

CPE Lyon - 3ICS - Année 2020/21

Développement informatique - 1

TP 3 – Structure de données

(travail individuel, temps estimé : 8h)

1/ Travail à faire

Pour ce TP vous allez devoir récupérer le projet « DataStructure » sur le serveur. Ce projet contient les déclarations de différentes fonctions permettant de manipuler des piles, des files et des listes et les tas. L'implémentation sera basée sur l'utilisation de tableaux.

Partie 1 : implémentation d'une pile

Vous devez coder les fonctions suivantes :

```
void init_stack(Stack s);
void push(Stack s, float value);
float pop(Stack s);
bool is_stack_empty(Stack s);
float peek(Stack s); // aka top or front
void dup(Stack s);
void swap(Stack s);
void clear(Stack s);
```

Partie 2 : implémentation d'une file

Vous devez coder les fonctions suivantes :

```
void init_queue(Queue q);
void enqueue(Queue q, float value);
float dequeue(Queue q);
bool is_queue_empty(Queue q);
float front(Queue q); // aka peek
void clear(Queue q);
```

Partie 3 : implémentation d'une liste

Vous devez coder les fonctions suivantes :

```
void init_array_list(Array_list l);
void insert_at(Array_list l, int position, float value);
```

```
void add(Array_list l, float value);
float remove_at(Array_list l, int position);
float get_at(Array_list l, int position);
void clear(Array_list l)
```

Partie 4 : Implémentation d'un tas

```
void init_heap(Heap *h);
bool is_heap_empty(Heap *h);
float pop(Heap *h); //pop the root value
void push(Heap *h, float value);
float replace(Heap *h, float value); // pop root and push a new key.
float peek(Heap *h); // return root value but dont remove it

//En option
void heapify(Heap *s, float array[], size_t array_size); //create Heap from array
void merge(Heap *dest, Heap *src); //merge src into dest
```

2/ Consignes

-Vérifiez le comportement de vos fonctions (créez une fonction test_stack, test_queue test_arraylist, test_heap dans le fichier main.c. Ces fonctions contiendront le code que vous avez utilisé pour tester vos implémentations)

-Utilisez les assertions dans un premier temps.

-Utilisez les frameworks de test unitaire. Cunit est particulièrement simple à prendre en main (surtout avec Netbeans, après l'installation des packages CUnit et CUnit-devel)

-Vos fonctions doivent être annotées (javadoc ou doxygen)