMinutes)
```

Fin

Algorithme 4 : HeureDansUneSeconde

Principe : Demander trois entiers (heure, minute et seconde). Ajouter 1 à seconde. Si == 60 mettre à 0 et ajouter 1 à minute. Si == 60 mettre à 0 et ajouter 1 à heure. Si == 24 mettre à 0. Afficher les trois entiers.

Entrée : /

Local :

- heure : entier
- minute : entier
- seconde : entier
- resHeure : entier
- resMinute : entier
- resSeconde : entier

Sortie : void

Début

```
Soit heure = SaisirNombreEntier()
Soit minute = SaisirNombreEntier()
Soit seconde = SaisirNombreEntier()
Soit resHeure = heure
Soit resMinute = minute
Soit resSeconde = seconde
Si seconde < 59 Alors
    Soit resSeconde = resSeconde + 1
Sinon
    Soit resSeconde = 0
    Si minute < 59 Alors
        Soit resMinute = resMinute + 1
    Sinon
        Soit resMinute = 0
        Si heure < 23 Alors
            Soit resHeure = resHeure + 1
        Sinon
            Soit resHeure = 0
        Fin Si
    Fin Si
Fin Si
Afficher(resHeure, resMinute, resSeconde)
```

Fin