

NE PAS IMPRIMER CE DOCUMENT
il va être modifié et complété

Document de travail pour le TP7

Si vous avez un peu de temps, testez :

- reconnaissance d'expressions parenthésées avec un seul type de parenthèses puis avec plusieurs types de parenthèses,
- les algorithmes cités dans le TD sur la complexité que vous n'avez jamais programmé (tri par comptage, recherche du minimum et du maximum dans un tableau).

Multiplications

Pour la méthode de multiplication égyptienne (ou russe), vous pouvez faire afficher les valeurs des opérandes et du produit à chaque étape de calcul.

Pour la multiplication de Karatsuba, écrivez un programme récursif dont les paramètres et le résultat sont des chaînes de caractères (classe `String`).

Choisissez une longueur minimum (par exemple 2 chiffres) à partir de laquelle vous ne ferez pas d'appel récursif à la multiplication de Karatsuba mais une multiplication simple.

Faites afficher les valeurs des paramètres sur les appels récursifs.

Vous pouvez utiliser :

- les fonctions Java `substring(deb, fin)` qui extrait la sous-chaîne de caractères entre les indices *deb* et *fin*
`substring(deb)` qui extrait la sous-chaîne débutant à l'indice *deb*
 les indices varient entre 0 et `length()-1`
 voir <http://docs.oracle.com/javase/6/docs/api/java/lang/String.html/>
- les fonctions Java `Integer.parseInt(ch)` qui convertit la chaîne *ch* en entier
`String.valueOf(valeur)` qui convertit l'entier *valeur* en chaîne de caractères
- la concaténation de « 0 » en fin de chaînes pour les multiplications par des puissances de 10.

Jeu « Le compte est bon »

La plaque de jeu est un tableau d'entiers.

Ecrire une classe `compteBon` avec les méthodes permettant :

- de chercher si un entier donné existe déjà dans la plaque, le dernier indice occupé dans la plaque est donné en paramètre. Le résultat est un booléen ;
- d'effectuer le tirage des entiers distincts qui vont constituer la plaque de jeu. Le nombre d'entiers à choisir est passé en paramètre. Le résultat est le tableau d'entier constituant la plaque ;
- d'afficher la plaque.

Ajoutez à cette classe une méthode `compteAdd` permettant d'effectuer le calcul d'une valeur cible par addition d'entiers de la plaque. Les algorithmes correspondants sont donnés ci-dessous.

```
compteAdd ( plaque, maxPlaque, cible ) :
// méthode récursive de calcul de la valeur cible
// par additions des entiers
// contenus dans plaque entre indice 1 et maxPlaque
trouve = faux
i=1
tant que i<=maxPlaque ET trouve==faux
faire // choix du ieme entier de la plaque
selon
cas cible==plaque[i] : trouve=vrai
cas cible < plaque[i] : // trouve reste à faux
i++
cas cible > plaque[i] :
// appel récursif
// calcul de la nouvelle plaque sans plaque[i]
np=supprimer ( plaque, i, maxPlaque)
trouve=compteAdd(np, max-1, cible-plaque[i])
i++
fcas
ftq
résultat trouve
```

```
supprimer ( plaque, ind, max ) :  
// construction d'une nouvelle plaque de jeu np  
// par recopie de plaque  
// en supprimant l'entier d'indice ind  
i=1 // indice dans plaque  
j=1 // indice dans np  
tant que i < ind  
faire np[j]=plaque[i]  
    i++  
    j++  
}  
si i == ind  
alors i++  
fsi  
tant que i <= max  
faire np[j]=plaque[i]  
    i++  
    j++  
ftq  
résultat np
```

Où devez vous ajouter la construction de la chaîne de caractères qui permettra d'afficher le calcul de la cible dans le cas où une solution a été trouvée ?

Comment faut-il modifier la méthode de calcul pour prendre en compte aussi des multiplications ?

Quid de la prise en compte des soustractions et divisions ?