



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

**TP10**

---

Made by :  
Lilian Naretto

```
1 #include "tp.h"
2 /*!\author Lilian Naretto <narettolil@eisti.eu>
3 \date 5 decembre 2019
4 \file tp.c
5 \brief tp
6 \version 0.1 premier jet*/
7
8 typedef struct nombreComplexe
9 {
10     double reel;
11     double imaginaire;
12 }complexe;
13
14 typedef struct solutionEqu2D
15 {
16     double a;
17     double b;
18     double c;
19     double d;
20     double racine1;
21     double racine2;
22     double racine3;
23     double racine4;
24     double racine5;
25     double racine6;
26     double discriminant;
27 }solution;
28
29 void resolution(double a,double b,double c){
30     solution resultat;
31     complexe nb;
32     resultat.discriminant = b * b - 4 * a * c;
33
34     if (resultat.discriminant > 0) {
35         resultat.racine1 = (-b + sqrt(resultat.discriminant)) / (2 * a);
36         resultat.racine2 = (-b - sqrt(resultat.discriminant)) / (2 * a);
37         printf("racine1 = %lf et racine2 = %lf \n", resultat.racine1, resultat.racine2);
38     }
39
40     else if (resultat.discriminant == 0) {
41         resultat.racine1 = resultat.racine2 = -b / (2 * a);
42         printf("racine1 = racine2 = %lf \n", resultat.racine1);
43     }
44     else {
45         nb.reel = -b / (2 * a);
46         nb.imaginaire = sqrt(-resultat.discriminant) / (2 * a);
47         printf("racine1 = %lf+%lf et racine2 = %lf-%lf \n", nb.reel, nb.imaginaire, nb.reel, nb.imaginaire);
48     }
49 }
```

```
48 }
49 }
50
51 void afficher(){
52     solution resultat;
53     printf("donne a : ");
54     scanf("%lf",&resultat.a);
55     printf("donne b : ");
56     scanf("%lf",&resultat.b);
57     printf("donne c : ");
58     scanf("%lf",&resultat.c);
59     resolution(resultat.a,resultat.b,resultat.c);
60 }
61
62 void approcher(double a, double b,double c, double d){
63     solution resultat;
64     if (a == 0)
65     {
66         printf("Erreur \n");
67         return;
68     }
69     if (d == 0)
70     {
71         printf("Erreur \n");
72         return;
73     }
74     b /= a;
75     c /= a;
76     d /= a;
77     double disc, q, r, dum1, s, t, term1, r13;
78     q = (3.0*c - (b*b))/9.0;
79     r = -(27.0*d) + b*(9.0*c - 2.0*(b*b));
80     r /= 54.0;
81     disc = q*q*q + r*r;
82     term1 = (b/3.0);
83     resultat.racine1=0;
84     if (disc > 0) {
85         s = r + sqrt(disc);
86         s = ((s < 0) ? -pow(-s, (1.0/3.0)) : pow(s, (1.0/3.0)));
87         t = r - sqrt(disc);
88         t = ((t < 0) ? -pow(-t, (1.0/3.0)) : pow(t, (1.0/3.0)));
89         resultat.racine4 = -term1 + s + t;
90         term1 += (s + t)/2.0;
91         resultat.racine5 = resultat.racine6 = -term1;
92         term1 = sqrt(3.0)*(-t + s)/2;
93         resultat.racine1 = term1;
94         resultat.racine3 = -term1;
95         resultat.racine2 = 0.0000;
```

```
96     printf("( %f + %f i , %f + %f i , %f + %f i )\n", resultat.racine4, resultat.racine2, resultat.racine5, resultat.racine1, resultat.racine6, resultat.racine3);
97     return;
98 }
99
100 resultat.racine3 = resultat.racine2 = 0;
101 if (disc == 0){
102     r13 = ((r < 0) ? -pow(-r, (1.0/3.0)) : pow(r, (1.0/3.0)));
103     resultat.racine1 = -term1 + 2.0*r13;
104     resultat.racine3 = resultat.racine2 = -(r13 + term1);
105     printf("( %f , %f , %f )\n", resultat.racine1, resultat.racine2, resultat.racine3);
106     return;
107 }
108 q = -q;
109 dum1 = q*q*q;
110 dum1 = acos(r/sqrt(dum1));
111 r13 = 2.0*sqrt(q);
112 resultat.racine1 = -term1 + r13*cos(dum1/3.0);
113 resultat.racine2 = -term1 + r13*cos((dum1 + 2.0*PI)/3.0);
114 resultat.racine3 = -term1 + r13*cos((dum1 + 4.0*PI)/3.0);
115 printf("( %f , %f , %f )\n", resultat.racine1, resultat.racine2, resultat.racine3);
116 return;
117 }
118 }
119
120 void afficher2(){
121     solution resultat;
122     printf("donne a : ");
123     scanf("%lf", &resultat.a);
124     printf("donne b : ");
125     scanf("%lf", &resultat.b);
126     printf("donne c : ");
127     scanf("%lf", &resultat.c);
128     printf("donne d : ");
129     scanf("%lf", &resultat.d);
130     approcher(resultat.a, resultat.b, resultat.c, resultat.d);
131 }
132
133 /*! \fn int main(int argc, char** argv)
134 \param argc nombre d'arguments en entrée
135 \param argv valeur des arguments en entrée
136 \brief lance les calculs*/
137 int main(int argc, char** argv){
138     printf("SECOND DEGREE \n");
139     afficher();
140     printf("TROISIEME DEGREE \n");
```