enge AIM I page to have man of Dans persons 234

Write a program to implement error detection and correction using Hamming code concept. Make a test run to imput data stream and verify error correction.

Error correction at Data Rink Rayer:

Hanning code is set of error correction codes that can be used to detect and correct the errors that can when the data is transmitted from the sender to the receiver. It is a technique developed by R.W. Hamming for error correction!

Create sender program with below features:

- 1) Input to sender file should be a text any length Program should convert the text to billary.
- 2) Apply ranning code concept on the binary data add redundant bits to it.

: (T. T POUT COU (COUNT) : create a receiver program with below features.

- 1) Receiver program should react the input from channel file.
- (2) Apply havining code on the binary data to check · ((1 = " 5) = (1 " = (1)) } for errors.
 - 3) If there is an error, display the position of error

4) Then remove the redemidant bits and convey the vinary data to ascil and display the oupy anguage of our morpore of a source Prograu! def calc_r(l): possessed possess some for i'm range (2): 130333337433 7077 収(2××1)=1+じ+1): def pos-red-bits(b, r): j. k = 0,11 3237 193 30110 939999 94 953611 99 1953 belts " " bothwelted is transmetted " from sind for i in range (1, leu(b) + r+1)! if (ita 2 to promo of promote was you bits + = "0" grithalf walled the walk paral feathing 9+=1 signede purele est o ou senous oil souse of sugue (1 bits+=16[-13k]3+ sound believe morpary 2) Apply rawling code concept on the burden dance retuen bits [:-1] I of edial duplous poor det calc-parity (arr, r): n=len(arr) MARCHARD PARCEUL for i in range (r): 2123 33303353 Val = 0 do to for of ell range (1, 4+1) is prevent peg 14 (5 4 (12(2**i)==(2**i): 230100 101 val = val^int[arr[-1*j])
arr-arr[:n-(2**i)]+str(val)+arr[n-(2**i)+1:]

```
return arr
    det detect-err(arr, r):
( topos u- len (arr) mossessor mos por journey
      res = 0
                               33377 3323993
for i in range(r):
        Val = 0
       (100000 sus) moss of 1, 20 but board -1) quit
        for jeurange (1, u+1):
         if (j2(2++i)==(2++i)):
    val-val^int[arr[-i*j])
       nes = nes + val + (10 + x)
                                2121373393
      return int (str(res), 2)
    det fup (data, pos): 2009 quit 119 0000-2003
  if post 1 or posteu (data):
        print (" Invalid position!")
        retuen data be sure dos sons besides besides
                          data-list = list (data)
      data_list[pos-1]='1' if data_list[pos-1]=='0'else'o'
      return ". join (data-list)
               det bill-to-dec(b):
          110 - (110 adde 200) 110) - 122 - 209 - 149
      return int (b, 2)
   S= input (" Enter a string to encode: ")
   bin-val=. join ([bin(ord(c))[2:].zfill(8) for cins])
   print(f"Binary representation of '{$3': ?bin-val]")
  l=leu(bin-val)
                 wall man on " ) wall 303 mas
  print (f'Number of redundant bits: ? r's)
```

```
pos= pos-red-bits (bill-val, r)
      enc-data - calc-parity cpos, r) 1000 100 100 100 100
      print (f" Data with redundant bits: senc data)
       err-pos-int(input(f"Enter the position of the bit,
         flip(1-based index.1 to Elen(enc.data)3):"))
        y err. pos un [2**i for i un range(r)]:
         print ("cannot flip a redundant bit position.
                  Please enter a valid position!
                         (6, (228) 42) July ward
          continue
        else:
         enc.data.err: flip(enc.data, err-pos)
         print (f" Data with error introduced: senc data en
                     Three borders is
          break
      err-detected = detect_err(euc_data_err,r)
                         (1000) Jul : Jest 1000
      lf err_detected = = 0:
print ('No error detected en the received data")
                        (JULL DIDD) NIBE BELLING
      else
        err-pos-detected = err-detected
                                  err-pos-left = leuceuc-data_err)-err-pos-detected+1
        bill-err-pos: bill(err-pos-left)[2:].zfill(4)
       dec-err-pos: bin-to-dec(bin-err-pos)
       print(f" Error detected at position: ferr-pos-left?")
print (f"Binary ernor position: { bin_err-pos}.
               Decimal position: ¿ dec-err-pos j')
      correct : input (" Do you want to correct the error?
          (yes/no)."). strip(). lower()
```

if correct = - 'yes': corrected.data-flip(euc-data_err, err-pos_left) prille (f" cornected data: Ecometted data)") else: THE BOOK PROPERTY DESIGNATION OF THE PARTY O print ("Error was not corrected.") 35 3800 3W3 WORK BOOK 10 WOJ 2M STONES output: enter a string to encode: Hi Binary representation of 'Hi': 0100100001101001 Number of redundant bits: 5 PARIS SELECTIONS OF SELECTIONS OF SELECTION SE Data with redundant bits: 01001000001101000100 Enter the position of the bit to flip (1 to 21): 4 cannot flip a redundant bit position Please enter a valid position Enter the position of the bit to flip (1 to 21): 6 Data with error introduced: 010011000011001000100 I ATAT Enter detected at position: 6 Binary error position: 0110, Decimal position: 6 po you want to correct the error? (yes/no): yes vected data: 010010000011001000000 code Execution successful =

esuit.

Thus, the error detection and correction using tamming code concept was successfully implement and executed.