function randomPoly(n, field)

poly:=[field | 0: x in[1..n]];

for i in [1..n] do

poly[i]:=Random(field);

end for;

return poly;

end function;

n:=m\*l;

Z4:=IntegerRing(4);

Z2:=GF(2);

P2<x>:=PolynomialRing(Z2);

P4<x>:=PolynomialRing(Z4);

TimeLimit:=10;

skippedTime:=0;

skippedDimCap:=0;

bestLeeDistanceLGray:=0;

bestLeeDistanceNLGray:=0;

bestNLGrayCode:=[];

bestLGrayCode:=[];

degreeGen:=Degree(gen);

try\_max:=100000;

f:=P4!(x^m-1);

check:= f div gen;

for tries in [1..try\_max] do

genList:=[];

fList:=[];

Append(~fList, gen);

for l\_i in [1..l] do

repeat

rowVec:=randomPoly(m-degreeGen, Z4);

f2:=P2!rowVec;

until Gcd(f2, (P2!check)) eq 1;

f1:=P4!rowVec;

Append(~fList, f1);

Append(~genList, gen\*f1);

end for;

C:=QuasiCyclicCode(n, genList);

MLW:=MinimumLeeWeight(C);

D, k2, k1:=MinRowsGeneratorMatrix(C);

if n gt 61 then

if (Minimum(2\*n-2\*k1-k2,2\*k1+k2) le 60) then

MLW:=MinimumLeeWeight(C: MaximumTime:=TimeLimit);

if MLW eq -1 then

skippedTime +:= 1;

end if;

else

skippedDimCap +:= 1;

end if;

else

MLW:=MinimumLeeWeight(C); // no time limit in this case when n is smaller or equal to 61

end if;

if (HasLinearGrayMapImage(C)) then

if (MLW gt bestLeeDistanceLGray) then

bestLeeDistanceLGray:=MLW;

PrintFile(file, "&&&");

PrintFile(file, "&");

PrintFile(file, "With Linear Gray Map Image");

PrintFile(file, "&");

PrintFile(file, n);

PrintFile(file, "&");

PrintFile(file, m-degreeGen);

PrintFile(file, "&");

PrintFile(file, bestLeeDistanceLGray);

PrintFile(file, "&");

PrintFile(file, C);

PrintFile(file, "&");

PrintFile(file, fList);

PrintFile(file, "&&&&");

end if;

else

if (MLW gt bestLeeDistanceNLGray) then

bestLeeDistanceNLGray:=MLW;

PrintFile(file, "&&&");

PrintFile(file, "Without Linear Gray Map Image");

PrintFile(file, "&");

PrintFile(file, n);

PrintFile(file, "&");

PrintFile(file, m-degreeGen);

PrintFile(file, "&");

PrintFile(file, bestLeeDistanceNLGray);

PrintFile(file, "&");

PrintFile(file, C);

PrintFile(file, "&");

PrintFile(file, fList);

PrintFile(file, "&&&&");

end if;

end if;

end for;

print("Number of codes skipped due to time limit:"), skippedTime;

print("Number of codes skipped due to the limit of minimum between dimension and n-dimension:"), skippedDimCap;