

Java Initiation

Exercice 2 : java de base

TP 2.1 : Crypter & Decrypter

Coder une méthode de la classe TP2 :

String CrypterRomain(String cible , String key);

Qui crypte une chaine de caractère cible tels que :

$cible[i] = cible[i] + key[i]$ pour i allant de 0 à la longueur de la chaine de caractère cible

Coder une méthode de la classe TP2

String DeCrypterRomain(String cible , String key)

Qui décrypte une chaine de caractère cible qui a été cryptée par la méthode crypterRomain tels que :

$cible[i] = cible[i] - key[i]$ pour i allant de 0 à la longueur de la chaine de caractère cible

Ecrire une fonction **void TestTP2_1()** qui test les fonctions **CrypterRomain** et **DeCrypterRomain**

Coder une fonction **double Mypow (double r , int N)** ; qui calcule la puissance d'un nombre réel passé en paramètre et élevé à la puissance N qui est un entier

Écrire la procédure **double Suite(double x, int Nn)** qui programme la suite et qui rend la dernière valeur calculée :

- $U_0 = x$
- $U_n = U_{n-1} + (-1)^n * x^{(2^n-1)} / (2^n-1)$

Remarque : Vous devez utiliser la fonction **Mypow**

Ecrire une fonction **void TestTP2_2()** qui test les fonctions la fonction et compare le résultat de **Suite** avec **atan()** et avec un $N > 50$

Soit deux types définis tels que :
Créer les fonctions suivantes :

void permuter(double[] tab, int a, int b) :
qui permute les 2 éléments d'un tableau tab
tels que , tab[a] prend la valeur de tab[b] ,
tab[b] prend la valeur de tab[a] ;

void permuter_Tableau(double [] tab) qui
permute les valeurs du tableau :

void afficher (double [] tab) qui affiche le
tableau de double passé en paramètre.

Ecrire une fonction **void TestTP2_3()** qui
test les fonctions

Tableau avant permutation :

```
Tab[0]=249.05  
Tab[1]=291.41  
Tab[2]=231.28  
Tab[3]=299.57  
Tab[4]= 0.78
```

Tableau apres permutation :

```
Tab[0]= 0.78  
Tab[1]=299.57  
Tab[2]=231.28  
Tab[3]=291.41  
Tab[4]=249.05
```

Soit l'algorithme de seuil suivant tel que :

P le prix d'un article et $P < 1000$

T le taux d'inflation par mois

$P = P * T$ tant que $P < 1000$

N le nombre de mois représente le nombre de boucles effectuées pour atteindre 1000

Coder la fonction **int Seuil(double P, int T)** et qui implémente l'algorithme de seuil et qui rend N

Ecrire une fonction **void TestTP2_4()** qui test la fonction seuil() et qui présente les résultats sous forme de tableau, voir l'exemple d'affichage.

```
Rentrer P > 300
Rentrer T > 1.08
|   P   |   N   |
| 324.00|      1|
| 349.92|      2|
| 377.91|      3|
| 408.15|      4|
| 440.80|      5|
| 476.06|      6|
| 514.15|      7|
Nmax : 7
```

On désire calculer le montant de la facture d'électricité d'un abonné sachant que l'abonné :

1. Paye 200 euros (frais d'abonnement) même s'il n'a rien consommé.
2. Paye sa consommation selon un tarif à tranches :
 - a) 0.100 euros, par KWh, pour les 100 premiers KWh
 - b) 0.175 euros, par KWh, pour les 150 KWh suivants
 - c) 0.225 euros, par KWh, pour la fraction de consommation qui excède 250 KWh
3. Paye la TVA pour un taux de 8%.

Écrire une méthode **int ConsomationKWH(int nouveau , int ancien)** qui prend en paramètre deux valeurs entières : l'ancien index et le nouvel index pour déterminer la consommation d'électricité en KWh (Consommation en KWh = nouvel index - ancien index)

Écrire une méthode **double PayeKWH(int nbKWH , double tva = 8.)** : qui prend en paramètre un entier qui représente le nombre de KWH consommé et un double qui représente la TVA et qui rends le prix à payer (Montant toutes taxes comprises = montant hors taxe * (1 + TVA))

Ecrire une méthode **void TP2_5()** qui teste les deux fonctions précédentes pour 0 , 100 , 250 et 1000 kwh consommées.