for q := 2 to 97 do

if IsPrimePower(q) then

// C3

e := 3;

f := 3;

n := q^2 + q + 1;

k := 4;

/\*

// C4

e := 4;

f := 2;

n := q^2 + 1;

k := 5;

// C6

e := 6;

f := 1;

n := q^2 - q + 1;

k := 7;

\*/

GF := GF(q);

GFe := ext<GF | e>;

P<x> := PolynomialRing(GF);

pr := PrimitiveElement(GFe);

a := pr^((q^e - 1) div n);

m := MinimalPolynomial(a, GF);

h := (x-1)\*m;

g := (x^n-1) div h;

Ce := CyclicCode(n, g);

printf "C%o is [%o, %o, %o]\_%o code\n", e, Length(Ce), Dimension(Ce), MinimumDistance(Ce), q;

fac := Factorization(q);

p := fac[1][1];

l := fac[1][2];

fl := Floor(2\*Sqrt(q));

if IsOdd(l) and (p ne q) and (fl mod p eq 0) then

NqOne := q + fl;

else

NqOne := q + fl + 1;

end if;

printf "n - f\*(Nq(1) div f) = %o\n\n", n - f\*(NqOne div f);

end if;

end for;