

## TP3

Programmez une procédure qui calcul et affiche le tirage du loto de 3 manières différentes :

- Utilisation de la boucle « for » à l'exception des autres boucles
- Utilisation de la boucle « while » à l'exception des autres boucles
- Utilisation de la boucle « do » à l'exception des autres boucles

Les sept numéros doivent être tous différents. L'algorithme est présenté ci-dessous.

Le code de la fonction Hasard(n) qui retourne un entier  $x$ ,  $1 \leq x \leq n$  est dans le fichier :

/net/Bibliotheque/AS/TP\_ASD\_PROG/tp3.cc

### Action Tirage\_Loto()

constantes NBNUM = 49, NBSORT = 7

variables

```
tab_numéros_pos : tableau[0..NBNUM] d'entiers
/      / on "gaspille" la case d'indice 0 pour faciliter la lisibilité.
i, nb_numéros_pos, indice_tirage : entier
```

### début

```
// initialisation : tous les numéros sont disponibles
```

```
Pour i de 0 à NBNUM faire tab_numéros_pos [i] ← i
```

```
nb_numéros_pos ← NBNUM
```

```
// tirage
```

```
Pour i de 1 à NBSORT
```

```
faire début
```

```
indice_tirage ← Hasard(nb_numéros_pos)
```

```
écrire tab_numéros_pos [indice_tirage]
```

```
Supprimer_Elem_Indice(indice_tirage, tab_numéros_pos, nb_numéros_pos)
```

```
fin
```

```
fin
```

### Action Supprimer\_Elem\_Indice(

```
E indice : entier ;
```

```
E/S tab : tableau d'entiers, taille : entier)
```

### début

```
tab[indice] ← tab[taille] ; taille ← taille - 1 ;
```

```
fin
```

### Code de la fonction Hasard en c++

```
// auteur : Colette Johnen
```

```
#include <cstdlib> // srand, rand
```

```
#include <ctime> // time
```

```
#include <iostream> // cout, cin
```

```
using namespace std;
```

```
int Hasard(int n){
```

```
static bool first = 1;
```

```
if (first) { srand (time(NULL)); first = 0; cerr << "first time " << endl; }
```

```
return((rand() % n)+1);
```

```
}
```

## Exercice facultatif

Programmez une procédure de trois manières différentes :

- Utilisation de la boucle « for » à l'exception des autres boucles
- Utilisation de la boucle « while » à l'exception des autres boucles
- Utilisation de la boucle « do » à l'exception des autres boucles

La procédure affiche le nombre de jetés de trois dès correspondant (par sommation des valeurs des dès) à valeur.

**Action Calculer\_nombres\_coup\_3\_des(valeur : entier)**

variables

compteur, d1, d2 : entiers

fin : booléen

**Début**

compteur ← 0

fin ← false

Si valeur < 3 Alors fin ← true

Si valeur > 18 Alors fin ← true

Si (fin == false) Alors

début

d1 ← maximum(1, valeur-12)

Tant que d1 ≤ minimum(6, valeur -2)

Faire début

d2 ← maximum(1, valeur-d1-6)

Tant que d2 ≤ minimum(6, valeur-d1-1)

Faire début

compteur ← compteur +1

d2 ← d2 +1

fin

d1 ← d1 +1

fin

fin

écrire compteur

**fin**