#include<iostream>

#include<cmath>

#include<complex>

#include<fstream>

using namespace std;

const float pi = 3.1415926535;

void welcome();

void start();

float argcosh();

float argsinh();

class filter

{

float AMAX, AMIN, FC, FS;

int ORD;

public:

filter();

bool test();

void ordre();

void poles();

void module();

};

void main()

{

welcome();

while (true)

{

start();

filter F;

if (F.test()==0)

{

F.ordre();

F.poles();

F.module();

}

}

}

void welcome()

{

cout << "Bonjour," << endl << "Ce programme vous permet de determiner l'ordre minimale d'un filtre

de CHEBYCHEV pass bas, anisi ses poles, comme il sovgarde le module de votre filtre dans un

fichier." << endl;

cout << "..........................................................................." << endl<<endl;

}

void start()

{

cout << "Tout d'abord entrez les informations du gabarit en question:" << endl;

}

filter::filter()

{

cout << "Amax (en dB) = ";

cin >> AMAX;

cout << "Amin (en dB) = ";

cin >> AMIN;

cout << "fc (en KHz) = ";

cin >> FC;

cout << "fs (en KHz) = ";

cin >> FS;

cout << "..........................................................................." << endl<<endl;

}

bool filter::test()

{

if (AMAX > AMIN || FC > FS)

{

cout << "Il y a un erreur dans votre informations... Reessayez..." << endl;

cout << "..........................................................................." << endl;

return true;

}

else

return false;

}

float argcosh(float x)

{

float s;

s = log(x + sqrt(x\*x - 1));

return s;

}

float argsinh(float x)

{

float s;

s = log(x + sqrt(x\*x + 1));

return s;

}

void filter::ordre()

{

float N, X1, X2;

X1 = sqrt((pow(10, AMIN / 10) - 1) / ((pow(10, AMAX / 10)) - 1));

X2 = (FS / FC);

N = argcosh(X1) / argcosh(X2);

if (N - int(N) == 0)

ORD = N;

else

ORD = N + 1;

cout << "l'ordre minimale de votre filtre est N = " << ORD << endl;

cout << "..........................................................................." << endl<<endl;

}

void filter::poles()

{

cout << "les " << ORD << " poles de votre filtre sont representes sous dessous de la forme pi =

(Re(pi),Im(pi)):" << endl;

double P1 = 0, P2 = 0, P3 = 0, P4 = 0, PT = 0;

double eps = sqrt(pow(10, AMAX / 10) - 1);

double argsieps =argsinh( 1 / eps);

double dpi = pi / 2;

double iORD = (1 / double(ORD));

double w = 2 \* pi\*FC \* 1000;

for (int K = 1; K <= ORD; K++)

{

PT = (2 \* K - 1);

P1 = sin((PT / ORD)\*dpi);

P2 = sinh(iORD \* argsieps);

P3 = cos((PT / ORD)\*dpi);

P4 = cosh(iORD \* argsieps);

if (abs(P3\*P4) < 0.000001) { P3=P4=0; }

complex<double> z(-w\*P1\*P2, w\*P3\*P4);

cout << "P" << K << " = " << z << endl;

}

cout << "..........................................................................." << endl<<endl;

}

void filter::module()

{

char nam[20];

cout << "Entrez un nom pour le fichier ou vous voulez sovgarder le module du filtre:";

cin >> nam;

cout << "..........................................................................." << endl<<endl;

ofstream file;

file.open(nam);

file<< "Vous trouvez dans ce fichier le module de votre filtre de CHEBYSHEV repondant au gabarit du

filtre passe-bas suivant:" << endl;

file<< " Amax = " << AMAX << endl;

file<< " Amin = " << AMIN << endl;

file<< " fc = " << FC << endl;

file<< " fs = " << FS << endl << endl;

file<< "module du filtre:" << endl;

double eps = sqrt(pow(10, AMAX / 10) - 1);

double eps2 = pow(eps, 2);

for (int f = (0.1\*FC); f < FC; f += (0.1\*FC))

{

float rf = (f / FC);

double imodum = eps2\*pow(cos(ORD\*acos(rf)), 2);

double imodu =sqrt( 1 + imodum);

double modu = 1 / imodu;

file<< "Pour f = " << f << " KHz : |H(jw)| = " << modu << endl;

}

for (int f = FC; f <= (10\*FC); f += FC)

{

float rf = (f / FC);

double imodum = eps2\*pow(cosh(ORD\*argcosh(rf)), 2);

double imodu = sqrt(1 + imodum);

double modu = 1 / imodu;

file<< "Pour f = " << f << " KHz : |H(jw)| = " << modu << endl;

}

file.close();

cout << "vous pouvez maintenant voir le module de votre filtre entre les valeures " << FC / 10 << "

et " << FC\* 10 << " sur le fichier texte intitule \""<<nam<<"\"" <<endl;

cout << "..........................................................................." << endl<<endl;

cout << "vous pouvez recommencer......" << endl;

cout << "..........................................................................." << endl<<endl;

}