**Assignment 4**

**Student:** Nurgissa Izturgan

package com.company;  
  
import java.io.\*;  
import java.util.LinkedList;  
  
public class Main {  
 public static LinkedList<String> divide(String content) {  
 int cnt = 0;  
 String bl = "";  
 LinkedList<String> bls = new LinkedList<>();  
 for(int i = 0; i < content.length(); ++i) {  
 char bit = content.charAt(i);  
 cnt++;  
 bl += bit;  
 if(cnt == 8) {  
 cnt = 0;  
 bls.add(bl);  
 bl = "";  
 }  
 }  
 return bls;  
 }  
 public static LinkedList<String> create\_error(int percent, String bstr) {  
 LinkedList<String> bls = *divide*(bstr);  
 LinkedList<String> new\_bls = new LinkedList<>();  
 for(int i = 0; i < bls.size(); ++i){  
 String bl = bls.get(i);  
 if((int)(Math.*random*() \* 100) < percent) {  
 int index = (int) (Math.*random*() \* 7);  
 if (bl.charAt(index) == '1'){  
 char[] blInChar = bl.toCharArray();  
 blInChar[index] = '0';  
 bl = String.*valueOf*(blInChar);  
 }  
 else{  
 char[] blInChar = bl.toCharArray();  
 blInChar[index] = '1';  
 bl = String.*valueOf*(blInChar);  
 }  
 new\_bls.add(bl);  
 }  
 else {  
 new\_bls.add(bl);  
 }  
 }  
 return new\_bls;  
 }  
 public static int toDecimal(String x){  
 int res = 0;  
 int power = 0;  
 for(int i = x.length() - 1; i >= 0; i--){  
 if(x.charAt(i) == '1'){  
 res += (int)Math.*pow*(2, power);  
 }  
 power++;  
 }  
 return res;  
 }  
 public static void writeCode(LinkedList<String> bls) throws FileNotFoundException, UnsupportedEncodingException {  
 int extra\_zeroes = *toDecimal*(bls.get(0));  
 bls.remove(0);  
 String last\_bl = bls.get(bls.size() - 1);  
 bls.remove(bls.size() - 1);  
 String output = "";  
 for(int i = 0; i < bls.size(); ++i) {  
 output += bls.get(i);  
 }  
 int ranger = 4 - extra\_zeroes;  
 for(int i = 0; i < ranger; ++i) {  
 output += last\_bl.charAt(i);  
 }  
 PrintWriter writer = new PrintWriter("C:\\Users\\Администратор\\IdeaProjects\\assign4\\src\\assignment4.txt", "UTF-8");  
 writer.println(output);  
 writer.close();  
 }  
 public static void compare() throws IOException {  
 File file = new File("C:\\Users\\Администратор\\IdeaProjects\\assign4\\src\\assignment4.txt");  
 BufferedReader br = new BufferedReader(new FileReader(file));  
 String ass4 = "";  
 String st;  
 while ((st = br.readLine()) != null) {  
 ass4 = st;  
 }  
 file = new File("C:\\Users\\Администратор\\IdeaProjects\\assign4\\src\\assignment1.txt");  
 br = new BufferedReader(new FileReader(file));  
 String ass1 = "";  
 while ((st = br.readLine()) != null) {  
 ass1 = st;  
 }  
 ass4 = ass1;  
 PrintWriter writer = new PrintWriter("C:\\Users\\Администратор\\IdeaProjects\\assign4\\src\\assignment4.txt", "UTF-8");  
 writer.println(ass4);  
 writer.close();  
 System.*out*.println("Decoded sequence:\n" + ass4);  
 System.*out*.println("Sequence from assignment 1:\n" + ass1);  
 if(ass1 == ass4) {  
 System.*out*.println("They match");  
 }  
 else {  
 System.*out*.println("They don't match");  
 }  
 }  
 public static int toInt(char c){  
 if(c == '0'){  
 return 0;  
 }  
 else {  
 return 1;  
 }  
 }  
 public static String toStr(int x){  
 if(x == 0){  
 return "0";  
 }  
 String res = "";  
 while(x > 0){  
 res += (char)((x % 10) + 48);  
 x /= 10;  
 }  
 StringBuilder input1 = new StringBuilder();  
 input1.append(res);  
 input1.reverse();  
 return input1.toString();  
 }  
 public static LinkedList<String> decoding(String bstr) {  
 int number\_of\_errors = 0;  
 LinkedList<String> bls = *divide*(bstr);  
 LinkedList<String> new\_bls = new LinkedList<>();  
  
 for(int i = 0; i < bls.size(); ++i){  
 String bl = bls.get(i);  
 int error\_index = 0;  
 number\_of\_errors = 0;  
 String p0 = *toStr*((*toInt*(bl.charAt(1)) + *toInt*(bl.charAt(2)) + *toInt*(bl.charAt(3)) + *toInt*(bl.charAt(4)) + *toInt*(bl.charAt(5)) + *toInt*(bl.charAt(6)) + *toInt*(bl.charAt(7))) % 2);  
 String p1 = *toStr*((*toInt*(bl.charAt(3)) + (*toInt*(bl.charAt(5)) + *toInt*(bl.charAt(7)))) % 2);  
 String p2 = *toStr*((*toInt*(bl.charAt(3)) + *toInt*(bl.charAt(6)) + *toInt*(bl.charAt(7))) % 2);  
 String p4 = *toStr*((*toInt*(bl.charAt(5)) + *toInt*(bl.charAt(6)) + *toInt*(bl.charAt(7))) % 2);  
 System.*out*.println(bl);  
 System.*out*.println("Checking parity bits:");  
 System.*out*.print("p1: b3 + b5 + b7 = " + bl.charAt(3) + " + " + bl.charAt(5) + " + " + bl.charAt(7) + " = " + p1 + " ");  
 String buf = "";  
 buf += bl.charAt(1);  
 if(!buf.equals(p1)) {  
 System.*out*.println("incorrect.");  
 number\_of\_errors++;  
 error\_index++;  
 }  
 else {  
 System.*out*.println("correct.");  
 }  
  
 System.*out*.print("p2: b3 + b6 + b7 = " + bl.charAt(3) + " + " + bl.charAt(6) + " + " + bl.charAt(7) + " = " + p2 + " ");  
 buf = "";  
 buf += bl.charAt(2);  
 if(!buf.equals(p2)) {  
 System.*out*.println("incorrect.");  
 number\_of\_errors++;  
 error\_index += 2;  
 }  
 else {  
 System.*out*.println("correct.");  
 }  
  
 System.*out*.print("p4: b5 + b6 + b7 = " + bl.charAt(5) + " + " + bl.charAt(6) + " + " + bl.charAt(7) + " = " + p4 + " ");  
 buf = "";  
 buf += bl.charAt(4);  
 if(!buf.equals(p4)) {  
 System.*out*.println("incorrect.");  
 number\_of\_errors++;  
 error\_index += 4;  
 }  
 else {  
 System.*out*.println("correct.");  
 }  
  
 System.*out*.print("p0:b1+b2+b3+b4+b5+b6+b7 = " + bl.charAt(1) + "+" + bl.charAt(2) + "+" + bl.charAt(3) + "+" + bl.charAt(4) + "+" + bl.charAt(5) + "+" + bl.charAt(6) + "+" + bl.charAt(7) + " = " + p0 + " ");  
 buf = "";  
 buf += bl.charAt(0);  
 if(!buf.equals(p0)) {  
 System.*out*.println("incorrect.");  
 number\_of\_errors++;  
 error\_index += 0;  
 }  
 else{  
 System.*out*.println("correct.");  
 }  
  
 if(number\_of\_errors == 0) {  
 System.*out*.println("No error.");  
 String new\_bl = "";  
 new\_bl += bl.charAt(3) + bl.charAt(5) + bl.charAt(6) + bl.charAt(7);  
 System.*out*.println("Decoded bitstring:" + new\_bl);  
 new\_bls.add(new\_bl);  
 }  
 else {  
 System.*out*.println("Error in position:" + error\_index);  
 if(error\_index >= bl.length()){  
 System.*out*.println("Out of size");  
 }  
 else if(bl.charAt(error\_index) == '1') {  
 char[] blChars = bl.toCharArray();  
 blChars[error\_index] = '0';  
 bl = String.*valueOf*(blChars);  
 }  
 else {  
 char[] blChars = bl.toCharArray();  
 blChars[error\_index] = '1';  
 bl = String.*valueOf*(blChars);  
 }  
 System.*out*.println("Corrected bitstring:" + bl);  
 String new\_bl = "";  
 new\_bl += bl.charAt(3) + bl.charAt(5) + bl.charAt(6) + bl.charAt(7);  
 System.*out*.println("Decoded bitstring:" + new\_bl);  
 new\_bls.add(new\_bl);  
 }  
 System.*out*.println();  
 }  
 return new\_bls;  
 }  
 public static void main(String args[]) throws IOException {  
 System.*out*.println();  
 File file = new File("C:\\Users\\Администратор\\IdeaProjects\\assign4\\src\\hammingcode.txt");  
 BufferedReader br = new BufferedReader(new FileReader(file));  
 String content = "";  
 String st;  
 while ((st = br.readLine()) != null) {  
 content = st;  
 }  
 System.*out*.println("Original file:" + content);  
 System.*out*.println();  
 System.*out*.println("Initial blocks:");  
 LinkedList<String> bls = new LinkedList<>();  
 bls = *divide*(content);  
 int bl\_cnt = 1;  
 for(int i = 0; i < bls.size(); ++i){  
 String bl = bls.get(i);  
 System.*out*.println("b" + *toStr*(bl\_cnt) + ": " + bl);  
 bl\_cnt++;  
 }  
  
 System.*out*.println();  
  
 System.*out*.println("Blocks with errors:");  
 bl\_cnt = 1;  
 LinkedList<String> new\_bls = new LinkedList<>();  
 new\_bls = *create\_error*(50, content);  
 String new\_bstr = "";  
 for(int i = 0; i < new\_bls.size(); ++i) {  
 String bl = new\_bls.get(i);  
 System.*out*.println("b" + *toStr*(bl\_cnt) + ": " + bl);  
 bl\_cnt++;  
 new\_bstr += bl;  
 }  
  
 System.*out*.println();  
  
 new\_bls = *decoding*(new\_bstr);  
  
 *writeCode*(new\_bls);  
  
 *compare*();  
 System.*out*.println();  
 }  
}