

TD 1

Les fonctions

Programmation C
Semestre 2

1 Exercice 1

1.1 Tri par insertion d'une suite de valeurs

```
1  #define LG_MAX 100
2
3  typedef enum {FAUX=0, VRAI=1} Booleen;
4
5  typedef int TabEntiers[LG_MAX];
6  void rechercherOccurrence(const TabEntiers tab,
7                           const int n,
8                           const int ele,
9                           Booleen * trouve,
10                          int * rang);
```

1.2

```
1  void rechercherOccurrence(const TabEntiers tab, const int n, const int ele,
2                           Booleen * trouve, int * rang)
3  {
4      Booleen fini ;
5      int i ;
6
7      i = 0 ;
8      fini = FAUX ;
9
10     while (! fini)
11     {
12         if (i < n)
13         {
14             if (ele == tab[i])
15             {
16                 *trouve = VRAI ;
17                 *rang = i ;
18                 fini = VRAI ;
19             }
20             else
21             {
22                 i++ ;
23             }
24         }
25     }
```

```

26         * trouve = FAUX ;
27         fini = VRAI ;
28     }
29 }
30 }

```

```

1  rechercherOccurrence(listeNotes, nbNotes, noteZero, &trouve, &rangNote);

```

1.3

```

1  int main(int argc, char *argv[])
2  {
3      TabEntiers listeNotes ;
4      int nbreNotes ;
5      int noteZero = 0 ;
6      int rangNote ;
7      Booleen trouve ;
8
9      .....
10     /* rechercher la note 0 */
11     rechercherOccurrence(listeNotes, nbreNotes, noteZero, &trouve, &rangNote);
12     .....
13 }

```

2 Exercice 2

2.1

```

1  typedef enum {FAUX=0, VRAI=1} Booleen ;
2  typedef int * PtrEntier ;
3
4  void rechercherSousTableau (const PtrEntier adeb,
5                             const PtrEntier afin,
6                             const int ele,
7                             Booleen * trouve,
8                             PtrEntier * arang);

```

2.2

```

1  void rechercherSousTableau (const PtrEntier adeb, const PtrEntier afin,
2                             const int ele, Booleen * trouve, PtrEntier * arang)
3  {
4      Booleen fini ;
5      PtrEntier acour ;
6
7      acour = adeb ;
8      fini = FAUX ;
9
10     while (! fini)
11     {
12         if (acour <= afin)
13         {
14             if (ele == * acour)
15             {

```

```
15         *trouve = VRAI ;
16         *arang = acour ;
17         fini = VRAI ;
18     }
19     else
20     {
21         acour++ ;
22     }
23 }
24 else
25 {
26     *trouve = FAUX ;
27     fini = VRAI ;
28 }
29 }
30 }
```

2.3

```
1  int main(int argc, char *argv[])
2  {
3      PtrEntier premiereNote ;
4      PtrEntier derniereNote ;
5      int noteZero = 0 ;
6      Booleen trouve ;
7      PtrEntier refNote ;
8      .....
9      /* rechercher la note 0 */
10     rechercherSousTableau(premiereNote, derniereNote, noteZero, &trouve, &
        refNote);
11     .....
12 }
```