UNIVERSITÉ SAVOIE MONT BLANC

Travaux dirigés 2
Structure et Vector
Introduction au c++
—MMAA—

▸ **Exercice 1.** ***Valeurs aléatoires et histogramme***

Cet exercice propose de calculer l'histogramme d'un ensemble de notes calculées de manière aléatoire.

1. *Sachant que* `time()` *renvoie le nombre de secondes écoulées depuis le 1 janvier 1970. Lire et comprendre les instructions suivantes*

```
#include <cstdlib>
#include <ctime>
#include <iostream>
#include <unistd.h>

int main(){
    srand(time(NULL));
    std::cout << (rand())%10 <<std::endl;
    std::cout << (rand())%10 <<std::endl;
    std::cout << (rand())%10 <<std::endl;
}
```

   Écrire une fonction `void initAlea(std::vector<int>* t,int taille, int max)` *qui remplit les* `tailles` *premières cases du* `std::vector t` *avec des entiers positifs* ***aléatoires*** *inférieurs à* `max`*.*

2. *Écrire une fonction* `int position(std::vector<int> t,int taille, int x);` *prenant en paramètre un* `std::vector` *, sa taille et un entier x, et qui renvoie l'indice de la première occurrence de x dans t si x apparaît dans les* `taille` *premiers éléments de t et -1 sinon.*

3. *Écrire une fonction* `histogramme` *qui reçoit deux* `std::vector` *d'entiers Notes et Histo. Le* `std::vector` *Notes a pour taille N et contient des notes de 0 à 20. Le* `std::vector` *Histo devra représenter, après l'appel de la fonction, l'histogramme des notes du premier tableau, c'est à dire que Histo[i] est le nombre de valeurs égales à i dans le vector Notes.*

▸ **Exercice 2.** ***Analyse d'une chaine de caractères***

Soit `chaine` *une chaine de caractères initialisée à "3,2,1". En utilisant la fonction* `atoi`*, et le fait que le type string est un conteneur de la stl, extraire les 3 entiers présents dans la chaîne.*

▸ **Exercice 3.** ***Polygone***

En reprenant la structure `Point`*, faire les programmes suivants:*

1. *Définir une structure* `Polygone` *possédant deux champs : un pour stocker sa taille (son nombre de sommets) et l'autre pour stocker l'ensemble de ses points.*
2. *Ecrire la fonction membre* `push_back` *permettant d'ajouter un point à un polygone.*

▸ **Exercice 4.** ***Lecture et écriture de fichiers***

1. *Écrire une méthode* `save(std::string filename,std::vector monContenu)` *qui sauvegarde dans le fichier* `filename` *les composantes du vecteur.*
2. *Écrire une méthode* `load(std::string filename)` *qui lit un ensemble de coordonnées de sommets d'un polygone dans le fichier* `filename` *puis qui affecte le résultat dans une variable de type Polygone.*