

Exercices

Exercice 1

Écrire un programme qui récupère deux entiers sur la "ligne de commande" et qui en affiche la somme en fenêtre console, comme dans cet exemple :

$$23 + 25 = 48$$

On vérifiera que les arguments fournis sont formés uniquement de chiffres, dans le cas contraire, le programme s'interrompra.

Exercice 2

Écrire un programme permettant de lire le nom et le prénom d'un étudiant, ensuite transforme le nom en majuscule, la première lettre du prénom en majuscule et le reste en minuscule.

Exercice 3

2	4	5
3	33	32

Tableau 1 (*Tableau Régulier*)

3		
3	4	5
3	5	
3	1	0

Tableau 2 (*Tableau non régulier*)

- 1.Déclarer les deux tableaux ci-dessus:
- 2.Créer une fonction permettant d'afficher les éléments de chaque tableau ligne par ligne.
- 3.Créer un programme de Test (main).

Exercice 4

Créer un tableau de dix chaînes de caractères puis remplir ce tableau avec des adresses e-mail. Calculer ensuite, à partir des informations présentes dans le tableau, la part de marché de chacun des fournisseurs d'accès.

Indice : dans une adresse e-mail, le nom du fournisseur d'accès est la partie située après le caractère @ de l'adresse e-mail.

Exercice 5

Générer trois nombres aléatoires compris entre 0 et 1000, puis vérifier si vous avez deux nombres pairs suivis par un nombre impair. Si ce n'est pas le cas, recommencer jusqu'à ce que vous ayez la combinaison pair, pair, impair. Afficher ensuite le nombre d'essais nécessaires pour obtenir cette combinaison.

Indice : la classe Math propose la méthode statique random qui génère un nombre aléatoire compris entre 0 et 1.

Exemple : `double nb=Math.random();`

Exercice 6

Générer un nombre aléatoire compris entre 0 et 1000. Demander ensuite à l'utilisateur de deviner le nombre choisi par l'ordinateur. Il doit saisir un nombre compris entre 0 et 1000 lui aussi. Comparer le nombre saisi avec celui choisi par l'ordinateur et afficher sur la console « c'est plus » ou « c'est moins » selon le cas. Recommencer jusqu'à ce que l'utilisateur trouve le bon nombre. Afficher alors le nombre d'essais nécessaires pour trouver la bonne réponse.

Indice : pour récupérer les caractères saisis au clavier, nous avons à notre disposition le flux System.in. Malheureusement, celui-ci ne propose que des fonctions rudimentaires pour la récupération des saisies de l'utilisateur (lecture caractère par caractère). Pour une utilisation plus confortable, il vaut mieux utiliser un objet Scanner. Nous aurons ainsi à notre disposition une série de fonctions permettant la récupération d'entiers, de float, de chaînes de caractères... Ces fonctions sont nommées nextxxxx où xxxx doit être remplacé par le type de données que l'on souhaite obtenir, par exemple nextInt pour un entier, nextLine pour une chaîne de caractères, etc.

```
String chaine;  
Scanner sc;  
sc=new Scanner(System.in);  
chaine=sc.nextLine();
```

Exercice 7

Ajouter au jeu de l'exercice 6 l'affichage du temps mis par l'utilisateur pour obtenir la bonne réponse