

Algorithmique - TP2

IUT 1ère Année

30 septembre 2012

1 Construire des nombres aléatoires

Concernant plusieurs exercices, vous serez amenés à produire des nombres aléatoires. La procédure en C++, donnée par le code ci-dessous, permet de construire dix entiers aléatoires entre 1 et n .

Listing 1 – entierAleatoire.cpp

```
#include <cstdlib>
#include <ctime>
#include <iostream>
using namespace std;

int main()
{
    int n, entier_aleatoire;

    cout << "Taille de l'intervalle: ";
    cin >> n;

    srand((unsigned)time(0));

    for(int index=0; index<10; index++)
    {
        entier_aleatoire = (rand() % n) + 1;
        cout << entier_aleatoire << endl;
    }

    return 0;
}
```

2 Exercices du TP2

Exercice 1. Construire un programme permettant de donner le résultat de la suite harmonique.

- Données : un nombre entier positif n
- Résultat : le résultat de la suite harmonique : $\sum_{i=1}^n \frac{1}{i}$

Exercice 2. Construire un programme permettant d'évaluer vos chances de gagner dans l'ordre ou dans le désordre au tiercé, quarté ou quinté.

Exercice 3. Écrire un programme permettant à deux joueurs de jouer au jeu des allumettes : 21 allumettes sont disposées sur une table, à tour de rôle chaque joueur retire 1, 2 ou 3 allumettes. Le joueur ramassant la dernière allumette a perdu. A chaque tour vous afficherez les allumettes se trouvant sur la table (une allumette sera représentée par le caractère I) ainsi que le numéro du joueur qui doit jouer. Les saisies réalisées (le nombre d'allumettes à retirer) doivent être vérifiées et redemandées dans le cas où elles ne sont pas correctes.

Exercice 4. Écrire un programme permettant de calculer la somme des éléments d'un tableau. Le tableau en question contient 100 entiers positifs générés de manière aléatoire entre 0 et 10.

Exercice 5. Écrire un programme permettant de retrouver l'index du plus grand élément d'un tableau. Comme précédemment, le tableau contient 100 entiers positifs générés de manière aléatoire entre 0 et 10.

Exercice 6. Écrire un programme permettant de retrouver le nombre d'occurrences d'un élément x entre 0 et 10 dans un tableau. A nouveau, le tableau contient 100 entiers positifs générés de manière aléatoire entre 0 et 10.

Exercice 7. Écrire un programme permettant de trier un tableau. Vous utiliserez la méthode de tri que vous préférez. Le tableau initial contient 1000 entiers positifs générés de manière aléatoire entre 0 et 100.

Exercice 8. Écrire un programme permettant d'effectuer une recherche dichotomique d'un élément entre 0 et 100 dans un grand tableau trié. Pour cela, vous utiliserez le tableau trié obtenu à partir du programme de l'exercice 7.

Pour les exercices 9, 10, 11 et 12, vous pourrez vous mettre en binôme. Au *blackjack*, chaque carte d'un jeu de 52 cartes possède

- une *face* parmi les 13 figures possibles : 2,3,4,5,6,7,8,9,10,valet,dame,roi,as,
- une *couleur* parmi les 4 couleurs possibles trèfle, carreau, pique, coeur,
- une *valeur* donnée dans le tableau ci-dessous.

de 2 à 10	valeur nominale
valet, dame, roi	10
as	11

Rappelons qu'un *talon* (ou une *pioche*) est un jeu de cartes, faces cachées, initialement mélangées (ou *battues*).

La seule difficulté de ces exercices est de *modéliser* le jeu de cartes du blackjack (un tableau ? plusieurs tableaux?).

Exercice 9. Écrire un programme qui permet de choisir aléatoirement une carte dans un jeu de 52 cartes.

Exercice 10. En vous basant sur l'exercice précédent, inventez une méthode permettant de *battre* un jeu de 52 cartes. Écrire en C++ son programme correspondant.

Exercice 11. Écrire un programme qui pioche m cartes dans le talon (initialement plein) et affiche la somme des valeurs de ces cartes.

Exercice 12. Écrire un programme permettant de jouer au *blackjack* (version simplifiée) entre un joueur et la banque.

- La banque commence par battre les cartes, puis distribue une carte au joueur et une carte à elle-même.
- Une fois que le joueur a reçu sa carte, il annonce "carte !" s'il veut une carte supplémentaire. Il peut faire l'annonce autant de fois qu'il veut jusqu'à ce qu'il dise "je reste !". Si le joueur atteint 21 il a gagné, et s'il dépasse 21 il a perdu.
- Enfin (si le joueur n'a pas atteint ni dépassé 21), la banque tire des cartes jusqu'à qu'elle dépasse 21 (elle perd) ou qu'elle dépasse le joueur mais reste inférieure ou égale à 21 (elle gagne).