

TP1 – Standard Template Library

L'objectif de la séance est de s'approprier la syntaxe des patrons de fonctions et de savoir utiliser et manipuler différents conteneurs en se référant à la documentation en ligne cpluplus.com.

1. Pile ou face

On souhaite écrire un programme de calcul d'expressions arithmétiques simples sur la base de leur description postfixée. Par exemple pour l'expression $(12+3)*4$, on aura "12" "3" "+" "4" "*". Cette expression est stockée dans un conteneur `<vector>`.

L'algorithme d'évaluation procède de la manière suivante :

1. On empile tout nombre (après conversion) dans la pile d'évaluation `<stack>`.
 2. On dépile deux nombres (opérandes) de la pile pour calcul lorsqu'on a une opération : le résultat est empilé en tête de la pile.
 3. Le résultat final correspond à la dernière valeur sur la tête de pile.
 4. On vide les conteneurs à la fin du calcul.
- Donnez la déclaration C++ du tableau conteneur `<vector>`.
 - Donnez la définition d'une fonction de saisie de l'expression postfixée par l'utilisateur avec arrêt de la saisie après appui de la touche « ! ».
 - Donnez la définition d'une fonction qui permet de renvoyer le résultat du calcul de l'expression passée en paramètre conformément à l'algorithme décrit ci-dessus.

2. Entraînement course 200 mètres

Un champion sprinteur effectue une course d'entraînement typique de 200 mètres dans les temps suivants : 26.1, 25.6, 25.7, 25.2, 24.7 et 25.0 secondes.

On souhaite connaître le temps minimal, le temps maximal, le temps moyen et le temps médian du champion.

A partir d'un conteneur `<vector>` contenant les différents temps du champion, calculez les temps significatifs demandés* et stockez les différents temps dans une table associative `<map>` puis affichez la médiane.

*vous utiliserez autant que possible les itérateurs `begin` et `end`.

Calcul de la médiane

Si les observations d'une variable sont ordonnées par valeur, la valeur médiane correspond à l'observation qui se trouve au point milieu de cette liste ordonnée. Elle correspond plus précisément à un pourcentage cumulé de 50 % (c'est-à-dire que 50 % des valeurs sont supérieures à la médiane et 50 % lui sont inférieures).

La position de la médiane est la valeur,

$\frac{n+1}{2}$, le n désignant le nombre de valeurs dans un ensemble de données.

Pour calculer la médiane, il faut d'abord classer les données (les trier dans l'ordre ascendant). La médiane est le nombre qui se situe au point milieu. Si le nombre de valeurs est pair il faudra alors faire la moyenne des 2 valeurs contiguës au point milieu.

Annexes

Méthodes de la classe `std::stack`

`void pop()` : supprime l'élément en tête de stack
`void push(T& unV)` : ajoute unV en tête de stack
`T& top()` : renvoie l'élément en tête de stack
`unsigned int size()` : donne le nombre d'éléments
`bool empty()` : renvoie true si le stack est vide

Modifieurs `std::vector`

assign	Assign vector content (public member function)
push_back	Add element at the end (public member function)
pop_back	Delete last element (public member function)
insert	Insert elements (public member function)
erase	Erase elements (public member function)
swap	Swap content (public member function)
clear	Clear content (public member function)
emplace <small>C++11</small>	Construct and insert element (public member function)
emplace_back <small>C++11</small>	Construct and insert element at the end (public member function)

Algorithmes

`void sort(InputIterator first, InputIterator last)` : trie les éléments d'un conteneur.

function template

`std::stoi` C++11 `<string>`

```
int stoi (const string& str, size_t* idx = 0, int base = 10);
int stoi (const wstring& str, size_t* idx = 0, int base = 10);
```

Convert string to integer

Parses *str* interpreting its content as an integral number of the specified *base*, which is returned as an `int` value.

If *idx* is not a null pointer, the function also sets the value of *idx* to the position of the first character in *str* after the number.

The function uses `strtol` (or `wcstol`) to perform the conversion (see `strtol` for more details on the process).