

## Contrôle continu (2 heures)

Le sujet vous laisse libre de programmer en style orienté-objet (avec des classes) ou en décrivant une structure de données encapsulée et des fonctions sur cette structure de données. Les fichiers .h correspondant à ces deux approches sont fournis à la fin du sujet.

Vous préciserez au début de votre copie le choix de programmation que vous avez fait et vous devrez rester cohérent avec ce choix tout au long de votre composition.

Les deux parties sont indépendantes.

### 1 Calcul de variance sur plusieurs ensembles

On considère  $n$  ensembles finis  $F_1 \dots F_n$  de valeurs réelles. On désire calculer la variance de l'ensemble  $F = \bigcup_1^n F_i$  sans avoir à connaître tous les éléments des  $F_i$ .

On va pour cela utiliser le théorème de König-Huyghens qui établit que la variance d'un ensemble est égale à la moyenne des carrés de cet ensemble moins le carré de la moyenne de cet ensemble :

$$V[X] = \sigma_X^2 = E[X^2] - E[X]^2$$

**1-** Proposez une structure de données `donnees` permettant de stocker le cardinal d'un ensemble, la moyenne de ses éléments et la moyenne des carrés de ses éléments. Les éléments eux-mêmes ne sont pas stockés. Quelles sont les valeurs à stocker pour l'ensemble  $F = \{1, 2\}$  ?

**2-** Proposez une opération de construction des données sur un ensemble réduit à un unique élément. Le seul argument sera la valeur de cet élément.

**3-** Proposez une opération de construction des données sur un ensemble  $F$  à partir des données sur deux ensembles  $F_1$  et  $F_2$  formant une partition de  $F$ . Les seuls arguments seront les structures `donnees` concernant les deux sous-ensembles. Déduisez-en une opération recalculant les données concernant un ensemble lorsqu'on y ajoute une valeur.

**4-** En utilisant la formule déduite du théorème de König-Huyghens, proposez une opération calculant la variance et l'écart-type d'un ensemble à partir d'une structure `donnees` le concernant.

**5-** On souhaite disposer d'un mécanisme d'affichage des données sur un ensemble. Écrivez le code de la fonction d'entête `ostream& operator<<(ostream &o, donnees d)` permettant de surcharger l'opérateur `<<`. Précisez si le paramètre de type `donnees` est passé par adresse ou par valeur. Justifiez votre réponse. Expliquez pourquoi cette fonction doit être déclarée avec le mot-clé `friend` lorsqu'elle se trouve dans une classe.

**6-** On désire maintenant calculer récursivement les données statistiques d'un ensemble. Écrivez une fonction qui, à partir d'un tableau de réels, d'un indice de départ  $i$  et d'un indice d'arrivée  $j$ , avec  $i \leq j$ , va calculer récursivement les données sur les deux moitiés de l'intervalle  $[i, j]$  puis en déduira, à l'aide de la fonction écrite à la question 3 (que vous pouvez considérer écrite si vous n'avez pas su répondre), les données sur l'intervalle  $[i, j]$ . Proposez alors une fonction qui, à partir d'un tableau de réels et de sa taille  $n$ , calcule les données statistiques sur l'intervalle  $[1, n]$ .

**7-** Proposez une fonction de saisie des données `istream& operator>>(istream &i, donnees &r)`. Précisez si l'argument `r` est passé par adresse ou par valeur. Justifiez votre réponse.

### 2 Représentation concrète d'un ensemble

On va maintenant chercher à représenter concrètement les ensembles. Pour cela, on propose d'utiliser un tableau de taille  $N$  (connue à la compilation). Ce tableau permettra de représenter

les ensembles d'entiers de cardinal inférieur à  $N$ . Chaque case contiendra un élément de l'ensemble. La taille de l'ensemble décrit dans le tableau sera stockée dans un autre champ de la structure de données.

8- Ecrire la structure de données permettant de représenter ces ensembles. Comment sera représenté concrètement l'ensemble vide? et l'ensemble  $\{3,5,9\}$ ?

9- Proposez deux constructeurs, l'un pour construire l'ensemble vide, et l'autre pour construire un ensemble singleton réduit à un unique élément.

10- Proposez une opération pour faire l'union de deux ensembles.

11- Proposez une opération permettant de construire un objet de type `donnees` à partir d'un ensemble.

12- Proposez une opération pour tester si un ensemble et une structure de type `donnees` sont compatibles.

## A Fichers .h

### A.1 Cas des structures de données

```
struct donnees {
    ...
};

donnees un_seul_element(float e);

donnees union_d_ensembles(donnees d1, donnees d2);
donnees ajoute_element(donnees d, float e);

float variance(donnees d);
float ecart_type(donnees d);

std::ostream& operator<<(std::ostream&o, donnees d);

donnees calcul_recuratif(float *tab, int i, int j);
donnees calcul_donnees(float *tab, int n);
```

### A.2 Cas des classes

```
class donnees
{
    // structure de données choisie

public:
    donnees(float e);

    donnees(donnees d1, donnees d2);
    donnees(donnees d, float e);

    float variance();
    float ecart_type();

    friend std::ostream& operator<<(std::ostream&o, donnees d);
};

donnees calcul_recuratif(float *tab, int i, int j);
donnees calcul_donnees(float *tab, int n);
```