

TD 2

Types primitifs, instructions de contrôle

Exercices table : 3, 4, 6, 10, 11, 17, 18.

Rappel : respectez les consignes de la page *Fonctionnement*.

1. Placez-vous dans le répertoire adéquat indiqué dans la page *Environnement*.
2. **Familiarisation** Tapez et exécutez tous les programmes de la feuille d'exemples du cours, en respectant les consignes de la page *Fonctionnement*.
3. Calculer la somme des entiers entre 0 et n . Testez, avec plusieurs valeurs, et en utilisant pour vérifier la formule (admise) $n(n+1)/2$.
4. Lire un entier n et afficher la somme des entiers impairs entre 0 et n .
5. Afficher $\sqrt{3}$, $\cos(\pi)$, $\sin(\pi/6)$. Tester (exécuter et comparer avec le résultat attendu, en particulier pour $\sin(\pi/6)$).
6. Ecrire la factorielle, en utilisant une variable de type *int*. Tester pour tous les n de 0 à 7 uniquement.
7. Reprenez l'exercice 6 sur la factorielle. Testez maintenant pour tous les n de 0 à 20. Expliquer ce qui se passe à partir d'un certain n . Dites ce qu'il faudrait faire pour corriger, puis faites-le.
8. Ecrire un programme qui lit sur la ligne de commande un prénom x et un entier. Si l'entier est 6 (représente F), il affiche *Bonjour Mademoiselle x*. Si c'est 13 (représente M), il affiche *Bonjour Monsieur x*. Sinon il affiche *Bonjour E.T. x*. Tester. Vous n'utiliserez que des *if* (pas de *switch*).
9. Reprendre l'exercice 8 mais avec un *switch*.
10. Ecrire un programme qui tire aléatoirement un entier entre 0 et 10 inclus, et affiche s'il est pair ou impair. (Inspirez-vous l'exemple de tirage aléatoire dans la feuille d'exemples.)
11. Faites 10 tirages aléatoires entre 0 et 10 inclus, et à chaque fois dites si le nombre est pair ou impair (faut-il une boucle *for* ou *while* ?).
12. Tirez des entiers aléatoires entre 0 et 10 inclus (en les affichant), jusqu'à ce que l'un d'eux soit strictement supérieur à 7, et affichez alors le nombre de tirages qu'il vous a fallu.
13. Machine à café. Lire le prix p du café (99 centimes maximum), puis tirer des pièces aléatoirement pour payer le café. Comme une vraie machine, afficher après chaque pièce sa valeur et le montant restant à régler. Quand il est payé afficher la monnaie rendue. (Pour le tirage des pièces, vous tirerez un entier entre 1 et 6 inclus, et à chaque entier vous associez une pièce : pièce de 1 centime à 1, pièce de 2 centimes à 2, de 5 à 3, de 10 à 4, de 20 à 5, et de 50 à 6.)
14. Lire n au clavier et afficher n lignes de n étoiles (*).
15. Lire n au clavier et afficher n lignes d'étoiles (*), la première contenant une étoile, la deuxième deux, ... et la n -ième n . Autrement dit c'est un triangle, la première colonne contient n étoiles, la deuxième $n-1$, ..., la dernière une.

16. Lire n au clavier et afficher n lignes d'étoiles (*), la première contenant n étoiles, la deuxième $n - 1$, ... et la n -ième une. Autrement dit c'est un triangle sur la pointe, la première colonne contient n étoiles, la deuxième $n - 1$, ..., la dernière une.
17. Lire un entier n et afficher la factorielle de chaque entier de 0 à n . Votre code devra tourner pour $n = 7$, mais cela n'est pas demandé pour $n > 7$. Ne pas utiliser de fonction.
18. Affichez le plus petit entier n tel que $1/2^n < \epsilon$ lu au clavier. Remarque : comme $u_{i+1} = u_i/2$, il est inutile de recalculer en totalité $1/2^i$ à chaque pas, ce qui évite aussi d'imbriquer des boucles. Vous pourrez maintenir une variable *terme Courant* contenant $1/2^n$ à l'étape n . Affichez toutes les valeurs à chaque passage dans la boucle. (Vous utiliserez *Double.parseDouble* pour transformer une chaîne en *double*.)
19. Affichez le plus petit entier n tel que $2 - \sum_{i=0}^n 1/2^i < \epsilon$ lu au clavier. (Inspirez-vous de l'exercice correspondant du TD 2, même si ce n'est pas exactement la même question.)
20. *Déroutement mémoire. Travail personnel.*
On considère l'exécution du programme suivant (il compile sans erreur).
 - (a) Donner la configuration de la mémoire immédiatement après l'exécution de la ligne contenant le commentaire : // ici. Vous utiliserez un croquis détaillé et l'algorithme vu en cours et TD. En particulier, on allouera obligatoirement les cases mémoire dans l'ordre croissant à partir de 101. Comme en cours et en TD, si une case contient successivement plusieurs valeurs, on les écrira de gauche à droite dans la case en les barrant d'un seul trait au fur et à mesure. De même, on ne réutilisera pas les zones de fonctions.
 - (b) Donner le dernier état de la mémoire avant l'arrêt du programme.
 - (c) Dire ce qu'affiche ce programme.

```
// ExMemoire.java

class ExMemoire {
    public static void main(String[] args) {
        int n = 3;
        for (int i=0; i<n-1; i++)
            for (int j=i+1; j>0; j--)
                System.out.print(j+" "+i+" ");
        System.out.println(); // ici
    }
}
```

Si vous avez fini tous les exercices ci-dessus avant la fin du TD

Choisissez ce que vous voulez dans la suite. Les exercices 21 à 24 sont progressifs mais délicats en s'amusant avec les étoiles. Les 26 à 29 sont similaires, mais sur les entiers, plus scolaires donc mais aussi utiles.

21. reprenez l'exercice 15, mais cette fois c'est la dernière colonne qui contient n étoiles, l'avant-dernière $n - 1$, ..., la première une.
22. Lire n au clavier et afficher n lignes d'étoiles (*), la première contenant n étoiles, la deuxième $n - 1$, ... et la n -ième une. Autrement dit c'est un triangle, mais cette fois c'est la n -ième colonne qui contient n étoiles, l'avant-dernière $n - 1$, ..., la première une.

23. Lire n et afficher n lignes de une étoile (*). Sur la première l'étoile est en première position tout à gauche, sur la deuxième elle est en position 2, ..., sur la n -ième en position n . Autrement dit c'est une diagonale descendante.
24. Lire n et afficher n lignes de une étoile (*). Sur la première l'étoile est en position n tout à droite, sur la deuxième elle est en position $n - 1$, ..., sur la n -ième en position 1. Autrement dit c'est une diagonale montante.
25. Lire n au clavier et afficher n lignes, chacune de n entiers, la première ne contenant que des 0, la deuxième que des 1, ..., et la dernière que des n .
26. Lire n au clavier et afficher n lignes de n colonnes comme suit. La première colonne ne contient que des 0, la deuxième que des 1, ..., la n -ième que des n .
27. Lire un entier n et afficher $n + 1$ lignes, chacune contenant n entiers séparés par des espaces : $n, n - 1, \dots, 2, 1$, dans cet ordre.
28. Lire n au clavier et afficher n lignes, chacune de n entiers, la première ne contenant que des n , la deuxième que des $n - 1$, ..., et la dernière que des 1.
29. Lire n au clavier et afficher n lignes de n entiers comme suit. La première colonne ne contient que des n , la deuxième que des $n - 1$, ..., la n -ième que des 1.
30. Comme pour $1/2^n$ ci-dessus, tester la convergence de la série $\sum 1/n^2$ vers $\pi^2/6$.