

## I. Présentation générale :

### 1. Objet du TP :

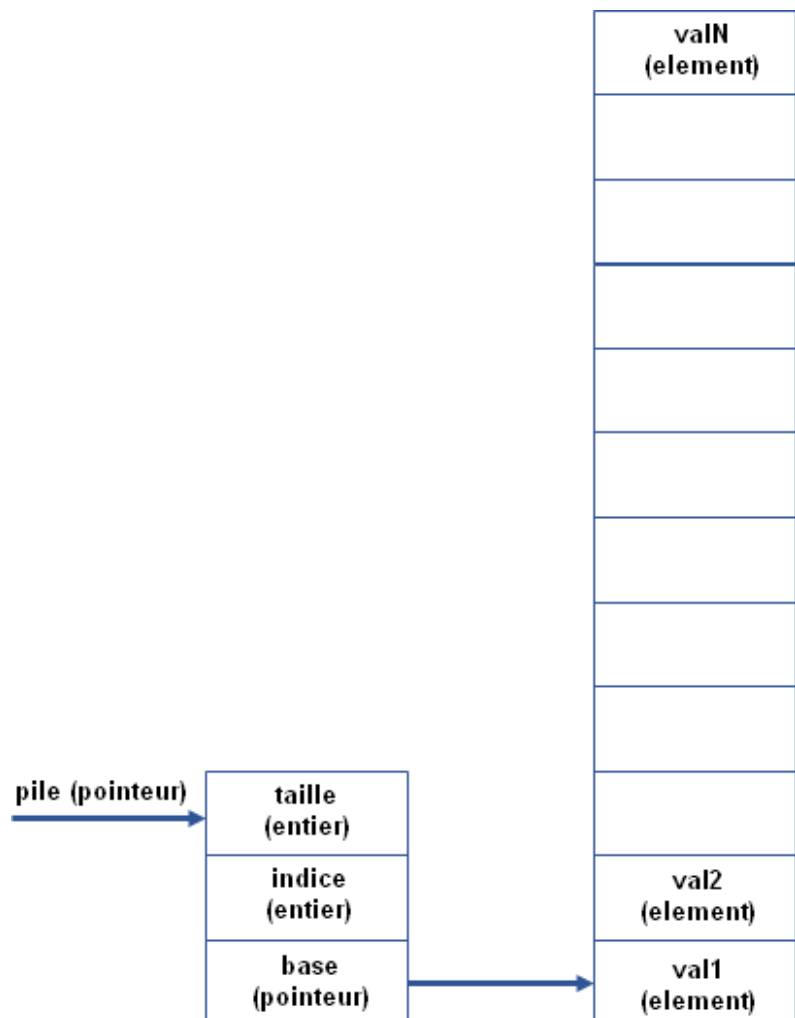
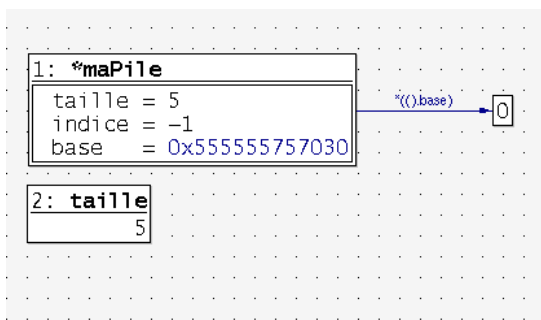
Implémenter les principes vus en cours de SDD à propos des files et piles, créer deux modules regroupant ces diverses fonctions. Vérifier le bon fonctionnement des deux modules sur l'exemple d'une inversion de pile.

### 2. Structures utilisées :

- Pile :

- Pile pointe sur une liste contiguë de 3 cases

- Taille (entier) : nombre de case de la pile
    - Indice (entier) : indice de l'élément au sommet
    - Base (pointeur) : pointe sur la liste contiguë des éléments de la pile

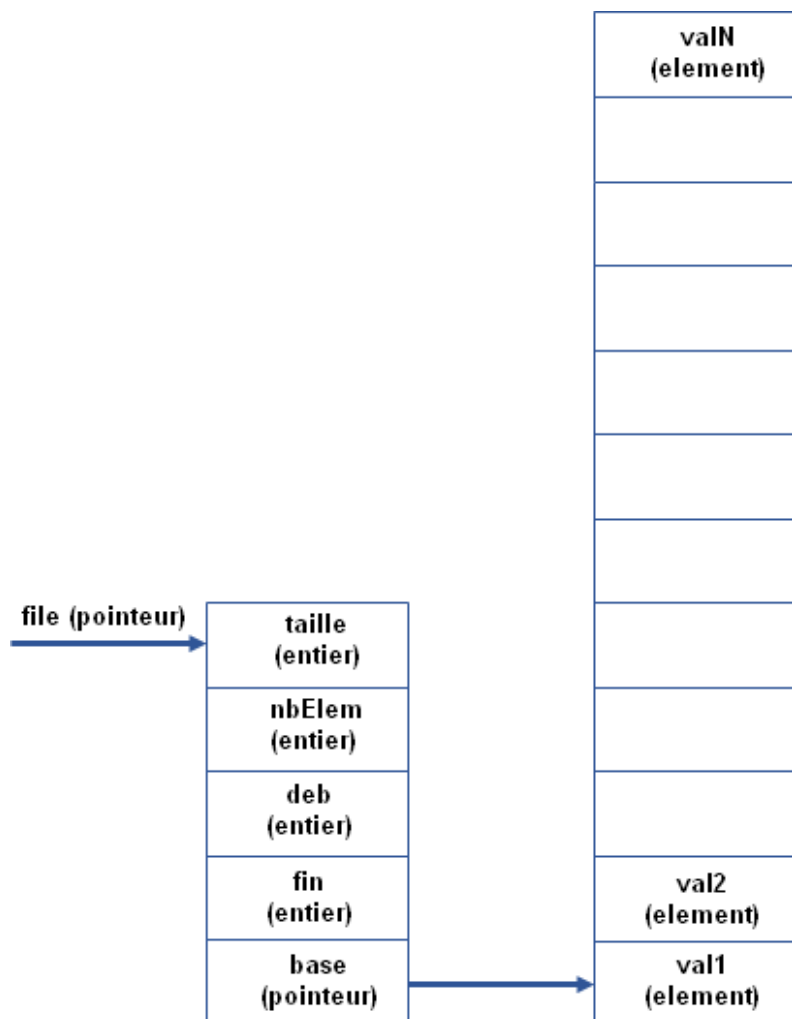


- File :
  - File pointe sur une liste contiguë de 3 cases
    - Taille (entier) : nombre de case de la file
    - nbElem (entier) : nombre d'éléments actuellement dans la file
    - deb (entier) : indice du premier élément arrivé
    - fin (entier) : indice du dernier élément arrivé
    - Base (pointeur) : pointe sur la liste contiguë des éléments de la file

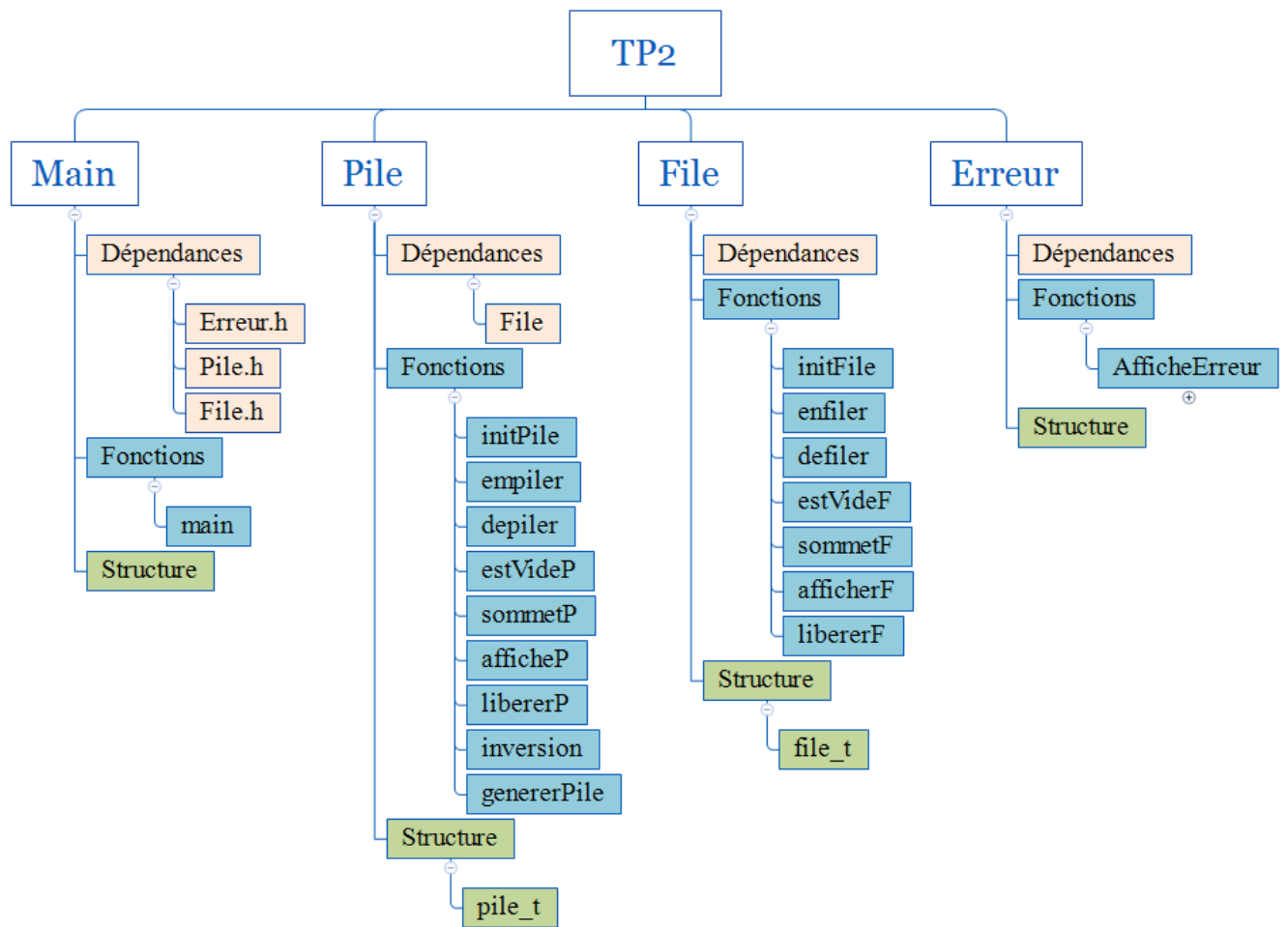
1: *maFile
taille = 5
nbElem = 0
deb = 0
fin = -1
base = 0x555555757030

2: taille
5



### 3. Organisation du code :



## II. Détails des fonctions

L'ensemble des captures d'écran sont tirées du fichier texte : doc.txt

```
1  -----
2      main
3      Rôle : Tester le fonctionnement des fonctions du
4              module Pile et File en inversant une pile
5      Principe : Génération d'une pile
6                  Affichage pile
7                  Inversion (confère fonction inversion)
8                  Affichage pile inversée
9
10     Entrées : - Taille de la pile
11                - Nombre d'élément dans la pile
12                - Si ces valeurs ne sont pas renseignées:
13                    la pile est initialisée à :
14                        taille = 10
15                        NbElement = 10
16
17     Sortie : -
18     Sous fonctions : Affiche Erreur
19                     genererPile
20                     afficheP
21                     inversion
22                     libererP
23
24     Dépendance : Pile.h
25                  File.h
26                  Erreur.h
27  -----
28
29  *****
30  *                               *
31  *****
32  -----
33      initPile
34
35      Rôle : Ajouter une valeur au sommet de la pile
36      CONDITION NÉCESSAIRE : -
37
38      Entrée : taille : taille de la pile (nombre d'élément max)
39      Sortie : pointeur sur structure pile (confère fichier en-tête)
40                indice du premier élément = -1
41                NULL si allocation échouée
42
43      Variable locales : -
44      Sous fonctions : -
45  -----
46  -----
47      estVide
48
49      Rôle : retourner 1 si la pile est vide, 0 sinon
50      CONDITION NÉCESSAIRE : -
51      Entrée : pile : pointeur sur structure pile
52
53      Sortie : un entier :
54                - 1 si la pile est vide
55                - 0 sinon
56      Variable locales : -
57      Sous fonctions : -
58  -----
```

```

59 -----
60         empiler
61
62     Rôle : Ajouter une valeur au sommet de la pile
63     CONDITION NÉCESSAIRE : il doit y avoir de l'espace libre dans la pile
64
65     Entrée : pile      : pointeur sur structure pile
66             valeur    : valeur à ajouter à la pile
67     Sortie : pile avec l'élément ajouté au sommet
68
69     Variable locales : -
70     Sous fonctions : -
71 -----
72 -----
73         depiler
74         .....
75     Rôle : retourner l'élément au sommet de la pile et le supprimer
76     CONDITION NÉCESSAIRE : la pile doit être NON VIDE
77
78     Entrée : pile      : pointeur sur structure pile
79     Sortie : élément au sommet de la pile
80
81     Variable locales : valSommet (type élément) : valeur au sommet de
82                       la pile
83     Sous fonctions : -
84 -----
85 -----
86         sommet
87
88     Rôle : retourne l'élément au sommet de la pile
89     CONDITION NÉCESSAIRE : la pile doit être NON VIDE
90
91     Entrée : pile : pointeur sur structure pile
92     Sortie : valeur au sommet de la pile (type élément)
93
94     Variable locales : -
95     Sous-fonctions : -
96 -----
97 -----
98         afficheP
99         .....
100    Rôle : afficher la pile - espace disponible/occupé
101    CONDITION NÉCESSAIRE : -
102    Entrée : pile : pointeur sur structure pile
103    Sortie : -
104
105    Variable locales : i : indice de parcours des éléments de la pile
106                      (liste contiguë)
107    Sous fonction : -
108 -----

```

```

109 -----
110 libererP
111 -----
112 Rôle : libérer l'espace occupé par la structure pile
113 CONDITION NÉCESSAIRE : -
114
115 Entrée : pile : pointeur sur structure pile
116 Sortie : -
117 Variable locales : -
118 Sous fonctions : -
119 -----
120 -----
121 inversion
122 -----
123 Rôle : inverser les éléments d'une pile
124 CONDITION NÉCESSAIRE : -
125 Principe : 1. chaque valeur de la pile est dépilée et ajoutée à la file
126            2. chaque valeur de la file est supprimée et ajoutée à la pile
127
128 Entrée : pile : pointeur sur structure pile
129          codeErreur : 1 si erreur
130 Sortie : pile inversée (même pile)
131 Variable locales :
132       - file : structure file dans laquelle on va empiler tous les
133         éléments de la pile
134 Sous fonctions : - initFile
135                  - estVideP
136                  - enfiler
137                  - estVideF
138                  - empiler
139 -----
140 -----
141 genererPile
142 -----
143 Rôle : remplir une pile de i éléments allant de 0 à taille-1
144 CONDITION NÉCESSAIRE : taille > nombre d'éléments
145
146 Entrée : - taille : taille de la pile (nombre d'éléments max)
147          - nbElem : nombre d'élément
148 Sortie : - pile contenant taille nombre allant de 0 à taille-1 (si
149          allocation réussie)
150          - NULL sinon
151
152 Variable locales : i : indice courant allant de 0 à taille-1
153 Sous fonction : empiler
154                initPile
155 -----

```

```

159 *****
160 *                               FILE                               *
161 *****
162 -----
163         initFile
164         .....
165         Fichier : File.c
166         Rôle : initialise la structure file (retourne NULL si échec)
167         CONDITION NÉCESSAIRE : -
168
169         Entrée : taille : taille de la file (nombre max d'éléments)
170         Sortie : pointeur sur structure file si allocation OK
171                 NULL sinon
172                 .....
173         Variable locales : f : structure file
174         Sous fonctions : -
175         -----
176         -----
177         enfiler
178         .....
179         Fichier : File.c
180         Rôle : insérer un élément dans la file
181         Condition NÉCESSAIRE : la file ne doit pas être pleine
182
183         Entrée : file : pointeur sur structure file
184                 val : valeur à insérer
185         Sortie : pile contenant le nouvel élément
186
187         Variable locales : indice : indice où est inséré l'élément
188         Sous fonctions : -
189         -----
190         -----
191         defiler
192         .....
193         Fichier : File.c
194         Rôle : retourner l'élément au sommet de F et le supprime
195         CONDITION NÉCESSAIRE : file non vide
196
197         Entrée : file : pointeur sur structure file
198         Sortie : élément au sommet de la file
199                 suppression de cet élément dans la file
200
201         Variable locales : -
202         Sous fonctions : sommetF
203                 .....
204         -----
205         estVideF
206         .....
207         Fichier : File.c
208         Rôle : dire si la file est vide
209         CONDITION NÉCESSAIRE : -
210         Entrée : file : pointeur sur structure file
211         Sortie : entier : 1 si la file est vide
212                 0 sinon
213         Variable locales : -
214         Sous fonctions : -
215         -----

```

```

216 -----
217             sommetF
218             .....
219 Fichier : File.c
220 Rôle : retourner l'élément au sommet de F
221 CONDITION NÉCESSAIRE : file non vide
222
223 Entrée : file      : pointeur sur structure file
224 Sortie : élément au sommet de la file
225
226 Variable locales : -
227 Sous fonctions : -
228 -----
229 -----
230             afficherF
231             .....
232 Fichier : file.c
233 Rôle : afficher la structure file
234 CONDITION NÉCESSAIRE :
235
236 Entrée : file      : pointeur sur structure file
237 Sortie : -
238 Variable locales : i : indice de parcours de la file
239 Sous fonctions : -
240 -----
241 -----
242             libererF
243             .....
244 Fichier : File.c
245 Rôle : libérer l'espace occupé par la structure file
246 CONDITION NÉCESSAIRE : -
247
248 Entrée : file      : pointeur sur structure file
249 Sortie : -
250 Variable locales : -
251 Sous fonctions : -
252 -----
253
254
255 *****
256 *                               Erreur                               *
257 *****
258 -----
259             AfficheErreur
260             .....
261 Fichier : Erreur.c
262 Rôle : Afficher un message d'erreur lorsqu'une erreur se produit
263 CONDITION NÉCESSAIRE : -
264
265 Entrée : codeErreur : code de l'erreur
266 Sortie : -
267 Variable locales : -
268 Dépendances : -
269 -----

```



### III. Compte rendu des exécutions :

#### 1. Makefile

- Confère fichier makefile

#### 2. Liste des cas :

- Inversion pile :
  - Pile vide (inversion d'une pile de 0 élément)
  - Cas général : nombre d'éléments < taille pile
  - Pile pleine : nombre d'éléments = taille pile
  - Cas d'erreur : nombre d'éléments > taille pile
- Module pile :
  - Fonctions empiler-dépiler
  - Fonctions EstVide – sommetP
    - Cas pile vide
    - Cas pile pleine
    - Cas général
- Module file :
  - Fonction enfiler-defiler
    - File vide
    - Insertion-suppression normal (indice fin > indice début)
    - Insertion-suppression avec retour en début de liste contiguë (indice début > indice fin)
  - Fonction estVide :
    - Cas file vide
    - Cas file pleine
    - Cas général
  - Fonction sommetP :
    - Cas file pleine
    - Cas général

- Inversion pile : cas d'erreur : nombre d'éléments > taille pile

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$ valgrind ./prog 5 10
==2203== Memcheck, a memory error detector
==2203== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==2203== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==2203== Command: ./prog 5 10
==2203==
- Nombre d'éléments > taille
==2203==
==2203== HEAP SUMMARY:
==2203==   in use at exit: 0 bytes in 0 blocks
==2203==   total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==2203==
==2203== All heap blocks were freed -- no leaks are possible
==2203==
==2203== For counts of detected and suppressed errors, rerun with: -v
==2203== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$
```

- Inversion pile : pile vide

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$ valgrind ./prog 0 0
==2064== Memcheck, a memory error detector
==2064== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==2064== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==2064== Command: ./prog 0 0
==2064==
PILE :
- Nombre d'éléments : 0
- Nombre cases libres : 0
- Pile :
File :
- Nombre d'éléments : 0
- Nombre de cases libres : 0
- File :
PILE :
- Nombre d'éléments : 0
- Nombre cases libres : 0
- Pile :
==2064==
==2064== HEAP SUMMARY:
==2064==   in use at exit: 0 bytes in 0 blocks
==2064==   total heap usage: 5 allocs, 5 frees, 1,064 bytes allocated
==2064==
==2064== All heap blocks were freed -- no leaks are possible
==2064==
==2064== For counts of detected and suppressed errors, rerun with: -v
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$
```

- Inversion pile : pile pleine

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$ valgrind ./prog 5 5
==2109== Memcheck, a memory error detector
==2109== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==2109== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==2109== Command: ./prog 5 5
==2109==
FILE :
- Nombre d'éléments : 5
- Nombre cases libres : 0
- Pile :
    || 4.00 ||
    -----
    || 3.00 ||
    -----
    || 2.00 ||
    -----
    || 1.00 ||
    -----
    || 0.00 ||
    -----
File :
- Nombre d'éléments :5
- Nombre de cases libres : 0
- File :
    || 4.00 ||
    || 3.00 ||
    || 2.00 ||
    || 1.00 ||
    || 0.00 ||
PILE :
- Nombre d'éléments : 5
- Nombre cases libres : 0
- Pile :
    || 0.00 ||
    -----
    || 1.00 ||
    -----
    || 2.00 ||
    -----
    || 3.00 ||
    -----
    || 4.00 ||
    -----
==2109==
==2109== HEAP SUMMARY:
==2109==   in use at exit: 0 bytes in 0 blocks
==2109==   total heap usage: 5 allocs, 5 frees, 1,104 bytes allocated
==2109==
==2109== All heap blocks were freed -- no leaks are possible
==2109==
==2109== For counts of detected and suppressed errors, rerun with: -v
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$
```

- Inversion pile : cas général

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$ valgrind ./prog 10 5
==2116== Memcheck, a memory error detector
==2116== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==2116== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==2116== Command: ./prog 10 5
==2116==
PILE :
- Nombre d'éléments : 5
- Nombre cases libres : 5
- Pile :
    || 4.00 ||
    -----
    || 3.00 ||
    -----
    || 2.00 ||
    -----
    || 1.00 ||
    -----
    || 0.00 ||
    -----
File :
- Nombre d'éléments :5
- Nombre de cases libres : 5
- File :
    || 4.00 ||
    || 3.00 ||
    || 2.00 ||
    || 1.00 ||
    || 0.00 ||
PILE :
- Nombre d'éléments : 5
- Nombre cases libres : 5
- Pile :
    || 0.00 ||
    -----
    || 1.00 ||
    -----
    || 2.00 ||
    -----
    || 3.00 ||
    -----
    || 4.00 ||
    -----
==2116==
==2116== HEAP SUMMARY:
==2116==    in use at exit: 0 bytes in 0 blocks
==2116== total heap usage: 5 allocs, 5 frees, 1,144 bytes allocated
==2116==
==2116== All heap blocks were freed -- no leaks are possible
==2116==
==2116== For counts of detected and suppressed errors, rerun with: -v
==2116== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/TP2_Pile_File-master$
```

- Module pile : fonctions empiler-dépiler

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$ valgrind ./prog 5 5
==8966== Memcheck, a memory error detector
==8966== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==8966== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==8966== Command: ./prog 5 5
==8966==
CAS GENERAL : empiler

PILE :
- Nombre d'éléments : 0
- Nombre cases libres : 5
- Pile :
Element empilé : 0
Element empilé : 1
Element empilé : 2
Element empilé : 3
Element empilé : 4

PILE :
- Nombre d'éléments : 5
- Nombre cases libres : 0
- Pile :
    || 4.00 ||
    -----
    || 3.00 ||
    -----
    || 2.00 ||
    -----
    || 1.00 ||
    -----
    || 0.00 ||
    -----

CAS GENERAL : depiler
Element dépilé : 4.00
Element dépilé : 3.00
Element dépilé : 2.00
Element dépilé : 1.00

PILE :
- Nombre d'éléments : 1
- Nombre cases libres : 4
- Pile :
    || 0.00 ||
    -----

==8966==
==8966== HEAP SUMMARY:
==8966==   in use at exit: 0 bytes in 0 blocks
==8966== total heap usage: 3 allocs, 3 frees, 1,060 bytes allocated
==8966==
==8966== All heap blocks were freed -- no leaks are possible
==8966==
==8966== For counts of detected and suppressed errors, rerun with: -v
==8966== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$
```

- Module pile : fonctions EstVide – sommetP

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$ valgrind ./prog 2 2
==9287== Memcheck, a memory error detector
==9287== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==9287== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==9287== Command: ./prog 2 2
==9287==
CAS PILE VIDE :
PILE :
    - Nombre d'éléments : 0
    - Nombre cases libres : 2
    - Pile :
      sommetP -> -
      estVide -> 1

CAS GENERAL:
PILE :
    - Nombre d'éléments : 1
    - Nombre cases libres : 1
    - Pile :
      || 1.00 ||
      -----
      sommetP -> 1.00
      estVide -> 0

CAS PILE PLEINE
PILE :
    - Nombre d'éléments : 2
    - Nombre cases libres : 0
    - Pile :
      || 2.00 ||
      -----
      || 1.00 ||
      -----
      sommetP -> 2.00
      estVide -> 0

==9287==
==9287== HEAP SUMMARY:
==9287==      in use at exit: 0 bytes in 0 blocks
==9287==    total heap usage: 3 allocs, 3 frees, 1,048 bytes allocated
==9287==
==9287== All heap blocks were freed -- no leaks are possible
==9287==
==9287== For counts of detected and suppressed errors, rerun with: -v
==9287== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$
```

- Module file : fonction enfiler-defiler

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$ valgrind ./prog 10 10
==6753== Memcheck, a memory error detector
==6753== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==6753== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==6753== Command: ./prog 10 10
==6753==
File :
- Nombre d'éléments : 10
- Nombre de cases libres : 0
- indice début : 0
- indice fin : 9
    || 0.00 ||
    || 1.00 ||
    || 2.00 ||
    || 3.00 ||
    || 4.00 ||
    || 5.00 ||
    || 6.00 ||
    || 7.00 ||
    || 8.00 ||
    || 9.00 ||

Suppression des 5 premiers éléments
File :
- Nombre d'éléments : 5
- Nombre de cases libres : 5
- indice début : 5
- indice fin : 9
    || 5.00 ||
    || 6.00 ||
    || 7.00 ||
    || 8.00 ||
    || 9.00 ||

Ajout de 5 nouvelles valeurs
File :
- Nombre d'éléments : 10
- Nombre de cases libres : 0
- indice début : 5
- indice fin : 4
    || 5.00 ||
    || 6.00 ||
    || 7.00 ||
    || 8.00 ||
    || 9.00 ||
    || 10.00 ||
    || 11.00 ||
    || 12.00 ||
    || 13.00 ||
    || 14.00 ||

Suppression de 7 éléments
File :
- Nombre d'éléments : 3
- Nombre de cases libres : 7

Ajout de 5 nouvelles valeurs
File :
- Nombre d'éléments : 10
- Nombre de cases libres : 0
- indice début : 5
- indice fin : 4
    || 5.00 ||
    || 6.00 ||
    || 7.00 ||
    || 8.00 ||
    || 9.00 ||
    || 10.00 ||
    || 11.00 ||
    || 12.00 ||
    || 13.00 ||
    || 14.00 ||

Suppression de 7 éléments
File :
- Nombre d'éléments : 3
- Nombre de cases libres : 7
- indice début : 2
- indice fin : 4
    || 12.00 ||
    || 13.00 ||
    || 14.00 ||

==6753==
==6753== HEAP SUMMARY:
==6753==   in use at exit: 0 bytes in 0 blocks
==6753==   total heap usage: 3 allocs, 3 frees, 1,088 bytes allocated
==6753==
==6753== All heap blocks were freed -- no leaks are possible
==6753==
==6753== For counts of detected and suppressed errors, rerun with: -v
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$
```

- Module file : fonction estVide :

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$ valgrind ./prog 2 2
==7520== Memcheck, a memory error detector
==7520== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==7520== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==7520== Command: ./prog 2 2
==7520==

CAS FILE VIDE :
File :
  - Nombre d'éléments : 0
  - Nombre de cases libres : 2
  - indice début : 0
  - indice fin : -1
  Test fonction : estVideF -> 1

CAS GENERAL
File :
  - Nombre d'éléments : 1
  - Nombre de cases libres : 1
  - indice début : 0
  - indice fin : 0
  || 8.00 ||
  Test fonction : estVideF -> 0

CAS FILE PLEINE :
File :
  - Nombre d'éléments : 2
  - Nombre de cases libres : 0
  - indice début : 0
  - indice fin : 1
  || 8.00 ||
  || 6.00 ||
  Test fonction : estVideF -> 0

==7520==
==7520== HEAP SUMMARY:
==7520==   in use at exit: 0 bytes in 0 blocks
==7520==   total heap usage: 3 allocs, 3 frees, 1,056 bytes allocated
==7520==
==7520== All heap blocks were freed -- no leaks are possible
==7520==
==7520== For counts of detected and suppressed errors, rerun with: -v
==7520== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$
```

- Module file : fonction sommetP :

```
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$ valgrind ./prog 5 5
==7636== Memcheck, a memory error detector
==7636== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==7636== Using Valgrind-3.12.0.SVN and LibVEX; rerun with -h for copyright info
==7636== Command: ./prog 5 5
==7636==

File :
  - Nombre d'éléments : 1
  - Nombre de cases libres : 4
  - indice début : 0
  - indice fin : 0
  || 0.00 ||

CAS GENERAL
  Sommet : 0.00
File :
  - Nombre d'éléments : 5
  - Nombre de cases libres : 0
  - indice début : 0
  - indice fin : 4
  || 0.00 ||
  || 1.00 ||
  || 2.00 ||
  || 3.00 ||
  || 4.00 ||

CAS FILE PLEINE
  Sommet : 0.00
==7636==
==7636== HEAP SUMMARY:
==7636==   in use at exit: 0 bytes in 0 blocks
==7636==   total heap usage: 3 allocs, 3 frees, 1,068 bytes allocated
==7636==
==7636== All heap blocks were freed -- no leaks are possible
==7636==
==7636== For counts of detected and suppressed errors, rerun with: -v
==7636== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
alexandre@ALEX-PC:/media/alexandre/Mes documents/1.Cours/ISIMA/SDD/TP2/Capture_ecran/Capt$
```