//Michael Weyman

/\*\*This program takes in a message via input file, encrypts the message and outputs,

then decrypts the message and outputs\*/

#include <stdio.h>

#include <stdlib.h>

void main(){

//stores the message from input file

char message[5000];

// length of code array

int len = 10;

//the code array for shifting

int code[] = {5, 2, -1, 2, -3, 2, -4, -2, 6, 4};

//pointer for the message

char \*ip;

//pointer for the code

int \*ip1;

//pointing at the message array

ip = message;

//pointing at the code array

ip1 = code;

//this function takes a message from an input file and puts it into a pointed array

void inputMessage(char \*);

//this function encrypts the message using the code array

void encryptMessage(char \*, int \*, int);

//this function decrypts the encrypted message

void decryptMessage(char \*, int \*, int);

//outputs the message

void output(char \*);

inputMessage(ip);

encryptMessage(ip,ip1,len);

//prints off header for the encrypted message

printf("encrypted message:\n");

output(ip);

decryptMessage(ip,ip1,len);

//prints off header for the decrypted message

printf("decrypted message:\n");

output(ip);

}

/\*\*this function takes a message from an input file and puts it into a pointed array\*/

#include <stdio.h>

#include <stdlib.h>

void inputMessage(char \*ip){

//file variable

FILE \*fp;

//open the file

fp = fopen("message0.txt", "r");

//while loop puts the file into the message character by character

while(!feof(fp)){

\*ip = getc(fp);

ip += 1;

}

//adding \0 to the end of message

\*ip = '\0';

}

/\*\*this function encrypts the message using the code array\*/

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

//constant for number of letters in the alphabet

#define CON 26

void encryptMessage(char \*ip, int \*ip1, int len){

//count variable used to track the code array

int count = 0;

//temp variable used to hold message location

int \*temp;

//setting temp to beginning of message

temp = &ip1[0];

//runs until the message ends

while(\*ip != '\0'){

//reseting the code array

if(count == len){

temp = &ip1[0];

count = 0;

}

//checks if the character is part of the alphabet

if(isalpha(\*ip)){

//shifts the char

\*ip = (char)\*ip + \*temp;

//checks to see if it is not a letter

if(!isalpha(\*ip)){

//if the shift is negative shift to end of alphabet

if(\*temp < 0)

\*ip = \*ip + CON;

//if the shift is positive shift to beginning of alphabet

else if(\*temp > 0)

\*ip = \*ip - CON;

}

//increment the code

temp++;

//increment count

count = count + 1;

}

//increment the message

ip += 1;

}

}

/\*\*this function decrypts the encrypted message\*/

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

//constant for number of letters in the alphabet

#define CON 26

void decryptMessage(char \*ip, int \*ip1, int len){

//count variable used to track the code array

int count = 0;

//temp variable used to hold message location

int \*temp;

//setting temp to beginning of message

temp = &ip1[0];

//runs until the message ends

while(\*ip != '\0'){

//reseting the code array

if(count == len){

temp = &ip1[0];

count = 0;

}

//checks if the character is part of the alphabet

if(isalpha(\*ip)){

//shifts the char

\*ip = (char)\*ip - \*temp;

//checks to see if it is not a letter

if(!isalpha(\*ip)){

//if the shift is negative shift to end of alphabet

if(\*temp < 0)

\*ip = \*ip - CON;

//if the shift is positive shift to beginning of alphabet

else if(\*temp > 0)

\*ip = \*ip + CON;

}

//increment the code

temp++;

//increment count

count = count + 1;

}

//increment the message

ip += 1;

}

}

/\*\*Outputs the message\*/

#include <stdio.h>

#include <stdlib.h>

void output(char \*ip){

//outputs the message using pointers

while(\*ip != '\0'){

printf("%c", \*ip);

ip += 1;

}

}

Output:

Message0:

encrypted message:

Npeqoowross kr plv gluaqgcib,

mjmcpjffg fu jmz anucqj,

yeqjsr kr plv ppaxm,

vqwqj eq tsy ddcrvu,

zkezvx kp pkr rsag,

kqsg eq tsy otufe,

ISYMH kr veg XCYX!

decrypted message:

Information is not knowledge,

knowledge is not wisdom,

wisdom is not truth,

truth is not beauty,

beauty is not love,

love is not music,

MUSIC is the BEST!

Message1:

Zm tsy uogxm kd clfv vg'sg ockr, th vcqgn anspkmi xncyk.

Esf cwpv sfkvj hnwkvwgtw zudf qq xc, ox bkkn kqp zk ysfnpb.

C sgrhjtmgpu qlxeagkkki, sc'bi tpka gwor hilwm.

Fl pkr ytjcj qc kyc xiytdcq, qb gypfpcu bcpct fd vgg pgw.

Yth npcwpvnw kgtpnov, ya'tk ssnx lrup zkkzp.

Hv tkhj tsy dd wkfklk.

Ht pnv prayq, sm fn plv onkep, anwo vklmyj kr foa, umav aqheb ko ukep.

Vgg qkic nex rzupgz duv bqqfp vk qkip, ks yfnh lux gg tpaqjc.

decrypted message:

Do not speak of withered trees, of lichen strangled coverings.

And life just barely in the leaves, it will not be undone.

Do not speak of what we've seen, of water choking algae.

And dust where fountains used to be, it will not be undone.

A wilderness unraveling, we've only just begun.

Do not speak of ice retreat, of islands eaten by the sea.

And industry economy, we've only just begun.

It will not be undone.

Do not speak, oh do not speak, your tongue is dry, your voice is weak.

The time has passed for words to seek, it will not be undone.

Message2:

encrypted message:

Fdcwzvemt mx fdhfpab gw "npegogjak xt vgg ygor kbunzpxvemt," e ycrm lh cctiwcskki

wl kbunzpxvemt xt cbelwjr lsw vgg xrlcgvfpbg lh w eozjp rgq qb dorikmip qn bgxf

[8].

Yhvekj rni hqmvbzp ml lfpcyokprkr hjztxepcx vjenikkpgur, yjd vxug gy ssg nh

ealmzljuhbfpc rni hjztxepcxw/xaldlno/uuviu qgprklymgnd hlt pfk jjcswogo duysf hp

pewltii dhvjclq gri csvbolrorl vn hfpz rni ggrv zqhjkgyknp lh yfgvfesgou pm

kbunzkk vdcyi kgzvrtaq. Gfiwbvfqj fgw utdxfqqqrc ggdp xrljoii vn c tkzc xesid

qc kjrkvutdvxvemt twqanboo gtgqwckki onkihj qgzqcloxnqm [9], ffccluwnu [10, 11,

12],

lgakyyr xjus kkvapvvjvzvfqj [13], qzswa tpagnqzesfhpd [14], pwravfn kckiqymi

zpcgoupythnpf [9, 15], vegkpe jttlcqkkl gri ndixn ncgwtphpd [16].

decrypted message:

Abduction is defined as "inference to the best explanation," a task of generating

an explanation to account for the appearance of a given set of findings or data

[8].

Within the context of handwritten character recognition, the task is one of

hypothesizing the characters/symbols/words responsible for the features found in

scanned bitmaps and attempting to find the best collection of characters to

explain these features. Abduction has previously been applied to a wide range

of interpretation problems including speech recognition [9], diagnosis [10, 11,

12],

medical test interpretation [13], story understanding [14], natural language

understanding [9, 15], theory formation and legal reasoning [16].