**STAGEONE**

/\*\*

\* Write a description of class StageOne here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

import java.util.\*;

public class StageOne

{

String textValue;

ArrayList<Character> Zero;

ArrayList<Character> One;

ArrayList<Character> Two;

ArrayList<Character> Three;

ArrayList<Character> Four;

ArrayList<Character> Five;

ArrayList<Character> Six;

int currentBase;

String finalText = "";

public StageOne(String inputString){

textValue = inputString;

Zero = new ArrayList<Character>();

One = new ArrayList<Character>();

Two = new ArrayList<Character>();

Three = new ArrayList<Character>();

Four = new ArrayList<Character>();

Five = new ArrayList<Character>();

Six = new ArrayList<Character>();

}

public void newText(String inputString){

textValue = inputString;

}

private void resetArrayList(){

Zero.clear();

One.clear();

Two.clear();

Three.clear();

Four.clear();

Five.clear();

Six.clear();

}

public String Encrypt(){

int a;

int currentValue = 0;

String newTrail = "";

finalText = "";

for(int x = 0; x<textValue.length(); x++){

resetArrayList();

currentBase = (int)(Math.random()\*6)+2;

finalText += currentBase;

a = 0;

newTrail = "";

for(int i = 0; i<26; i++){

switch(a){

case 0: Zero.add((char)(i+65));

break;

case 1: One.add((char)(i+65));

break;

case 2: Two.add((char)(i+65));

break;

case 3: Three.add((char)(i+65));

break;

case 4: Four.add((char)(i+65));

break;

case 5: Five.add((char)(i+65));

break;

case 6: Six.add((char)(i+65));

break;

}

if(a==currentBase-1){

a = -1;

}

a++;

}

currentValue = textValue.codePointAt(x);

while(currentValue>0){

switch(currentValue%currentBase){

case 0: newTrail += Zero.get((int)(Math.random()\*Zero.size()));

break;

case 1: newTrail += One.get((int)(Math.random()\*One.size()));

break;

case 2: newTrail += Two.get((int)(Math.random()\*Two.size()));

break;

case 3: newTrail += Three.get((int)(Math.random()\*Three.size()));

break;

case 4: newTrail += Four.get((int)(Math.random()\*Four.size()));

break;

case 5: newTrail += Five.get((int)(Math.random()\*Five.size()));

break;

case 6: newTrail += Six.get((int)(Math.random()\*Six.size()));

break;

}

currentValue = (int)(currentValue/currentBase);

}

for(int y = newTrail.length()-1; y>=0; y--){

finalText += newTrail.substring(y,y+1);

}

}

return finalText;

}

public String Decrypt(){

int a = 0;

String StringStuffs = "";

int finalValue = 0;

finalText = "";

String test = "";

for(int i = 0; i<textValue.length(); i++){

if(textValue.codePointAt(i)<65){

resetArrayList();

if(StringStuffs!= ""){

for(int y=0; y<StringStuffs.length(); y++){

finalValue += (int)(StringStuffs.codePointAt(y)-48)\*Math.pow(currentBase,(StringStuffs.length()-1-y));

}

finalText += (char)(finalValue);

finalValue = 0;

}

currentBase = (int)(textValue.codePointAt(i)-48);

a = 0;

for(int j = 0; j<26; j++){

switch(a){

case 0: Zero.add((char)(j+65));

break;

case 1: One.add((char)(j+65));

break;

case 2: Two.add((char)(j+65));

break;

case 3: Three.add((char)(j+65));

break;

case 4: Four.add((char)(j+65));

break;

case 5: Five.add((char)(j+65));

break;

case 6: Six.add((char)(j+65));

break;

}

if(a==currentBase-1){

a = -1;

}

a++;

}

StringStuffs = "";

}

else{

for(int d = 0; d<Zero.size(); d++){

test += Zero.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 0;

}

test = "";

}

for(int d = 0; d<One.size(); d++){

test += One.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 1;

}

test = "";

}

for(int d = 0; d<Two.size(); d++){

test += Two.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 2;

}

test = "";

}

for(int d = 0; d<Three.size(); d++){

test += Three.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 3;

}

test = "";

}

for(int d = 0; d<Four.size(); d++){

test += Four.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 4;

}

test = "";

}

for(int d = 0; d<Five.size(); d++){

test += Five.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 5;

}

test = "";

}

for(int d = 0; d<Six.size(); d++){

test += Six.get(d);

if(test.equals(textValue.substring(i,i+1))){

StringStuffs += 6;

}

test = "";

}

}

}

if(StringStuffs!= ""){

for(int y=0; y<StringStuffs.length(); y++){

finalValue += (int)(StringStuffs.codePointAt(y)-48)\*Math.pow(currentBase,(StringStuffs.length()-1-y));

}

finalText += (char)(finalValue);

finalValue = 0;

}

return finalText;

}

public void printEncryption(){

System.out.println(finalText);

}

public void printDecryption(){

System.out.println(finalText);

}

}

**STAGETWO**

import java.io.\*;

public class StageTwo

{

int[] splitLength;

String textValue;

String pathStuff ="";

String finalText = "";

public StageTwo(String pathFILE, String text){

textValue = text;

pathStuff = pathFILE;

splitLength = new int[7];

}

public void newText(String text){

textValue = text;

finalText ="";

}

private BufferedReader constructStuffs(String pathFILE){

try{

return new BufferedReader(new InputStreamReader(new FileInputStream(pathStuff+"/" + pathFILE + ".txt")));

}catch(IOException ex){

System.out.println(ex);

return null;

}

}

public String Encrypt(){

finalText = "";

BufferedReader bi;

String newStuffs ="";

int sumValue = 0;

try{

for(int a = 1; a<= 7; a++){

sumValue = 0;

for(int i = 0; i<textValue.length() ; i++){

bi = constructStuffs(textValue.substring(i,i+1));

for(int b = 0; b<a; b++){

newStuffs = bi.readLine() + " ";

}

finalText +=newStuffs;

sumValue += newStuffs.length();

}

splitLength[a-1] = sumValue;

}

}catch(IOException ex){

System.out.println(ex);

return null;

}

return finalText;

}

public void printEncryptionScreen(){

int startPoint = 0;

for(int i = 0; i<splitLength.length;i++)

{

System.out.println(finalText.substring(startPoint,startPoint+splitLength[i]));

startPoint += splitLength[i];

}

}

public String Decrypt(){

int letterLength = 0;

String tempTest = textValue;

int lineLength = textValue.length()/7;

String checkText;

String testValue;

BufferedReader bi;

int letterStart = 0;

finalText = "";

for(int b = 0; tempTest.indexOf("= ")!= -1;b=b){

checkText = "";

testValue = "";

tempTest = tempTest.substring(tempTest.indexOf("="));

letterLength = tempTest.indexOf("= ");

tempTest = tempTest.substring(letterLength+2);

for(int i = 0; i<6; i++){

checkText += textValue.substring(letterStart+i\*lineLength, letterStart+i\*lineLength+letterLength+1);

}

for(int i = 0; i<36; i++)

{

testValue = "";

try{

if(i<10){

testValue += (char)(i+48);

bi = constructStuffs(testValue);

testValue = bi.readLine() +bi.readLine()+bi.readLine()+bi.readLine()+bi.readLine()+bi.readLine();

if(testValue.equals(checkText)){

finalText += (char)(i+48);

}

}

else{

testValue += (char)(i+55);

bi = constructStuffs(testValue);

testValue = bi.readLine() +bi.readLine()+bi.readLine()+bi.readLine()+bi.readLine()+bi.readLine();

if(testValue.equals(checkText)){

finalText += (char)(i+55);

}

}

}catch(IOException ex){

System.out.println(ex);

return null;

}

}

letterStart += letterLength+2;

}

return finalText;

}

public void printDecryptScreen(){

System.out.println(finalText);

}

}

**MAIN**

/\*\*

\* Write a description of class Main here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Main

{

String value;

StageOne O;

StageTwo T;

String Exchange;

boolean newStuff;

public Main(String input){

value = input;

O = new StageOne("");

T = new StageTwo("Letters", "");

newStuff = true;

}

public void newString(String input){

value = input;

newStuff = true;

}

public String Encrypt(){

if(newStuff)

O.newText(value);

else

O.newText(Exchange);

T.newText(O.Encrypt());

Exchange = T.Encrypt();

newStuff = false;

return Exchange;

}

public String Decrypt(){

if(newStuff)

T.newText(value);

else

T.newText(Exchange);

O.newText(T.Decrypt());

Exchange = O.Decrypt();

newStuff = false;

return Exchange;

}

public void PrintSlip(){

T.printEncryptionScreen();

}

}