



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

TP7

Made by :
Lilian Naretto

```
1 #include "tp7.h"
2 /*!\author Lilian Naretto <narettolil@eisti1.eu>
3 \date 15 novembre 2019
4 \file tp7.c
5 \brief tp7
6 \version 0.1 premier jet*/
7
8 void tri_insertion(int tableau[N]){
9     int i, j;
10    int valeur;
11    for (i = 1; i < N; i++) {
12        valeur = tableau[i];
13        for (j = i; j > 0 && tableau[j - 1] > valeur; j--) {
14            tableau[j]=tableau[j-1];
15        }
16        tableau[j] = valeur;
17    }
18 }
19
20 int* copiersoustableau(int* src, int debut, int fin){
21     int taille=fin-debut+1;
22     int* dest=malloc(taille*sizeof(int));
23     for (int i = 0; i <= fin - debut + 1; ++i)
24     {
25         dest[i] = src[debut+i];
26     }
27     return dest;
28 }
29
30 void fusion(int* tabl, int taille1, int* tab2,int taille2, int* tabRes){
31     int taille=taille1+taille2;
32     int k=0;
33     int debut1=0;
34     int debut2=0;
35     for (int i = 0; i < taille; i++)
36     {
37         if (debut2>=taille2)
38         {
39             tabRes[k++]=tabl[debut1++];
40         }
41         else if (debut1>=taille1)
42         {
43             tabRes[k++]=tab2[debut2++];
44         }
45         else{
46             if (tabl[debut1]<=tab2[debut2])
47             {
48                 tabRes[k++]=tabl[debut1++];
49             }
50             else
51             {
52                 tabRes[k++]=tab2[debut2++];
53             }
54         }
55     }
56 }
57 }
```

```
58
59 void triFusion(int* tab, int taille){
60     int taille2 = taille / 2;
61     int taille3 = taille - taille2;
62     if (taille >= 2)
63     {
64         int *tab1=copiersoustableau(tab, 0, taille3-1);
65         int *tab2=copiersoustableau(tab, taille3, taille-1);
66         triFusion(tab1, taille3);
67         triFusion(tab2, taille2);
68         fusion(tab1, taille3, tab2, taille2, tab);
69     }
70 }
71 }
72
73 void minMaxTableau(int* tab, int taille, int* min,int* max){
74     *min = tab[0];
75     *max = tab[0];
76     for (int i = 1; i < taille; ++i)
77     {
78         if (tab[i] > *max)
79         {
80             *max = tab[i];
81         }else if (tab[i] < *min)
82         {
83             *min = tab[i];
84         }
85     }
86 }
87
88 void histogramme(int* tab, int taille, int* histo, int tailleH, int min){
89     for (int i = 0; i < taille; ++i)
90     {
91         histo[tab[i]-min]++;
92     }
93 }
94
95 void triDenombrement(int* tab, int taille){
96     int* min=malloc(1*sizeof(int));
97     int* max=malloc(1*sizeof(int));
98     minMaxTableau(tab, taille, min, max);
99     int tailleH = *max - *min +1;
100     int histo[tailleH] ;
101     for (int i = 0; i < tailleH; ++i)
102     {
103         histo[i] = 0;
104     }
105     histogramme(tab, taille, histo, tailleH, *min);
106     int l = 0;
107     for (int i = 0; i < tailleH; ++i)
108     {
109         for (int v = 0; v < histo[i]; ++v)
110         {
111             tab[l] = i+*min;
112             l++;
113         }
114     }
115 }
```

```
115 }
116 /*! \fn int main(int argc, char** argv)
117 \param argc nombre d'arguments en entrée
118 \param argv valeur des arguments en entrée
119 \brief sert de menu pour choisir les tries voulus*/
120 int main(int argc, char** argv){
121     int i,question;
122     int k[N],b[N];
123     printf("k[N] = [");
124     for ( i = 0; i < N; i++)
125     {
126         k[i]=rand()%20;
127         printf("%d,",k[i]);
128     }
129     printf("]\n");
130     printf("b[N] = [");
131     for ( i = 0; i < N; i++)
132     {
133         b[i]=rand()%20;
134         printf("%d,",b[i]);
135     }
136     printf("]\n");
137     printf("tri insertion = 1\n");
138     printf("tri fusion = 2\n");
139     printf("tri dénombrement = 3\n");
140     printf("choisis le type de tri : ");
141     scanf("%d",&question);
142     if (question > 3 || question < 0)
143     {
144         printf("mauvais numero\n");
145     }
146     else
147     {
148         switch (question)
149         {
150             case 1:
151                 tri_insertion(k);
152                 printf("k[N] = [");
153                 for ( i = 0; i < N; i++)
154                 {
155                     printf("%d,",k[i]);
156                 }
157                 printf("]\n");
158                 return 0;
159             break;
160             case 2:
161                 triFusion(b,N);
162                 printf("b[N] = [");
163                 for ( i = 0; i < N; i++)
164                 {
165                     printf("%d,",b[i]);
166                 }
167                 printf("]\n");
168                 return 0;
169             break;
```