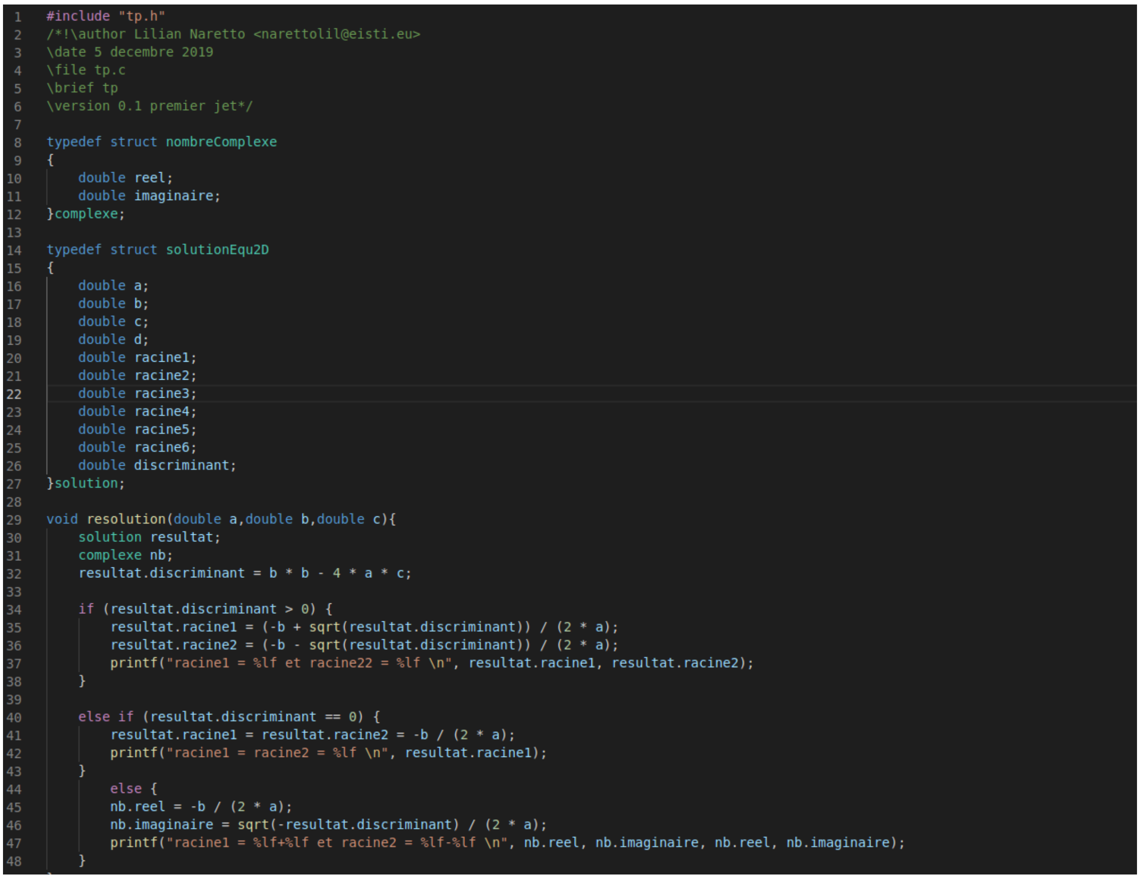
PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

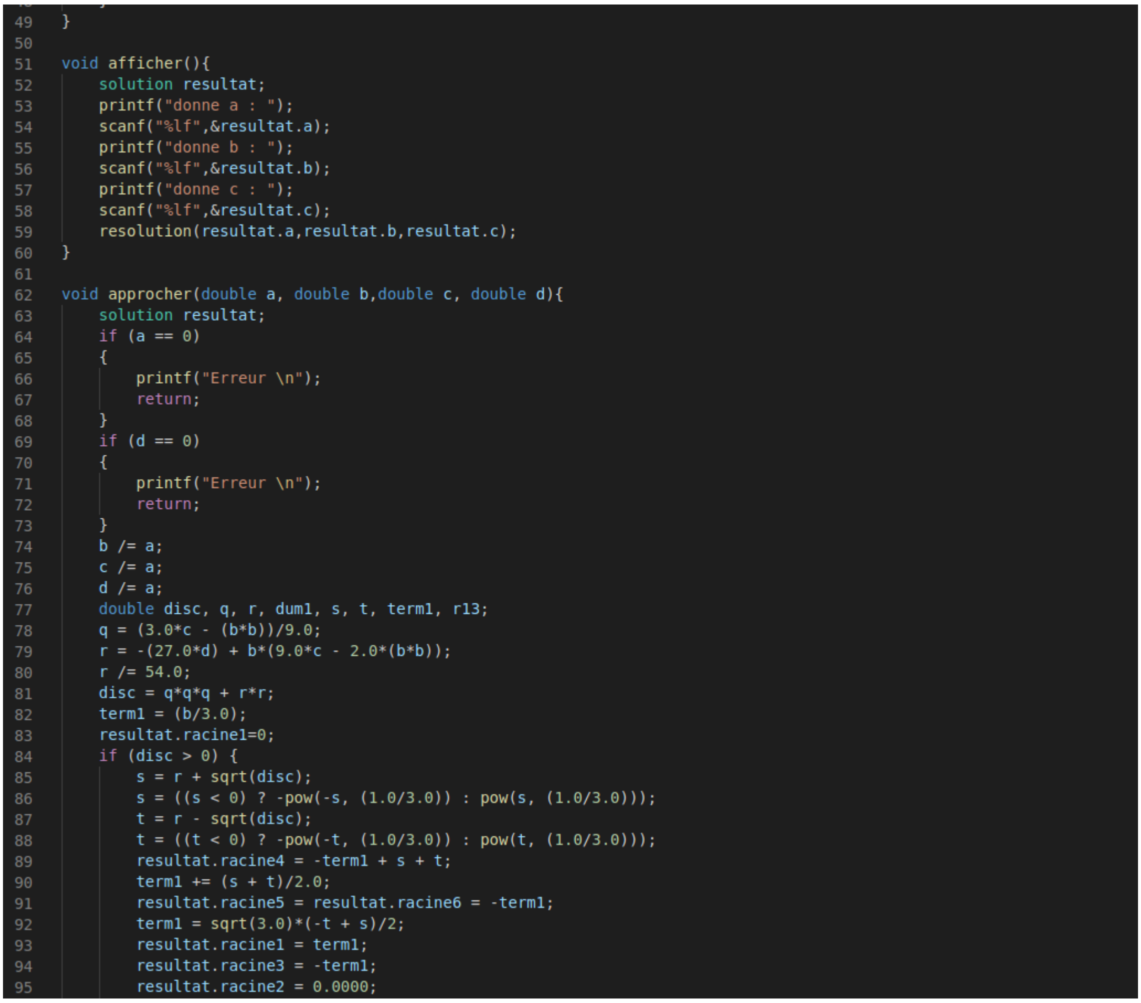
INTRODUCTION AU C :

TP10

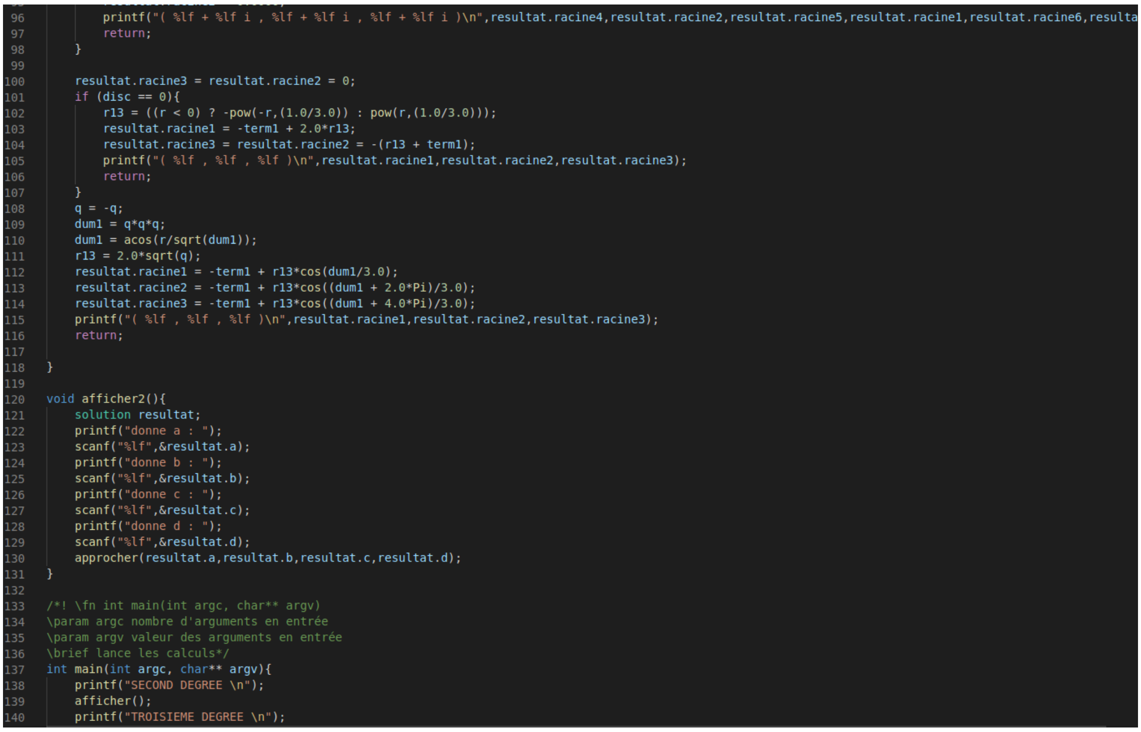
Made by :
Lilian Naretto



```c
#include "tp.h"
/*!\author Lilian Naretto <narettolil@eisti.eu>
\date 5 decembre 2019
\file tp.c
\brief tp
\version 0.1 premier jet*/

typedef struct nombreComplexe
{
    double reel;
    double imaginaire;
}complexe;

typedef struct solutionEqu2D
{
    double a;
    double b;
    double c;
    double d;
    double racine1;
    double racine2;
    double racine3;
    double racine4;
    double racine5;
    double racine6;
    double discriminant;
}solution;

void resolution(double a,double b,double c){
    solution resultat;
    complexe nb;
    resultat.discriminant = b * b - 4 * a * c;

    if (resultat.discriminant > 0) {
        resultat.racine1 = (-b + sqrt(resultat.discriminant)) / (2 * a);
        resultat.racine2 = (-b - sqrt(resultat.discriminant)) / (2 * a);
        printf("racine1 = %lf et racine22 = %lf \n", resultat.racine1, resultat.racine2);
    }

    else if (resultat.discriminant == 0) {
        resultat.racine1 = resultat.racine2 = -b / (2 * a);
        printf("racine1 = racine2 = %lf \n", resultat.racine1);
    }
        else {
        nb.reel = -b / (2 * a);
        nb.imaginaire = sqrt(-resultat.discriminant) / (2 * a);
        printf("racine1 = %lf+%lf et racine2 = %lf-%lf \n", nb.reel, nb.imaginaire, nb.reel, nb.imaginaire);
    }
```



```c
}

void afficher(){
    solution resultat;
    printf("donne a : ");
    scanf("%lf",&resultat.a);
    printf("donne b : ");
    scanf("%lf",&resultat.b);
    printf("donne c : ");
    scanf("%lf",&resultat.c);
    resolution(resultat.a,resultat.b,resultat.c);
}

void approcher(double a, double b,double c, double d){
    solution resultat;
    if (a == 0)
    {
        printf("Erreur \n");
        return;
    }
    if (d == 0)
    {
        printf("Erreur \n");
        return;
    }
    b /= a;
    c /= a;
    d /= a;
    double disc, q, r, dum1, s, t, term1, r13;
    q = (3.0*c - (b*b))/9.0;
    r = -(27.0*d) + b*(9.0*c - 2.0*(b*b));
    r /= 54.0;
    disc = q*q*q + r*r;
    term1 = (b/3.0);
    resultat.racine1=0;
    if (disc > 0) {
        s = r + sqrt(disc);
        s = ((s < 0) ? -pow(-s, (1.0/3.0)) : pow(s, (1.0/3.0)));
        t = r - sqrt(disc);
        t = ((t < 0) ? -pow(-t, (1.0/3.0)) : pow(t, (1.0/3.0)));
        resultat.racine4 = -term1 + s + t;
        term1 += (s + t)/2.0;
        resultat.racine5 = resultat.racine6 = -term1;
        term1 = sqrt(3.0)*(-t + s)/2;
        resultat.racine1 = term1;
        resultat.racine3 = -term1;
        resultat.racine2 = 0.0000;
```



```c
        printf("( %lf + %lf i , %lf + %lf i , %lf + %lf i )\n",resultat.racine4,resultat.racine2,resultat.racine5,resultat.racine1,resultat.racine6,resulta
        return;
    }

    resultat.racine3 = resultat.racine2 = 0;
    if (disc == 0){
        r13 = ((r < 0) ? -pow(-r,(1.0/3.0)) : pow(r,(1.0/3.0)));
        resultat.racine1 = -term1 + 2.0*r13;
        resultat.racine3 = resultat.racine2 = -(r13 + term1);
        printf("( %lf , %lf , %lf )\n",resultat.racine1,resultat.racine2,resultat.racine3);
        return;
    }
    q = -q;
    dum1 = q*q*q;
    dum1 = acos(r/sqrt(dum1));
    r13 = 2.0*sqrt(q);
    resultat.racine1 = -term1 + r13*cos(dum1/3.0);
    resultat.racine2 = -term1 + r13*cos((dum1 + 2.0*Pi)/3.0);
    resultat.racine3 = -term1 + r13*cos((dum1 + 4.0*Pi)/3.0);
    printf("( %lf , %lf , %lf )\n",resultat.racine1,resultat.racine2,resultat.racine3);
    return;

}

void afficher2(){
    solution resultat;
    printf("donne a : ");
    scanf("%lf",&resultat.a);
    printf("donne b : ");
    scanf("%lf",&resultat.b);
    printf("donne c : ");
    scanf("%lf",&resultat.c);
    printf("donne d : ");
    scanf("%lf",&resultat.d);
    approcher(resultat.a,resultat.b,resultat.c,resultat.d);
}

/*! \fn int main(int argc, char** argv)
\param argc nombre d'arguments en entrée
\param argv valeur des arguments en entrée
\brief lance les calculs*/
int main(int argc, char** argv){
    printf("SECOND DEGREE \n");
    afficher();
    printf("TROISIEME DEGREE \n");
```