

TD 1

Les fonctions

Programmation C
Semestre 2

1 Exercice 1

1.1 Tri par insertion d'une suite de valeurs

```
1 #define LG 100
```

1.2

```
1 typedef int TabEntiers[LG];
2
3 void rechercherOccurrence (const TabEntiers tab,
4                             const int n,
5                             const int ele,
6                             Booleen *trouve,
7                             int *rang );
```

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

1.3

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

```
22         else
23         {
24             i++;
25         }
26     else
27     {
28         *trouve = FAUX;
29         fini = VRAI;
30     }
31 }
```

2 Exercice 2

2.1

```
1  typedef int * PtrEntier;
```

2.2

```
1  //Avec ptrEntier
2  void rechercherSousTableau(const PtrEntier adeb,
3                           const PtrEntier afin,
4                           const int ele,
5                           Booleen *trouve,
6                           PtrEntier *arang);
7
8  //Sans ptrEntier
9  void rechercherSousTableau(const int *adeb,
10                           const int *afin,
11                           const int ele,
12                           Booleen *trouve,
13                           int **arang);
```

2.3

```
1  void rechercherSousTableau(const PtrEntier adeb,
2                           const PtrEntier afin,
3                           const int ele,
4                           Booleen *trouve,
5                           PtrEntier *arang)
6  {
7      Booleen fini;
8      ptrEntier aCour;
9      acour = adeb;
10     fini = FAUX;
11
12     while (!fini)
13     {
14         if (acour <= afin)
15         {
16             if (ele == *acour)
17             {
18                 *trouve = VRAI;
19                 *arang = acour;
```

```
20         fini = FAUX;
21     }
22     else
23     {
24         acour = acour + sizeof(int);
25     }
26 }
27 else
28 {
29     *trouve = FAUX;
30     fini = VRAI;
31 }
32 }
33 }
```