%%% Cyclic code

clc; clear all;

pkg load communications;

#{

adds named package 'communication package' here to the path ,

after loading package it is possible to use functions provided by the package

#}

n=input('Enter the length of code word:')

k=input('Enter the length of msg word:')

m=input('Enter the msg word:')

G=cyclpoly(n,k,'max')%https://octave.sourceforge.io/communications/function/cyclpoly.html

gx=polyout(G,'x') %https://octave.sourceforge.io/octave/function/polyout.html

disp('Codeword for the msg word is:')

C=encode(m,n,k,'cyclic',G)%%https://octave.sourceforge.io/communications/function/encode.html

disp('Decoded Codeword is:')

D=decode(C,n,k,'cyclic',G)

%{

OUTPUT:

Enter the length of code word7

n = 7

Enter the length of msg word4

k = 4

Enter the msg word [1 0 1 0]

m =

1 0 1 0

G =

1 0 1 1

gx = 1\*s^3 + 0\*s^2 + 1\*s^1 + 1

Codeword for the msg word is:

C =

0 1 1 1 0 1 0

Decoded Codeword is:

D =

1 0 1 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION DESCRIPTION:

CYCLPOLY (N, K, OPT)

This function returns the cyclic generator polynomials of the code [n,k]. By default

the polynomial with the smallest weight is returned. However this behavior can be overridden with the opt flag

"all"-Returns all of the polynomials of the code [n,k]

"min"-Returns the polynomial of minimum weight of the code [n,k]

"max"-Returns the polynomial of the maximum weight of the code [n,k]

l-Returns the polynomials having exactly the weight l

The polynomials are returns as row-vectors in the variable y.

Each row of y represents a polynomial with the least-significant term first.

POLYOUT (C),POLYOUT (C, X)

: str = polyout (…)

Display a formatted version of the polynomial c.

The formatted polynomial

c(x) = c(1) \* x^n + … + c(n) x + c(n+1)

is returned as a string or written to the screen if nargout is zero.

The second argument x specifies the variable name to use for each term and defaults to the string "s".

code = encode (msg, n, k, typ)

code = encode (msg, n, k, typ, opt)

code, added] = encode (…)

Top level block encoder. This function makes use of the lower level functions such as cyclpoly, cyclgen, hammgen, and bchenco. The message to code is pass in msg, the codeword length is n and the message length is k. This function is used to encode messages using either:

A [n,k] linear block code defined by a generator matrix

A [n,k] cyclic code defined by a generator polynomial

A [n,k] Hamming code defined by a primitive polynomial

A [n,k] BCH code code defined by a generator polynomial

The type of coding to use is defined by the variable typ. This variable is a string taking one of the values

cyclic/binary"

A cyclic code is assumed with the coded message code being in a binary format. The generator polynomial to use can be defined in opt.

The default generator polynomial to use will be cyclpoly (n, k)