



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

TP9

Made by :
Lilian Naretto

```
1  #include "tp9.h"
2  /*!\author Lilian Naretto <narettolil@eisti.eu>
3  \date 19 novembre 2019
4  \file tp9.c
5  \brief tp9
6  \version 0.1 premier jet*/
7
8  void creerg grille(int grille[hauteur][largeur]){
9      int x;
10     for (int i = 0; i < hauteur; i++)
11     {
12         for (int v = 0; v < largeur; v++)
13         {
14             printf("valeur case %d - %d : ",i,v);
15             scanf("%d",&x);
16             while (x!=0 && x!=1)
17             {
18                 printf("mauvais numero, recommence (1 ou 0)\n");
19                 printf("valeur case %d - %d : ",i,v);
20                 scanf("%d",&x);
21             }
22             grille[i][v]=x;
23         }
24     }
25 }
26
27
28 }
29
30 void affichergrille(int grille[hauteur][largeur]){
31     for (int i = 0; i < hauteur; i++)
32     {
33         for (int v = 0; v < largeur; v++)
34         {
35             if (grille[i][v]==1)
36             {
37                 printf("V ");
38             }
39             else
40             {
41                 printf(". ");
42             }
43         }
44         printf("\n");
45     }
46 }
47
48 }
49
50 int voisinnord(int grille[hauteur][largeur],int x, int y){
51     if (x==0)
52     {
53         return grille[hauteur-1][y];
54     } else
55     {
56         return grille[x-1][y];
57     }
58 }
59 }
```

```
60
61 int voisinnorddest(int grille[hauteur][largeur],int x, int y){
62     if (x==0)
63     {
64         if (y==largeur-1)
65         {
66             return grille[hauteur-1][0];
67         }
68         else
69         {
70             return grille[hauteur-1][y+1];
71         }
72     } else
73     {
74         if (y==largeur-1)
75         {
76             return grille[x-1][0];
77         }
78         return grille[x-1][y+1];
79     }
80 }
81
82
83
84
85 int voisinest(int grille[hauteur][largeur],int x, int y){
86     if (y==largeur-1)
87     {
88         return grille[x][0];
89     } else
90     {
91         return grille[x][y+1];
92     }
93 }
94
95
96 int voisinsudest(int grille[hauteur][largeur],int x, int y){
97     if (x==hauteur-1)
98     {
99         if (y==largeur-1)
100         {
101             return grille[0][0];
102         }
103         else
104         {
105             return grille[0][y+1];
106         }
107     } else
108     {
109         if (y==largeur-1)
110         {
111             return grille[x+1][0];
112         }
113         return grille[x+1][y+1];
114     }
115 }
116
117
```

```
120 int voisinsud(int grille[hauteur][largeur],int x, int y){
121     if (x==hauteur-1)
122     {
123         return grille[0][y];
124     } else
125     {
126         return grille[x+1][y];
127     }
128 }
129 }
130
131 int voisinsudouest(int grille[hauteur][largeur],int x, int y){
132     if (x==hauteur-1)
133     {
134         if (y==0)
135         {
136             return grille[0][largeur-1];
137         }
138         else
139         {
140             return grille[0][y-1];
141         }
142     } else
143     {
144         if (y==0)
145         {
146             return grille[x+1][largeur-1];
147         }
148         return grille[x+1][y-1];
149     }
150 }
151 }
152 }
153 }
154
155 int voisinouest(int grille[hauteur][largeur],int x, int y){
156     if (y==0)
157     {
158         return grille[x][largeur-1];
159     } else
160     {
161         return grille[x][y-1];
162     }
163 }
164 }
165
166 int voisinnordouest(int grille[hauteur][largeur],int x, int y){
167     if (x==0)
168     {
169         if (y==0)
170         {
171             return grille[hauteur-1][largeur-1];
172         }
173         else
174         {
```

```
174         {
175             return grille[hauteur-1][y-1];
176         }
177     } else
178     {
179         if (y==0)
180         {
181             return grille[x-1][largeur-1];
182         }
183     }
184     return grille[x-1][y-1];
185 }
186 }
187
188 }
189
190 int comptecellulevoisinage(int grille[hauteur][largeur],int x, int y){
191     int k=0;
192     if (voisinnord(grille,x,y)==1)
193     {
194         k++;
195     }
196     if (voisinnorddest(grille,x,y)==1)
197     {
198         k++;
199     }
200     if (voisinnordouest(grille,x,y)==1)
201     {
202         k++;
203     }
204     if (voisinsud(grille,x,y)==1)
205     {
206         k++;
207     }
208     if (voisinsudest(grille,x,y)==1)
209     {
210         k++;
211     }
212     if (voisinsudouest(grille,x,y)==1)
213     {
214         k++;
215     }
216     if (voisinest(grille,x,y)==1)
217     {
218         k++;
219     }
220     if (voisinouest(grille,x,y)==1)
221     {
222         k++;
223     }
224 }
225 return k;
226 }
227 }
```

```
227 }
228
229 int calculernouvellevaleur(int grille[hauteur][largeur],int x, int y){
230     if (grille[x][y]==1 && (comptecellulevoisinage(grille,x,y)==2 ||comptecellulevoisinage(grille,x,y)==3))
231     {
232         return 1;
233     }
234     else
235     {
236         if (comptecellulevoisinage(grille,x,y)==3)
237         {
238             return 1;
239         }
240         else
241         {
242             return 0;
243         }
244     }
245 }
246
247
248 }
249
250 void calculernouvellegrille(int grille[hauteur][largeur]){
251     int grille2[hauteur][largeur];
252     for (int i = 0; i < hauteur; i++)
253     {
254         for (int v = 0; v < largeur; v++)
255         {
256             grille2[i][v]=calculernouvellevaleur(grille,i,v);
257         }
258     }
259     for (int i = 0; i < hauteur; i++)
260     {
261         for (int v = 0; v < largeur; v++)
262         {
263             grille[i][v]=grille2[i][v];
264         }
265     }
266 }
267
268 }
269
270
271 void jeudelavie(int grille[hauteur][largeur],int n){
272     for (int i = 0; i < n; i++)
273     {
274         calculernouvellegrille(grille);
275         affichergrille(grille);
276         printf("\n");
277     }
278 }
279
280 }
```

```
280
281
282 /*! \fn int main(int argc, char** argv)
283 \param argc nombre d'arguments en entrée
284 \param argv valeur des arguments en entrée
285 \brief lance le jeu*/
286 int main(int argc, char** argv){
287     int l;
288     int grille[hauteur][largeur];
289     creergrille(grille);
290     printf("\n");
291     affichergrille(grille);
292     printf("nombre d'étapes du jeu de la vie : ");
293     scanf("%d",&l);
294     printf("\n");
295     jeudelavie(grille,l);
296 }
```