North American Railroad Signal Aspects and Indications Demonstration

Signal Controller Code

30 August 2018

Kyle Dick, Alex Christmas

4,038 Lines of Arduino C Code

#include <LCD.h>

#include <LiquidCrystal\_I2C.h>

#define I2C\_LCDA 0x27

#define I2C\_LCDB 0x23

#define Rs\_pin 0

#define Rw\_pin 1

#define En\_pin 2

#define BACKLIGHT\_PIN 3

#define D4\_pin 4

#define D5\_pin 5

#define D6\_pin 6

#define D7\_pin 7

LiquidCrystal\_I2C lcdA(I2C\_LCDA, En\_pin, Rw\_pin, Rs\_pin, D4\_pin, D5\_pin, D6\_pin, D7\_pin);

LiquidCrystal\_I2C lcdB(I2C\_LCDB, En\_pin, Rw\_pin, Rs\_pin, D4\_pin, D5\_pin, D6\_pin, D7\_pin);

/\* Copyright info

SignalController Program - Arduino C, GUI Java

Copyright 2018 by Alex Christmas and Kyle Dick

\*/

char aspect[13]; // used to store the selected rule from Java

char readIn = -1; // used to read in one character at a time from the Java output

char ack = 6; // character used to tell Java that serial output has temporarly ended

int indications[5]; // used to store what indications will go to what signals

int pin\_call\_A[14]; // Stores what LEDs will be lit on each signal

int pin\_call\_B[14];

int pin\_call\_C[14];

int pin\_call\_D[14];

int pin\_call\_E[14];

int pins[54]; // Matrix that tells the program what action to perform on the individual lights

int flash[10]; // stores what lights will be flashing

int rotate = 0; // tells the program if it will be rotating through aspects

int dly = 0; // stores the time to cycle through indications

int running = 0;

char L0[20] = "Modeled by:";

char L1[20] = "Alex Christmas";

char L2[20] = "Programmed by:";

char L3[20] = "Kyle Dick";

char L4[20] = "Modeled by:";

char L5[20] = "Alex Christmas";

char L6[20] = "Programmed by:";

char L7[20] = "Kyle Dick";

char L8[20] = "Modeled by:";

char L9[20] = "Alex Christmas";

char L10[20] = "Programmed by:";

char L11[20] = "Kyle Dick";

char L12[20] = "Modeled by:";

char L13[20] = "Alex Christmas";

char L14[20] = "Programmed by:";

char L15[20] = "Kyle Dick";

char L16[20] = "Modeled by:";

char L17[20] = "Alex Christmas";

char L18[20] = "Programmed by:";

char L19[20] = "Kyle Dick";

char L20[20] = "Modeled by:";

char L21[20] = "Alex Christmas";

char L22[20] = "Programmed by:";

char L23[20] = "Kyle Dick";

int f = 0;

int osc = 1;

int pages = 0;

int visTimr = 1;

// Initalizes the program

void setup() {

Serial.begin(9600); // Starts serial communication

lcdA.begin(20, 4); // 20 characters by 4 rows

lcdA.setBacklightPin(BACKLIGHT\_PIN, POSITIVE);

lcdA.setBacklight(HIGH);

lcdA.home();

lcdB.begin(20, 4); // 20 characters by 4 rows

lcdB.setBacklightPin(BACKLIGHT\_PIN, POSITIVE);

lcdB.setBacklight(HIGH);

lcdB.home();

// Initalizes all pins to output

pinMode(2, OUTPUT); // Top Green

pinMode(3, OUTPUT); // Top Yellow

pinMode(4, OUTPUT); // Top Red

pinMode(5, OUTPUT); // Top Lunar

pinMode(6, OUTPUT); // Middle Green

pinMode(7, OUTPUT); // Middle Yellow

pinMode(8, OUTPUT); // Middle Red

pinMode(9, OUTPUT); // Middle Lunar

pinMode(10, OUTPUT); // Bottom Green

pinMode(11, OUTPUT); // Bottom Yellow

pinMode(12, OUTPUT); // Bottom Red

pinMode(13, OUTPUT); // Bottom Lunar

pinMode(22, OUTPUT); // Top Green

pinMode(23, OUTPUT); // Top Yellow

pinMode(24, OUTPUT); // Top Red

pinMode(25, OUTPUT); // Top Lunar

pinMode(26, OUTPUT); // Middle Green

pinMode(27, OUTPUT); // Middle Yellow

pinMode(28, OUTPUT); // Middle Red

pinMode(29, OUTPUT); // Middle Lunar

pinMode(30, OUTPUT); // Spare

pinMode(31, OUTPUT); // Spare

pinMode(32, OUTPUT); // Top Green

pinMode(33, OUTPUT); // Top Yellow

pinMode(34, OUTPUT); // Top Red

pinMode(35, OUTPUT); // Top Lunar

pinMode(36, OUTPUT); // Spare

pinMode(37, OUTPUT); // Spare

pinMode(38, OUTPUT); // Spare

pinMode(39, OUTPUT); // Spare

pinMode(40, OUTPUT); // Top Green

pinMode(41, OUTPUT); // Top Yellow

pinMode(42, OUTPUT); // Top Red

pinMode(43, OUTPUT); // Top Lunar

pinMode(44, OUTPUT); // Middle Green

pinMode(45, OUTPUT); // Middle Yellow

pinMode(46, OUTPUT); // Middle Red

pinMode(47, OUTPUT); // Middle Lunar

pinMode(48, OUTPUT); // Top Green

pinMode(49, OUTPUT); // Top Yellow

pinMode(50, OUTPUT); // Top Red

pinMode(51, OUTPUT); // Top Lunar

pinMode(52, OUTPUT); // Spare

pinMode(53, OUTPUT); // Spare

// Title Lines

Serial.print("SignalController Signal Demonstrator - 4th Iteration\n");

Serial.print("Version 01-Aug-2018 | Copyright Alex Christmas & Kyle Dick\n\n");

}

void printLCD(char lcd, int cntr, char L0[20], char L1[20], char L2[20], char L3[20]) {

if (lcd == 'A') {

int len0 = strlen(L0);

int len1 = strlen(L1);

int len2 = strlen(L2);

int len3 = strlen(L3);

int crsr = 0;

if (strlen(L0) > 20) {

len0 = 20;

}

if (strlen(L1) > 20) {

len1 = 20;

}

if (strlen(L2) > 20) {

len2 = 20;

}

if (strlen(L3) > 20) {

len3 = 20;

}

lcdA.setCursor(0, 0);

if (cntr == 1) {

for (int i = 0; i < (20 - len0) / 2; i++) {

lcdA.print(" ");

crsr++;

}

}

for (int i = 0; i < len0; i++) {

lcdA.print(L0[i]);

crsr++;

}

while (crsr < 20) {

lcdA.print(" ");

crsr++;

}

crsr = 0;

lcdA.setCursor(0, 1);

if (cntr == 1) {

for (int i = 0; i < (20 - len1) / 2; i++) {

lcdA.print(" ");

crsr++;

}

}

for (int i = 0; i < len1; i++) {

lcdA.print(L1[i]);

crsr++;

}

while (crsr < 20) {

lcdA.print(" ");

crsr++;

}

crsr = 0;

lcdA.setCursor(0, 2);

if (cntr == 1) {

for (int i = 0; i < (20 - len2) / 2; i++) {

lcdA.print(" ");

crsr++;

}

}

for (int i = 0; i < len2; i++) {

lcdA.print(L2[i]);

crsr++;

}

while (crsr < 20) {

lcdA.print(" ");

crsr++;

}

crsr = 0;

lcdA.setCursor(0, 3);

if (cntr == 1) {

for (int i = 0; i < (20 - len3) / 2; i++) {

lcdA.print(" ");

crsr++;

}

}

for (int i = 0; i < len3; i++) {

lcdA.print(L3[i]);

crsr++;

}

while (crsr < 20) {

lcdA.print(" ");

crsr++;

}

crsr = 0;

} else if (lcd == 'B') {

int len0 = strlen(L0);

int len1 = strlen(L1);

int len2 = strlen(L2);

int len3 = strlen(L3);

int crsr = 0;

if (strlen(L0) > 20) {

len0 = 20;

}

if (strlen(L1) > 20) {

len1 = 20;

}

if (strlen(L2) > 20) {

len2 = 20;

}

if (strlen(L3) > 20) {

len3 = 20;

}

lcdB.setCursor(0, 0);

if (cntr == 1) {

for (int i = 0; i < (20 - len0) / 2; i++) {

lcdB.print(" ");

crsr++;

}

}

for (int i = 0; i < len0; i++) {

lcdB.print(L0[i]);

crsr++;

}

while (crsr < 20) {

lcdB.print(" ");

crsr++;

}

crsr = 0;

lcdB.setCursor(0, 1);

if (cntr == 1) {

for (int i = 0; i < (20 - len1) / 2; i++) {

lcdB.print(" ");

crsr++;

}

}

for (int i = 0; i < len1; i++) {

lcdB.print(L1[i]);

crsr++;

}

while (crsr < 20) {

lcdB.print(" ");

crsr++;

}

crsr = 0;

lcdB.setCursor(0, 2);

if (cntr == 1) {

for (int i = 0; i < (20 - len2) / 2; i++) {

lcdB.print(" ");

crsr++;

}

}

for (int i = 0; i < len2; i++) {

lcdB.print(L2[i]);

crsr++;

}

while (crsr < 20) {

lcdB.print(" ");

crsr++;

}

crsr = 0;

lcdB.setCursor(0, 3);

if (cntr == 1) {

for (int i = 0; i < (20 - len3) / 2; i++) {

lcdB.print(" ");

crsr++;

}

}

for (int i = 0; i < len3; i++) {

lcdB.print(L3[i]);

crsr++;

}

while (crsr < 20) {

lcdB.print(" ");

crsr++;

}

crsr = 0;

}

}

void loop() {

// Java GUI contains all rulebook choices

if (running == 0) {

printLCD('A', 0, "N. AMERICAN RAILROAD", "SIGNAL ASPECTS", "AND INDICATIONS", "DEMONSTRATION");

printLCD('B', 0, "MODELED BY:", "ALEX CHRISTMAS", "PROGRAMMED BY:", "KYLE DICK");

Serial.print("Enter in rule number: ");

Serial.print(ack); // character used to signify end of serial output to Java

Serial.print(" ");

running = 1;

}

int aspLimit = 13; // max number of characters in rule number text - adjust in indication[] initializer

/\* Erases existing rule number data \*/

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

aspect[i] = 0;

}

while (!Serial.available()) {

if (pages >= 1) {

visTimr = 0;

lcdA.setCursor(17, 3);

if (pages > 1) {

visTimr = 1;

Serial.print(pages);

lcdA.print("1/");

lcdA.print(pages);

} else {

lcdA.print(" ");

}

printLCD('B', 0, L0, L1, L2, L3);

delayDisp(5);

}

if (pages >= 2) {

lcdA.setCursor(17, 3);

lcdA.print("2");

printLCD('B', 0, L4, L5, L6, L7);

delayDisp(5);

}

if (pages >= 3) {

lcdA.setCursor(17, 3);

lcdA.print("3");

printLCD('B', 0, L8, L9, L10, L11);

delayDisp(5);

}

if (pages >= 4) {

lcdA.setCursor(17, 3);

lcdA.print("4");

printLCD('B', 0, L12, L13, L14, L15);

delayDisp(5);

}

if (strcmp(L16, "") != 0) {

if (pages >= 5) {

lcdA.setCursor(17, 3);

lcdA.print("5");

printLCD('B', 0, L16, L17, L18, L19);

delayDisp(5);

}

if (pages >= 6) {

lcdA.setCursor(17, 3);

lcdA.print("6");

printLCD('B', 0, L20, L21, L22, L23);

delayDisp(5);

}

}

}

int i = 0; // Variable used to track the position of where the character will go in the string

while (Serial.available()) { // Reads in the Java input one character at a time

delay(15); // Halts the program for a short period so that no communication is missed

readIn = Serial.read(); // Reads in one character at a time

aspect[i++] = readIn; // Puts one character in the aspect string at a time

}

Serial.print("\n");

selAspect();

}

void delayDisp(int sec) {

/\* for (int i = sec; i >= 0; i--) {

if (Serial.available()) {break;}

if (visTimr == 1) {

if (Serial.available()) {break;}

lcdB.setCursor(19, 3);

lcdB.print(i);

}

if (Serial.available()) {break;}

if (osc == 1) {

for (int j = 255; j >= 0; j--) {

for (int i = 0; i < f; i++) { // Goes through signals marked for flashing

analogWrite(flash[i], j); // and turns them off

}

delay(10);

}

osc = 0;

} else if (osc == 0) {

for (int j = 0; j <= 255; j++) {

for (int i = 0; i < f; i++) { // Goes through signals marked for flashing

analogWrite(flash[i], j); // and turns them off

}

delay(10);

}

osc = 1;

}

delay(1000);

} \*/

for (int i = sec; i >= 0; i--) {

if (Serial.available()) {break;}

if (visTimr == 1) {

if (Serial.available()) {break;}

lcdB.setCursor(19, 3);

lcdB.print(i);

}

if (Serial.available()) {break;}

if (osc == 1) {

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

digitalWrite(flash[j], LOW);

}

osc = 0;

} else if (osc == 0) {

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

digitalWrite(flash[j], HIGH);

}

osc = 1;

}

delay(1000);

}

}

void readIN(char data[]) {

while (Serial.available() == 0) { } // Hangs the program while waiting for input from Java

int i = 0; // Variable used to track the position of where the character will go in the string

while (Serial.available()) { // Reads in the Java input one character at a time

delay(15); // Halts the program for a short period so that no communication is missed

readIn = Serial.read(); // Reads in one character at a time

data[i++] = readIn; // Puts one character in the aspect string at a time

}

}

// Flushes any remaining user input

void serialFlush() {

while (Serial.available()) { Serial.read(); }

}

void setLCDB(int maxLines, char IN0[20], char IN1[20], char IN2[20], char IN3[20], char IN4[20], char IN5[20], char IN6[20], char IN7[20], char IN8[20], char IN9[20], char IN10[20], char IN11[20], char IN12[20], char IN13[20], char IN14[20], char IN15[20], char IN16[20], char IN17[20], char IN18[20], char IN19[20], char IN20[20], char IN21[20], char IN22[20], char IN23[20]) {

pages = maxLines / 4;

if (maxLines == 4) {

strncpy(L0, IN0, 20); // 1 page

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN0, 20);

strncpy(L5, IN1, 20);

strncpy(L6, IN2, 20);

strncpy(L7, IN3, 20);

strncpy(L8, IN0, 20);

strncpy(L9, IN1, 20);

strncpy(L10, IN2, 20);

strncpy(L11, IN3, 20);

strncpy(L12, IN0, 20);

strncpy(L13, IN1, 20);

strncpy(L14, IN2, 20);

strncpy(L15, IN3, 20);

strncpy(L16, IN0, 20);

strncpy(L17, IN1, 20);

strncpy(L18, IN2, 20);

strncpy(L19, IN3, 20);

strncpy(L20, IN0, 20);

strncpy(L21, IN1, 20);

strncpy(L22, IN2, 20);

strncpy(L23, IN3, 20);

} else if (maxLines == 8) {

strncpy(L0, IN0, 20); // 2 pages

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN4, 20);

strncpy(L5, IN5, 20);

strncpy(L6, IN6, 20);

strncpy(L7, IN7, 20);

strncpy(L8, IN0, 20);

strncpy(L9, IN1, 20);

strncpy(L10, IN2, 20);

strncpy(L11, IN3, 20);

strncpy(L12, IN4, 20);

strncpy(L13, IN5, 20);

strncpy(L14, IN6, 20);

strncpy(L15, IN7, 20);

strncpy(L16, IN0, 20);

strncpy(L17, IN1, 20);

strncpy(L18, IN2, 20);

strncpy(L19, IN3, 20);

strncpy(L20, IN4, 20);

strncpy(L21, IN5, 20);

strncpy(L22, IN6, 20);

strncpy(L23, IN7, 20);

} else if (maxLines == 12) {

strncpy(L0, IN0, 20); // 3 pages

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN4, 20);

strncpy(L5, IN5, 20);

strncpy(L6, IN6, 20);

strncpy(L7, IN7, 20);

strncpy(L8, IN8, 20);

strncpy(L9, IN9, 20);

strncpy(L10, IN10, 20);

strncpy(L11, IN11, 20);

strncpy(L12, IN0, 20);

strncpy(L13, IN1, 20);

strncpy(L14, IN2, 20);

strncpy(L15, IN3, 20);

strncpy(L16, IN4, 20);

strncpy(L17, IN5, 20);

strncpy(L18, IN6, 20);

strncpy(L19, IN7, 20);

strncpy(L20, IN8, 20);

strncpy(L21, IN9, 20);

strncpy(L22, IN10, 20);

strncpy(L23, IN11, 20);

} else if (maxLines == 16) {

strncpy(L0, IN0, 20); // 4 pages

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN4, 20);

strncpy(L5, IN5, 20);

strncpy(L6, IN6, 20);

strncpy(L7, IN7, 20);

strncpy(L8, IN8, 20);

strncpy(L9, IN9, 20);

strncpy(L10, IN10, 20);

strncpy(L11, IN11, 20);

strncpy(L12, IN12, 20);

strncpy(L13, IN13, 20);

strncpy(L14, IN14, 20);

strncpy(L15, IN15, 20);

strncpy(L16, " ", 20);

strncpy(L17, " ", 20);

strncpy(L18, " ", 20);

strncpy(L19, " ", 20);

strncpy(L20, " ", 20);

strncpy(L21, " ", 20);

strncpy(L22, " ", 20);

strncpy(L23, " ", 20);

} else if (maxLines == 20) {

strncpy(L0, IN0, 20); // 5 pages

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN4, 20);

strncpy(L5, IN5, 20);

strncpy(L6, IN6, 20);

strncpy(L7, IN7, 20);

strncpy(L8, IN8, 20);

strncpy(L9, IN9, 20);

strncpy(L10, IN10, 20);

strncpy(L11, IN11, 20);

strncpy(L12, IN12, 20);

strncpy(L13, IN13, 20);

strncpy(L14, IN14, 20);

strncpy(L15, IN15, 20);

strncpy(L16, IN16, 20);

strncpy(L17, IN17, 20);

strncpy(L18, IN18, 20);

strncpy(L19, IN19, 20);

strncpy(L20, " ", 20);

strncpy(L21, " ", 20);

strncpy(L22, " ", 20);

strncpy(L23, " ", 20);

} else if (maxLines == 24) {

strncpy(L0, IN0, 20); // 6 pages

strncpy(L1, IN1, 20);

strncpy(L2, IN2, 20);

strncpy(L3, IN3, 20);

strncpy(L4, IN4, 20);

strncpy(L5, IN5, 20);

strncpy(L6, IN6, 20);

strncpy(L7, IN7, 20);

strncpy(L8, IN8, 20);

strncpy(L9, IN9, 20);

strncpy(L10, IN10, 20);

strncpy(L11, IN11, 20);

strncpy(L12, IN12, 20);

strncpy(L13, IN13, 20);

strncpy(L14, IN14, 20);

strncpy(L15, IN15, 20);

strncpy(L16, IN16, 20);

strncpy(L17, IN17, 20);

strncpy(L18, IN18, 20);

strncpy(L19, IN19, 20);

strncpy(L20, IN20, 20);

strncpy(L21, IN21, 20);

strncpy(L22, IN22, 20);

strncpy(L23, IN23, 20);

} else {

pages = 1;

strncpy(L0, "ERROR:", 20);

strncpy(L1, "maxLines for aspect", 20);

strncpy(L2, "exceed 24 or is not", 20);

strncpy(L3, "divisible by 4.", 20);

strncpy(L4, "Report to dev team.", 20);

}

}

// Selects the specified aspect to output

void selAspect() {

if (strcmp(aspect, "CSX 1281") == 0) { // If the indication matches the rule...

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, "CSX 1281", " ", "CLEAR", " ");

setLCDB(4, "PROCEED.", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");

} else if (strcmp(aspect, "CSX 1281B") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, "CSX 1281B", " ", "APPROACH", "LIMITED");

setLCDB(4,

"PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING LIMITED",

"SPEED.", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");

} else if (strcmp(aspect, "CSX 1281C") == 0) {

initAspects(15, 16, 0, 16, 0);

printLCD('A', 1, "CSX 1281C", " ", "LIMITED", "CLEAR");

setLCDB(8, "LIMITED SPEED",

"THROUGH TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED AT",

"POSTED SPEED.", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");

} else if (strcmp(aspect, "CSX 1281D") == 0) {

initAspects(17, 18, 0, 18, 0);

printLCD('A', 1, "CSX 1281D", " ", "LIMTED", "APPROACH");

setLCDB(8, "LIMITED SPEED",

"THROUGH TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED,",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");

} else if (strcmp(aspect, "CSX 1282") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, " ", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING MEDIUM",

"SPEED.", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");

} else if (strcmp(aspect, "CSX 1282A") == 0) {

initAspects(21, 22, 0, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(4, "PROCEED, PREPARED",

"TO STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1283") == 0) {

initAspects(5, 6, 0, 6, 0);

printLCD('A', 1, aspect, " ", "MEDIUM", "CLEAR");

setLCDB(8, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED AT POSTED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1283A") == 0) {

initAspects(23, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED, APPROACHING",

"NEXT SIGNAL",

"NOT EXCEEDING MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1283B") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "SLOW");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS,",

" ",

"AND OVER POWER-",

"OPERATED SWITCHES;",

" ",

" ",

"THEN PROCEED,",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"SLOW SPEED.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1283C") == 0) {

initAspects(25, 0, 0, 22, 0);

printLCD('A', 1, aspect, "MEDIUM", "ADVANCE", "APPROACH");

setLCDB(8, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED, PREPARED TO",

"STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1284") == 0) {

initAspects(26, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING SLOW SPEED", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1285") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED,",

"AS SOON AS THE",

"ENGINE PASSES THE",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1286") == 0) {

initAspects(28, 29, 0, 29, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(8, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED, PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1287") == 0) {

initAspects(30, 0, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED AT POSTED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

pages = 2;

} else if (strcmp(aspect, "CSX 1288") == 0) {

initAspects(31, 0, 0, 8, 9);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED, PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1290") == 0) {

initAspects(32, 36, 37, 36, 37);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1291") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "RESTRICTED", "PROCEED");

setLCDB(4, "No. PLATE OR 'P'",

"PLAQUE REQUIRED:",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1292") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

"",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1293") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP AND", "CHECK");

setLCDB(16, "'C' PLAQUE REQ'D:",

"STOP AND CHECK",

"POSITION OF",

"DRAWBRIDGE,",

"SPRING SWITCH,",

"DERAILS OR GATES",

"PROTECTING RAILROAD",

"CROSSINGS;",

"IF WAY IS CLEAR AND",

"DRAWBRIDGE, SPRING",

"SWITCH, DERAILS OR",

"GATE ARE IN PROPER",

"POSITION, PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1294") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP AND", "OPEN SWITCH");

setLCDB(4, "ILLUMINATED 'S'",

"MARKER REQUIRED:",

"STOP AND OPEN HAND-",

"OPERATED SWITCH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX 1298") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "GRADE", "");

setLCDB(4, "'G' PLAQUE REQ'D:",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1281") == 0) {

initAspects(41, 2, 3, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED",

"",

"\*DWARF REQUIRES",

"NUMBER PLATE", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

pages = 1;

} else if (strcmp(aspect, "CSX C1281B") == 0) {

initAspects(47, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "", "");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1281C") == 0) {

initAspects(64, 16, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(8, "LIMITED SPEED",

"THROUGH TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED AT",

"POSTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1281D") == 0) { // TYPO

initAspects(68, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED", "APPROACH");

setLCDB(8, "LIMITED SPEED",

"THROUGH TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED,",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1282") == 0) {

initAspects(80, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1283") == 0) {

initAspects(81, 6, 0, 0, 51);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(8, "MEDIUM SPEED THROUGH",

"TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED AT",

"POSTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1283A") == 0) {

initAspects(23, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED,",

"",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1283B") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "SLOW");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS,",

"CROSSOVERS, SIDINGS,",

"AND",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED,",

"",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"SLOW SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1284") == 0) {

initAspects(46, 22, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING SLOW SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1285") == 0) {

initAspects(82, 8, 9, 0, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL.",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED;",

"WHEN THE ENGINE",

"PASSES THE SIGNAL",

" ",

"No. PLATE REQUIRED", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1286") == 0) {

initAspects(25, 0, 0, 0, 27);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND",

" ",

"OVER POWER-OPERATED",

"SWITCHES;",

"THEN PROCEED,",

" ",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1287") == 0) {

initAspects(30, 0, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED ",

"SWITCHES; THEN",

"PROCEED AT POSTED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1288") == 0) {

initAspects(28, 0, 0, 8, 0);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND OVER",

"POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED, PREPARED TO",

"STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1290") == 0) {

initAspects(40, 29, 0, 29, 37);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1291") == 0) {

initAspects(39, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "RESTRICTED", "PROCEED");

setLCDB(4, "No. PLATE OR 'P'",

"PLAQUE REQUIRED:",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1292") == 0) {

initAspects(39, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX C1298") == 0) {

initAspects(39, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "GRADE", "");

setLCDB(4, "'G' PLAQUE REQUIRED:",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1281") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1281A") == 0) { // TYPO

initAspects(53, 54, 51, 0, 0);

printLCD('A', 1, aspect, "", "CAB SPEED", "");

setLCDB(12, "PROCEED IN",

"ACCORDANCE WITH CAB",

"SIGNAL INDICATION;",

" ",

"REDUCE SPEED TO NOT",

"EXCEEDING 60 MPH IF",

"CAB SPEED CAB SIGNAL",

" ",

"IS DISPLAYED WITHOUT",

"A SIGNAL SPEED, OR",

"IF CAB SIGNALS ARE",

"NOT OPERATIVE.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1281B") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1281C") == 0) {

initAspects(15, 16, 0, 54, 51);

printLCD('A', 1, aspect, "", "", "");

setLCDB(8, "LIMITED SPEED",

"THROUGH TURNOUTS,",

"CROSSOVERS, SIDINGS,",

" ",

"AND OVER POWER-",

"OPERATED SWITCHES;",

"THEN PROCEED AT",

"POSTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1282") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1282A") == 0) {

initAspects(0, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(12, "PROCEED, PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL.",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST",

"BEGIN REDUCTION TO",

" ",

"LIMITED SPEED AS",

"SOON AS LOCOMOTIVE",

"PASSES THE SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

setLCDB(4, "ERROR:",

"No matrix was setup",

"for this indication.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1283") == 0) {

initAspects(5, 6, 0, 61, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(8, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND",

" ",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED AT POSTED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1283A") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND ",

" ",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED,",

" ",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1284") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(12, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING SLOW",

"SPEED.",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO ",

" ",

"MEDIUM SPEED AS SOON",

"AS THE LOCOMOTIVE",

"PASSES THE SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1285") == 0) {

initAspects(7, 8, 0, 0, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL.",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

" ",

"MEDIUM SPEED AS SOON",

"AS THE ENGINE PASSES",

"THE SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1286") == 0) {

initAspects(28, 18, 0, 43, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APROACH");

setLCDB(12, "MEDIUM SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND",

" ",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED,",

" ",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1287") == 0) {

initAspects(30, 0, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND",

" ",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED AT POSTED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1288") == 0) {

initAspects(44, 0, 0, 8, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(12, "SLOW SPEED THROUGH",

"TURNOUTS, CROSSOVERS",

"SIDINGS, AND",

" ",

"OVER POWER-OPERATED",

"SWITCHES; THEN",

"PROCEED,",

" ",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1290 Y") == 0) { //TYPO - ?

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(16, "PROCEED AT",

"RESTRICTED SPEED",

"UNTIL THE ENTIRE",

"TRAIN HAS CLEARED",

"ALL SWITCHES (IF",

"THE SIGNAL IS A",

"CONTROLLED POINT",

"SIGNAL), AND",

" ",

"THE LEADING WHEELS",

"HAVE PASSED A MORE",

"FAVORABLE SIGNAL, OR",

" ",

"ENTERED NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1290 L") == 0) { // TYPO - ?

initAspects(32, 36, 37, 0, 37);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(16, "PROCEED AT",

"RESTRICTED SPEED",

"UNTIL THE ENTIRE",

"TRAIN HAS CLEARED",

"ALL SWITCHES (IF",

"THE SIGNAL IS A",

"CONTROLLED POINT",

"SIGNAL), AND",

" ",

"THE LEADING WHEELS",

"HAVE PASSED A MORE",

"FAVORABLE SIGNAL, OR",

" ",

"ENTERED NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1291") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "RESTRICTED", "PROCEED");

setLCDB(4, "No. PLATE REQUIRED:",

" ",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CSX CR1292") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 803") == 0) {

initAspects(1, 2, 3, 42, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 804") == 0) {

initAspects(55, 56, 27, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(4, "PROCEED PREPARED",

"TO STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 805") == 0) {

initAspects(7, 8, 9, 0, 27);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(4, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 806") == 0) {

initAspects(104, 43, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "RESTRICTING");

setLCDB(4, "PROCEED PREPARED TO",

"PASS NEXT SIGNAL AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 807") == 0) {

initAspects(0, 54, 51, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "APPROACH", "DIVERGING");

setLCDB(4, "PROCEED PREPARED TO",

"ENTER DIVERGING AT",

"SECOND SIGNAL AT",

"PRESCRIBED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 808") == 0) {

initAspects(19, 20, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "DIVERGING");

setLCDB(8, "PROCEED, PREPARED TO",

"ENTER DIVERGING",

"ROUTE AT NEXT",

"SIGNAL,",

"AT PRESCRIBED SPEED.",

"PROCEED PREPARED TO",

"STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 809") == 0) {

initAspects(5, 6, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 810") == 0) {

initAspects(15, 16, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "CLEAR APPROACH", "DIVERGING");

setLCDB(12, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED PREPARED TO",

"ENTER DIVERGING",

"ROUTE AT NEXT SIGNAL",

"AT PRESCRIBED SPEED.",

" ",

" ",

"PROCEED PREPARED TO",

"STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 811") == 0) {

initAspects(17, 18, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "ADVANCE", "APPROACH");

setLCDB(8, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED PREPARED",

" ",

"TO STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 812") == 0) {

initAspects(25, 29, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "APPROACH");

setLCDB(8, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED PREPARED",

" ",

"TO STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 813") == 0) {

initAspects(113, 114, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "RESTRICTING");

setLCDB(8, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED PREPARED",

" ",

"TO PASS NEXT SIGNAL",

"AT RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 814") == 0) {

initAspects(115, 116, 79, 0, 0);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 815") == 0) {

initAspects(10, 11, 12, 120, 12);

printLCD('A', 1, aspect, "", "RESTRICTED", "PROCEED");

setLCDB(4, "No. PLATE REQUIRED:",

" ",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 816") == 0) {

initAspects(10, 11, 12, 120, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 817") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 818") == 0) {

initAspects(55, 56, 27, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(4, "PROCEED, PREPARED TO",

"STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 818.1") == 0) {

initAspects(25, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "ADVANCE", "APPROACH");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 25 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"PREPARED TO STOP AT",

"SECOND SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 818.2") == 0) {

initAspects(70, 0, 0, 0, 0);

printLCD('A', 1, aspect, "LIMITED", "ADVANCE", "APPROACH");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 40 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"PREPARED TO STOP AT",

"SECOND SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 819") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(4, "PROCEED, PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 820") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 40 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 821") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 25 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 822") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(4, "PROCEED, APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 15 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 823") == 0) {

initAspects(15, 0, 0, 54, 0);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(4, "PROCEED NOT",

"EXCEEDING 40 MPH",

"THROUGH TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 824") == 0) {

initAspects(65, 0, 0, 0, 0);

printLCD('A', 1, aspect, "LIMITED", "CLEAR", "LIMITED");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 40 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"40 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 825") == 0) {

initAspects(17, 0, 0, 56, 0);

printLCD('A', 1, aspect, "", "LIMITED", "APPROACH");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 40 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 826") == 0) {

initAspects(5, 0, 0, 2, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(4, "PROCEED NOT",

"EXCEEDING 25 MPH",

"THROUGH TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 827") == 0) {

initAspects(107, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "CLEAR", "MEDIUM");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 25 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"APPROACHING NEXT",

"SIGNAL NOT EXCEEDING",

"25 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 828") == 0) {

initAspects(28, 0, 0, 8, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 25 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 829") == 0) {

initAspects(30, 6, 0, 6, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(4, "PROCEED NOT",

"EXCEEDING 15 MPH",

"THROUGH TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 830") == 0) {

initAspects(44, 18, 0, 18, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "PROCEED NOT",

"EXCEEDING 15 MPH",

"THROUGH TURNOUTS,",

" ",

"THEN PROCEED",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 831") == 0) {

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "No. PLATE REQUIRED:",

" ",

"PROCEED AT",

"RESTRICTIED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 832") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 833") == 0) {

initAspects(0, 2, 3, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 834") == 0) { // TYPO

initAspects(0, 50, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED`");

setLCDB(4, "PROCEED APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 35 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 835") == 0) {

initAspects(0, 49, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED CLEAR", "");

setLCDB(4, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 35 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 836") == 0) {

initAspects(0, 20, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 30 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 837") == 0) {

initAspects(0, 16, 0, 0, 51);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(4, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 30 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 838") == 0) {

initAspects(0, 6, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(8, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 30 MPH.",

" ",

"APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING 30 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 839") == 0) {

initAspects(0, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(4, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 840") == 0) {

initAspects(0, 18, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(8, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 30 MPH.",

" ",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 841") == 0) {

initAspects(0, 0, 0, 0, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(4, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 20 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 842") == 0) {

initAspects(0, 0, 0, 0, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "PROCEED THROUGH",

"TURNOUTS NOT",

"EXCEEDING 20 MPH.",

" ",

"PREPARED TO STOP AT",

"THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 843") == 0) {

initAspects(0, 29, 12, 0, 79);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CN 844") == 0) {

initAspects(0, 11, 0, 0, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 230") == 0) {

initAspects(0, 2, 3, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 231") == 0) {

initAspects(0, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(4, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 232") == 0) {

initAspects(0, 6, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 233") == 0) {

initAspects(0, 29, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "APPROACH");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 234") == 0) {

initAspects(0, 35, 37, 0, 0);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "LS&I 235") == 0) {

initAspects(0, 11, 12, 0, 0);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.3") == 0) {

initAspects(1, 2, 3, 2, 3);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.4") == 0) {

initAspects(0, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(8, "PROCEED PREPARED TO",

"PASS THE NEXT SIGNAL",

"NOT EXCEEDING",

"60 MPH, AND",

"BE PREPARED TO ENTER",

"DIVERGING ROUTE AT",

"PRESCRIBED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.5") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(8, "PROCEED PREPARED TO",

"PASS NEXT SIGNAL",

"NOT EXCEEDING",

"50 MPH, AND",

"BE PREPARED TO ENTER",

"DIVERGING ROUTE AT",

"PRESCRIBED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.6") == 0) {

initAspects(55, 56, 27, 56, 27);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(8, "PROCEED PREPARED TO",

"PASS NEXT SIGNAL",

"NOT EXCEEDING",

"40 MPH, AND",

"BE PREPARED TO ENTER",

"DIVERGING ROUTE AT",

"PRESCRIBED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.7") == 0) {

initAspects(106, 105, 0, 105, 0);

printLCD('A', 1, aspect, "", "APPROACH", "RESTRICTING");

setLCDB(4, "PROCEED PREPARED TO",

"PASS NEXT SIGNAL AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.8") == 0) {

initAspects(7, 8, 9, 8, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(8, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL,",

" ",

" ",

"TRAINS EXCEEDING 30",

"MPH MUST IMMEDIATELY",

"REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.9") == 0) {

initAspects(5, 6, 0, 6, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT(S)", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.10") == 0) {

initAspects(25, 0, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "DIVERGING");

setLCDB(0, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.11") == 0) {

initAspects(17, 18, 0, 18, 0);

printLCD('A', 1, aspect, "", "", "");

setLCDB(12, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT,",

"PREPARED TO ADVANCE",

"ON DIVERGING ROUTE",

"AT THE NEXT SIGNAL,",

" ",

"NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.11") == 0) {

initAspects(0, 0, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "MEDIUM");

setLCDB(8, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT,",

"PREPARED TO PASS",

"NEXT SIGNAL NOT",

"EXCEEDING 35 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

setLCDB(4, "ERROR:",

"This matrix doesn't",

"appear to be",

"programmed in.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.12") == 0) {

initAspects(28, 29, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "APPROACH");

setLCDB(12, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT,",

"AND APPROACH NEXT",

"SIGNAL PREPARING TO",

"STOP. IF EXCEEDING",

"30 MPH,",

"IMMEDIATELY REDUCE",

"TO THAT SPEED. NOTE",

"SPEED IS 40 MPH FOR",

"PASSENGER TRAINS.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.13") == 0) {

initAspects(115, 116, 79, 116, 79);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "No. PLATE OR 'G'",

"PLAQUE REQUIRED:",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.14") == 0) {

initAspects(0, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

setLCDB(4, "ERROR:",

"This matrix doesn't",

"appear to be",

"programmed in.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "BNSF 9.1.15") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP.", "");

setLCDB(4, "STOP.",

"",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.1") == 0) {

initAspects(1, 2, 3, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.2") == 0) {

initAspects(55, 56, 25, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST",

"BEGIN REDUCTION TO",

"LIMITED SPEED,",

"AS SOON AS THE",

"ENGINE PASSES THE",

"ADVANCE APPROACH",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.3") == 0) {

initAspects(21, 22, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(12, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"SLOW SPEED;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED,",

"AS SOON AS THE",

"ENGINE PASSES THE",

"APPROACH SLOW",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.4") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED,",

"AS SOON AS THE",

"ENGINE PASSES THE",

"APPROACH SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.5") == 0) {

initAspects(0, 0, 0, 0, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(12, "PROCEED AT SLOW",

"SPEED,",

" ",

" ",

"UNTIL THE ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"MAXIMUM AUTHORIZED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.6") == 0) {

initAspects(0, 0, 0, 0, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL;",

" ",

" ",

"SLOW SPEED APPLIES",

"UNTIL ENTIRE TRAIN",

"CLEARS ALL",

" ",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN MEDIUM SPEED",

"APPLIES.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.7") == 0) {

initAspects(31, 29, 0, 0, 9);

printLCD('A', 1, aspect, "", "", "");

setLCDB(12, "PROCEED AT",

"RESTRICTED SPEED",

"UNTIL THE ENTIRE",

"TRAIN HAS CLEARED,",

"ALL INTERLOCKING AND",

"SPRING SWITCHES AND",

"THE LEADING WHEELS",

"HAVE PASSED A MORE",

"FAVORABLE FIXED",

"SIGNAL OR ENTERED",

"NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.8") == 0) {

initAspects(10, 11, 12, 0, 0);

printLCD('A', 1, aspect, "", "", "");

setLCDB(16, "STOP, THEN PROCEED",

"AT RESTRICTED SPEED",

"UNTIL THE ENTIRE",

"TRAIN HAS CLEARED",

"ALL INTERLOCKING AND",

"SPRING SWITCHES AND",

"THE LEADING WHEELS",

"HAVE PASSED A MORE",

"FAVORABLE FIXED",

"SIGNAL OR ENTERED",

"NON-SIGNALED",

"TERRITORY.",

"No. PLATE REQUIRED", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.9") == 0) {

initAspects(10, 11, 12, 0, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.10") == 0) {

initAspects(19, 20, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(4, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.11") == 0) {

initAspects(13, 14, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(4, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"LIMITED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.12") == 0) {

initAspects(5, 6, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"MAXIMUM AUTHORIZE",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.13") == 0) {

initAspects(15, 16, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"MAXIMUM AUTHORIZED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "AMTK 9.1.14") == 0) {

initAspects(28, 18, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED AS SOON",

"AS THE MEDIUM",

"APPROACH MEDIUM",

"SIGNAL IS CLEARLY",

"VISIBLE.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 306") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED AT",

"AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 307") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(4, "PROCEED APPROACHING",

"THE NEXT SIGNAL NOT",

"EXCEEDING LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 308") == 0) {

initAspects(15, 16, 0, 54, 51);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT, OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 309") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 310") == 0) {

initAspects(55, 56, 27, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST",

"BEGIN REDUCTION TO",

"LIMITED SPEED,",

"AS SOON AS ENGINE",

"PASSES THE SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 311") == 0) {

initAspects(5, 6, 0, 61, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT, OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 312") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(20, "PROCEED AT MEDIUM",

"SPEED...",

" ",

" ",

"UNTIL ENTIRE TRAIN",

"CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT,",

"OR SPRING SWITCHES,",

"THEN APPROACH THE",

"NEXT SIGNAL AT",

"MEDIUM SPEED;",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED AS",

"SOON AS THE MEDIUM",

"APPROACH MEDIUM",

"SIGNAL IS CLEARLY",

"VISIBLE.", "", "", "", "");

} else if (strcmp(aspect, "NS 313") == 0) {

initAspects(26, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(8, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"SLOW SPEED;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 314") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(8, "PROCEED PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 315") == 0) {

initAspects(28, 18, 0, 43, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED,",

"AS SOON AS THE",

"MEDIUM APPROACH",

"SIGNAL IS CLEARLY",

"VISIBLE.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 316") == 0) {

initAspects(30, 0, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "PROCEED AT SLOW",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT, OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 317") == 0) {

initAspects(44, 0, 0, 8, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL;",

" ",

" ",

"SLOW SPEED APPLIES",

"UNTIL ENTIRE TRAIN",

"CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT, OR",

"SPRING SWITCHES,",

"THEN MEDIUM SPEED",

"APPLIES.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 318") == 0) {

initAspects(31, 29, 12, 0, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(16, "PROCEED AT",

"RESTRICTING SPEED",

"UNTIL ENTIRE TRAIN",

"CLEARS ALL",

"INTERLOCKING,",

"CONTROLLED POINT, OR",

"SPRING SWITCHES, AND",

"THE LEADING END ",

"HAS EITHER PASSED A",

"MORE FAVORABLE FIXED",

"SIGNAL,",

" ",

"OR ENTERED RULE 171",

"(NON-SIGNALED)",

"TERRITORY",

"No. PLATE REQUIRED", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 319") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 326") == 0) {

initAspects(1, 2, 3, 2, 3);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED AT",

"AUTHORIZED SPEED. ",

"DWARF:",

"No. PLATE REQUIRED", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 327") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "DIVERGING");

setLCDB(8, "PROCEED PREPARING TO",

"TAKE DIVERGING ROUTE",

" ",

" ",

"BEYOND NEXT SIGNAL",

"AT AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 328") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(4, "PROCEED PREPARING TO",

"STOP AT THE SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 329") == 0) {

initAspects(5, 6, 0, 6, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(8, "PROCEED THROUGH",

"DIVERGING ROUTE,",

"OBSERVING AUTHORIZED",

" ",

"SPEED THROUGH",

"TURNOUT(S) OR",

"CROSSOVER(S).", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 330") == 0) {

initAspects(17, 127, 0, 127, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "DIVERGING");

setLCDB(12, "PROCEED THROUGH",

"TURNOUT(S) OR",

"CROSSOVER(S) AT ",

" ",

"AUTHORIZED SPEED,",

"PREPARING TO TAKE",

"DIVERGING ROUTE ",

" ",

"BEYOND NEXT SIGNAL",

"AT AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 331") == 0) {

initAspects(7, 8, 9, 0, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(8, "PROCEED PREPARING TO",

"STOP AT NEXT SIGNAL;",

" ",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 333") == 0) {

initAspects(28, 128, 0, 128, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "APPROACH");

setLCDB(12, "PROCEED THROUGH",

"DIVERGING ROUTE,",

"OBSERVING AUTHORIZED",

"SPEED THROUGH",

"TURNOUT(S) AND",

"CROSSOVER(S)",

"PREPARING TO STOP AT",

"NEXT SIGNAL;",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE BEGIN REDUCTION",

"TO THAT SPEED.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 334") == 0) {

initAspects(30, 0, 0, 0, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "PROCEED; SLOW SPEED",

"WITHIN CONTROLLED",

"POINT/INTERLOCKING",

"LIMITS,",

"OR THROUGH",

"TURNOUT(S) OR",

"CROSSOVER(S).", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 335") == 0) {

initAspects(0, 0, 0, 8, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "PROCEED PREPARING TO",

"STOP AT NEXT SIGNAL;",

"SLOW SPEED WITHIN",

"CONTROLLED",

"POINT/INTERLOCKING",

"LIMITS OR THROUGH",

"TURNOUT(S) OR",

"CROSSOVER(S).", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 336") == 0) { // TYPO

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(8, "PROCEED AT",

"RESTRICTED SPEED,",

"UNTIL THE LEADING",

"END EITHER PASSES",

"A MORE FAVORABLE",

"FIXED SIGNAL, OR",

"ENTERS NON-SIGNALED",

"TRACK.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 337") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

"",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 340") == 0) {

initAspects(1, 2, 3, 2, 3);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED AT",

"AUTHORIZED SPEED. ",

"DWARF:",

"No. PLATE REQUIRED", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 341") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "DIVERGING");

setLCDB(4, "PROCEED PREPARING TO",

"TAKE DIVERGING ROUTE",

"AT NEXT SIGNAL AT",

"AUTHORIZED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 342") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(4, "PROCEED PREPARING TO",

"STOP AT SECOND",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 343") == 0) {

initAspects(5, 6, 0, 6, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(8, "PROCEED THROUGH",

"DIVERGING ROUTE,",

" ",

" ",

"OBSERVING AUTHORIZED",

"SPEED THROUGH",

"TURNOUT(S) AND",

"CROSSOVER(S).", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 344") == 0) {

initAspects(55, 56, 27, 56, 27);

printLCD('A', 1, aspect, "", "APPROACH", "RESTRICTED");

setLCDB(12, "PROCEED, APPROACHING",

"NEXT SIGNAL AT",

"RESTRICTED SPEED,",

" ",

"NOT EXCEEDING 15",

"MPH;",

" ",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 345") == 0) {

initAspects(17, 18, 0, 18, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "RESTRICTED");

setLCDB(16, "PROCEED THROUGH",

"DIVERGING ROUTE,",

" ",

" ",

"OBSERVING AUTHORIZED",

"SPEED THROUGH",

"TURNOUT(S) AND",

"CROSSOVER(S),",

"APPROACHING NEXT",

"SIGNAL AT RESTRICTED",

"SPEED, NOT EXCEEDING",

"15 MPH;",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE BEGIN REDUCTION",

"TO THAT SPEED.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 346") == 0) { // TYPO

initAspects(7, 8, 9, 8, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(8, "PROCEED PREPARING TO",

"STOP AT NEXT SIGNAL;",

" ",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 347") == 0) {

initAspects(28, 29, 0, 29, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "APPROACH");

setLCDB(12, "PROCEED ONTO",

"DIVERGING ROUTE,",

"OBSERVING AUTHORIZED",

"SPEED THROUGH",

"TURNOUT(S) AND",

"CROSSOVER(S),",

"PREPARING TO STOP AT",

"THE NEXT SIGNAL;",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST AT",

"ONCE REDUCE TO THAT",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 348") == 0) { // TYPO?

initAspects(31, 29, 12, 0, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(12, "PROCEED AT",

"RESTRICTED SPEED;",

" ",

" ",

"RESTRICTED SPEED",

"MUST BE OBSERVED",

"UNTIL THE LEADING",

"WHEELS REACH",

"A MORE FAVORABLE",

"FIXED SIGNAL, OR",

"ENTER NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NS 349") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 281") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED AT MAXIMUM",

"TRACK SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 281A") == 0) {

initAspects(48, 49, 0, 49, 0);

printLCD('A', 1, aspect, "ADVANCE", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED APPROACHING",

"SECOND SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 281B") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(16, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"LIMITED SPEED;",

" ",

"TRAINS EXCEEDING 60",

"MPH MUST AT ONCE",

"REDUCE TO THAT",

"SPEED;",

"REDUCTION TO 60 MPH",

"MUST COMMENCE BEFORE",

"PASSING SIGNAL, AND",

" ",

"MUST BE COMPLETED",

"BEFORE ACCEPTING A",

"MORE FAVORABLE",

"INDICATION.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 281C") == 0) {

initAspects(15, 16, 0, 16, 51);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(4, "PROCEED. LIMITED",

"SPEED APPLIES",

"THROUGH INTERLOCKING",

"LIMITS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 281D") == 0) {

initAspects(17, 18, 0, 18, 27);

printLCD('A', 1, aspect, "", "LIMITED", "APPROACH");

setLCDB(4, "PROCEED AT LIMITED",

"SPEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 282") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(16, "PROCEED, APPROACHING",

"NEXT SIGNAL AT",

"MEDIUM SPEED;",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST,",

"AT ONCE, REDUCE TO",

"THAT SPEED;",

"REDUCTION TO LIMITED",

"SPEED MUST COMMENCE",

"BEFORE PASSING",

"SIGNAL, AND",

"MUST BE COMPLETED",

"BEFORE ACCEPTING A",

"MORE-FAVORABLE",

"INDICATION.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 282A") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(16, "PROCEED PREPARING TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST,",

"AT ONCE, REDUCE TO",

"THAT SPEED;",

"REDUCTION TO LIMITED",

"SPEED MUST COMMENCE",

"BEFORE PASSING",

"SIGNAL, AND",

"MUST BE COMPLETED",

"BEFORE ACCEPTING A",

"MORE-FAVORABLE",

"INDICATION.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 283") == 0) {

initAspects(5, 6, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(4, "PROCEED. MEDIUM",

"SPEED WITHIN",

"INTERLOCKING LIMITS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 283A") == 0) {

initAspects(25, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "ADVANCE", "APPROACH");

setLCDB(8, "PROCEED PREPARING TO",

"STOP AT SECOND",

"SIGNAL;",

" ",

"MEDIUM SPEED WITHIN",

"INTERLOCKING LIMITS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 283B") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "SLOW");

setLCDB(4, "PROCEED AT MEDIUM",

"SPEED, APPROACHING",

"NEXT SIGNAL AT SLOW",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 284") == 0) {

initAspects(26, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(8, "PROCEED, APPROACHING",

"NEXT SIGNAL AT",

"SLOW SPEED;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST,",

"AT ONCE, REDUCE TO",

"THAT SPEED;", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 285") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(16, "PROCEED PREPARED TO",

"TO STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST,",

"AT ONCE, REDUCE TO",

"THAT SPEED;",

"REDUCTION TO MEDIUM",

"SPEED MUST COMMENCE",

"BEFORE PASSING",

"SIGNAL, AND",

"MUST BE COMPLETED",

"BEFORE ACCEPTING A",

"MORE-FAVORABLE",

"INDICATION.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 286") == 0) {

initAspects(28, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(4, "PROCEED AT MEDIUM",

"SPEED PREPARED TO",

"STOP AT NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 287") == 0) {

initAspects(30, 6, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(4, "PROCEED. SLOW SPEED",

"WITHIN INTERLOCKING",

"LIMITS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 288") == 0) {

initAspects(0, 0, 0, 8, 0);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(8, "PROCEED PREPARED TO",

"STOP AT THE NEXT",

"SIGNAL;",

" ",

"SLOW SPEED WITHIN",

"INTERLOCKING LIMITS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 290") == 0) {

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 291") == 0) {

initAspects(10, 11, 12, 0, 0);

printLCD('A', 1, aspect, "", "STOP AND", "PROCEED");

setLCDB(8, "STOP BEFORE PASSING",

"SIGNAL. THEN PROCEED",

"AT RESTRICTED SPEED;",

" ",

"REQUIRES OFFSET",

"HEADS AND/OR",

"NUMBER PLATE.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NYC 292") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP BEFORE PASSING",

"SIGNAL.",

" ",

"ABS. SIGNALS ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.1") == 0) {

initAspects(1, 2, 3, 2, 3);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.2") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "CLEAR SIXTY");

setLCDB(16, "PROCEED. FREIGHT",

"TRAINS EXCEEDING 60",

"MPH MUST IMMEDIATELY",

"REDUCE TO 60 MPH;",

"PASSENGER TRAINS MAY",

"PROCEED, BUT BE",

"PREPARED TO PASS THE",

"NEXT SIGNAL NOT",

"EXCEEDING 60 MPH.",

"WHEN SIGNAL GOVERNS",

"THE APPROACH TO A",

"CONTROL POINT WITH",

"A 60 MPH TURNOUT",

"SPEED BE PREPARED TO",

"ADVANCE ON DIVERGING",

"ROUTE.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.3") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "CLEAR FIFTY");

setLCDB(16, "PROCEED. FREIGHT",

"TRAINS EXCEEDING 50",

"MPH MUST IMMEDIATELY",

"REDUCE TO 50 MPH;",

"PASSENGER TRAINS MAY",

"PROCEED, BUT BE",

"PREPARED TO PASS THE",

"NEXT SIGNAL NOT",

"EXCEEDING 50 MPH.",

"WHEN SIGNAL GOVERNS",

"THE APPROACH TO A",

"CONTROL POINT WITH",

"A 50 MPH TURNOUT",

"SPEED BE PREPARED TO",

"ADVANCE ON DIVERGING",

"ROUTE.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.4") == 0) { // TYPO

initAspects(55, 56, 27, 56, 27);

printLCD('A', 1, aspect, "", "ADVANCE", "APPROACH");

setLCDB(20, "PROCEED. PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;"

" ",

"TRAINS EXCEEDING 40",

"MPH MUST IMMEDIATELY",

"REDUCE TO 40 MPH;",

"PASSENGER ONES MAY",

"PROCEED, BUT MUST BE",

"PREPARED TO PASS THE",

"NEXT SIGNAL NOT",

"EXCEEDING 40 MPH.",

"WHEN SIGNAL GOVERNS",

"THE APPROACH TO A",

"CONTROL POINT WITH",

"A 40 MPH TURNOUT",

"SPEED BE PREPARED TO",

"ADVANCE NORMAL OR",

"DIVERGING ROUTE.", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.4P") == 0) {

initAspects(55, 56, 27, 56, 27);

printLCD('A', 1, aspect, "ADVANCE", "APPROACH", "PASSENGER");

setLCDB(16, "PROCEED. PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;"

" ",

"TRAINS EXCEEDING 40",

"MPH MUST IMMEDIATELY",

"REDUCE TO 40 MPH;",

"PASSENGER ONES MAY",

"PROCEED, BUT MUST BE",

"PREPARED TO PASS THE",

"NEXT SIGNAL NOT",

"EXCEEDING 60 MPH;",

"THIS INDICATION",

"REQUIRES A 'C'",

"COMMUTER PLAQUE.",

"", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.5") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "APPROACH", "DIVERGING");

setLCDB(8, "PROCEED PREPARED TO",

"ADVANCE ON DIVERGING",

"ROUTE AT NEXT",

"SIGNAL,",

"AT PRESCRIBED SPEED",

"THROUGH TURNOUT.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.6") == 0) {

initAspects(7, 8, 9, 8, 9);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED TO",

"STOP BEFORE ANY PART",

"OF THE TRAIN PASSES",

"THE NEXT SIGNAL;",

"FREIGHT TRAINS",

"EXCEEDING 30 MPH",

"MUST IMMEDIATELY",

"REDUCE TO 30 MPH;",

"PASSENGER TRAINS",

"EXCEEDING 40 MPH",

"MUST IMMEDIATELY",

"REDUCE TO 40 MPH.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.7") == 0) {

initAspects(106, 105, 0, 105, 0);

printLCD('A', 1, aspect, "", "APPROACH", "RESTRICTING");

setLCDB(20, "PROCEED PREPARED TO",

"PASS NEXT SIGNAL AT",

"RESTRICTED SPEED,",

" ",

"BUT NOT EXCEEDING 15",

"MPH; WHEN THE NEXT",

"SIGNAL IS SEEN TO",

"DISPLAY",

"A PROCEED",

"INDICATION, THE ",

"REQUIREMENT TO PASS",

"THE NEXT SIGNAL",

"AT RESTRICTED SPEED",

"NO LONGER APPLIES.",

" ",

" ",

"SPEED MAY BE RESUMED",

"ONCE LEADING WHEELS",

"OF TRAIN HAVE PASSED",

"THE NEXT SIGNAL.", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.8") == 0) {

initAspects(15, 16, 0, 16, 0);

printLCD('A', 1, aspect, "DIVERGING", "CLEAR", "LIMITED");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE. SPEED THROUGH",

"TURNOUT MUST NOT",

"EXCEED 40 MPH.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.9") == 0) {

initAspects(5, 6, 0, 6, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "CLEAR");

setLCDB(4, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.10") == 0) {

initAspects(17, 18, 0, 18, 0);

printLCD('A', 1, aspect, "DIVERGING", "ADVANCE", "APPROACH");

setLCDB(24, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT",

"AND BE PREPARED TO",

"STOP AT SECOND",

"SIGNAL. FREIGHT",

"TRAINS EXCEEDING",

"40 MPH MUST",

"IMMEDIATELY REDUCE",

"TO 40 MPH. PASSENGER",

"TRAINS MAY PROCEED,",

"BUT MUST BE PREPARED",

"TO PASS NEXT SIGNAL",

"NOT EXCEEDING 40",

"MPH. WHEN SIGNAL",

"GOVERNS THE APPROACH",

"TO A CONTROL POINT",

"WITH A 40 MPH",

"TURNOUT SPEED,",

"BE PREPARED TO",

"ADVANCE ON NORMAL OR",

"DIVERGING ROUTE.", "");

} else if (strcmp(aspect, "UP 9.2.10P") == 0) {

initAspects(17, 18, 0, 18, 0);

printLCD('A', 1, aspect, "", "", "");

setLCDB(16, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED THROUGH",

"TURNOUT PREPARED",

"TO STOP AT SECOND",

"SIGNAL. FREIGHT",

"TRAINS EXCEEDING 40",

"MPH MUST",

"IMMEDIATELY REDUCE",

"TO 40 MPH. PASSENGER",

"TRAINS EXCEEDING 60",

"MPH MUST",

"IMMEDIATELY REDUCE",

"TO 60 MPH;",

"REQUIRES 'C'",

"COMMUTER PLAQUE.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.11") == 0) {

initAspects(28, 29, 0, 29, 0);

printLCD('A', 1, aspect, "", "", "");

setLCDB(16, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED THROUGH",

"TURNOUT PREPARED",

"TO STOP BEFORE ANY",

"PART OF TRAIN OR",

"ENGINE PASSES THE",

"NEXT SIGNAL;",

"FREIGHT TRAINS",

"EXCEEDING 30 MPH",

"MUST IMMEDIATELY",

"REDUCE TO 30 MPH;",

"PASSENGER TRAINS",

"EXCEEDING 40 MPH",

"MUST IMMEDIATELY",

"REDUCE TO 40 MPH.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.12") == 0) {

initAspects(25, 0, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "DIVERGING");

setLCDB(12, "PROCEED ON DIVERGING",

"ROUTE NOT EXCEEDING",

"PRESCRIBED SPEED",

"THROUGH TURNOUT",

"PREPARED TO ADVANCE",

"ON DIVERGING ROUTE",

"AT THE NEXT SIGNAL ",

" ",

"AT PRESCRIBED SPEED",

"THROUGH THE TURNOUT.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.13") == 0) { // TYPO

initAspects(33, 35, 37, 35, 37);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(8, "PROCEED AT",

"RESTRICTED SPEED,",

"NOT EXCEEDING",

" ",

"PRESCRIBED SPEED",

"THROUGH TURNOUT WHEN",

"APPLICABLE.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.14") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "", "");

setLCDB(4, "No. PLATE REQUIRED:",

" ",

"PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.15") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP BEFORE ANY PART",

"OF TRAIN PASSES THE",

"SIGNAL;",

"ABS. SIGNALS ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "UP 9.2.16") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "APPROACH", "CLEAR FIFTY");

setLCDB(20, "PROCEED ON DIVERGING",

"ROUTE AT PRESCRIBED",

"SPEED THROUGH",

"TURNOUT;",

"FREIGHT TRAINS",

"EXCEEDING 50 MPH",

"MUST IMMEDIATELY",

"REDUCE TO 50 MPH;",

"PASSENGER TRAINS MAY",

"PROCEED, BUT MUST BE",

"PREPARED TO PASS THE",

"NEXT SIGNAL NOT",

"EXCEEDING 50 MPH.",

"WHEN SIGNAL GOVERNS",

"THE APPROACH TO A",

"CONTROL POINT WITH",

"A 50 MPH TURNOUT",

"SPEED, BE PREPARED",

"TO ADVANCE ON",

"DIVERGING ROUTE.", "", "", "", "");

} else if (strcmp(aspect, "CROR 405") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 406") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "CLEAR", "TO LIMITED");

setLCDB(4, "PROCEED, APPROACHING",

"THE NEXT SIGNAL AT",

"LIMITED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 407") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "CLEAR TO", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"THE NEXT SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 408") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "CLEAR TO", "DIVERGING");

setLCDB(4, "PROCEED, APPROACHING",

"THE NEXT SIGNAL AT",

"DIVERGING SPEED.",

"'DV' PLATE REQUIRED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 409") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "CLEAR TO", "SLOW");

setLCDB(4, "PROCEED, APPROACHING",

"THE NEXT SIGNAL AT",

"SLOW SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 410") == 0) {

initAspects(109, 43, 0, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR TO", "RESTRICTING");

setLCDB(4, "PROCEED. NEXT SIGNAL",

"IS DISPLAYING",

"RESTRICTING.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 411") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "CLEAR TO", "STOP");

setLCDB(4, "PROCEED, PREPARING",

"TO STOP AT THE NEXT",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 412") == 0) {

initAspects(96, 97, 0, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "CLEAR TO", "LIMITED");

setLCDB(4, "PROCEED, APPROACHING",

"SECOND SIGNAL AT",

"LIMITED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 413") == 0) {

initAspects(58, 95, 0, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "CLEAR TO", "MEDIUM");

setLCDB(4, "PROCEED, APPROACHING",

"SECOND SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 414") == 0) {

initAspects(99, 100, 0, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "CLEAR TO", "SLOW");

setLCDB(4, "PROCEED, APPROACHING",

"SECOND SIGNAL AT",

"SLOW SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 414A") == 0) {

initAspects(99, 100, 0, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "CLEAR TO", "DIVERGING");

setLCDB(4, "PROCEED, APPROACHING",

"SECOND SIGNAL AT",

"DIVERGING SPEED.",

"'DV' PLATE REQUIRED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 415") == 0) { // TYPO

initAspects(55, 56, 27, 0, 0);

printLCD('A', 1, aspect, "ADVANCE", "CLEAR TO", "STOP");

setLCDB(8, "PROCEED; NEXT SIGNAL",

"IS DISPLAYING CLEAR",

"TO STOP,",

" ",

"BE PREPARED TO STOP",

"AT SECOND SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 416") == 0) {

initAspects(15, 0, 0, 54, 0);

printLCD('A', 1, aspect, "", "LIMITED", "TO CLEAR");

setLCDB(4, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 417") == 0) {

initAspects(65, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "LIMITED");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 418") == 0) {

initAspects(66, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "MEDIUM");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 419") == 0) {

initAspects(67, 0, 0, 130, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "SLOW");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT SLOW",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 419A") == 0) {

initAspects(67, 0, 0, 130, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "DIVERGING");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT DIVERGING",

"SPEED.",

"\*NEEDS 'DV' PLATE.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 420") == 0) { // ?

initAspects(72, 131, 0, 101, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "RESTRICTING");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"NEXT SIGNAL IS",

"DISPLAYING",

"RESTRICTING.",

"\*NEEDS 'DV' PLATE.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 421") == 0) {

initAspects(17, 0, 0, 56, 0);

printLCD('A', 1, aspect, "", "LIMITED TO", "STOP");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"PREPARING TO STOP",

"AT THE NEXT SIGNAL", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 422") == 0) {

initAspects(5, 0, 0, 2, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "CLEAR");

setLCDB(4, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 423") == 0) {

initAspects(132, 0, 0, 133, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "LIMITED");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 424") == 0) {

initAspects(107, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "MEDIUM");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 425") == 0) {

initAspects(134, 0, 0, 50, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "SLOW");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT SLOW",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 425A") == 0) {

initAspects(134, 0, 0, 50, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "DIVERGING");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT DIVERGING",

"SPEED.",

"\*NEEDS 'DV' PLATE.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 426") == 0) {

initAspects(135, 61, 0, 43, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "RESTRICTING");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"NEXT SIGNAL IS",

"DISPLAYING",

"RESTRICTING.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 427") == 0) {

initAspects(28, 0, 0, 8, 0);

printLCD('A', 1, aspect, "", "MEDIUM TO", "STOP");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"PREPARING TO STOP",

"AT THE NEXT SIGNAL", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 428") == 0) {

initAspects(30, 6, 0, 6, 3);

printLCD('A', 1, aspect, "", "DIVERGING", "TO CLEAR");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT DIVERGING",

"SPEED,",

"PAST SIGNAL AND",

"THROUGH TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 429") == 0) {

initAspects(44, 18, 0, 18, 27);

printLCD('A', 1, aspect, "", "DIVERGING TO", "STOP");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT DIVERGING",

"SPEED,",

"PAST SIGNAL AND",

"THROUGH TURNOUTS.",

"PREPARING TO STOP",

"AT THE NEXT SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 430") == 0) {

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "DIVERGING", "");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT REDUCED",

"SPEED,",

"NOT EXCEEDING",

"DIVERGING SPEED PAST",

"SIGNAL AND THROUGH",

"TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 431") == 0) {

initAspects(30, 6, 0, 6, 3);

printLCD('A', 1, aspect, "", "SLOW TO", "CLEAR");

setLCDB(4, "PROCEED AT SLOW",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 432") == 0) {

initAspects(73, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "SLOW TO", "LIMITED");

setLCDB(8, "PROCEED AT SLOW",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 432A") == 0) {

initAspects(73, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "LIMITED");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT REDUCED",

"SPEED,",

"NOT EXCEEDING",

"DIVERGING SPEED PAST",

"SIGNAL AND THROUGH",

"TURNOUTS.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 433") == 0) {

initAspects(74, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "SLOW TO", "MEDIUM");

setLCDB(8, "PROCEED AT SLOW",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 433A") == 0) {

initAspects(74, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "DIVERGING", "TO MEDIUM");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT DIVERGING",

"SPEED, PAST SIGNAL &",

"THROUGH TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT MEDIUM",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 434") == 0) {

initAspects(70, 0, 0, 0, 0);

printLCD('A', 1, aspect, "", "SLOW TO", "SLOW");

setLCDB(8, "PROCEED AT SLOW",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT SLOW",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 434A") == 0) {

initAspects(70, 0, 0, 0, 0);

printLCD('A', 1, aspect, "DIVERGING", "TO", "DIVERGING");

setLCDB(8, "'DV' PLATE REQUIRED:",

" ",

"PROCEED AT DIVERGING",

"SPEED, PAST SIGNAL &",

"THROUGH TURNOUTS,",

"APPROACHING NEXT",

"SIGNAL AT DIVERGING",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 435") == 0) {

initAspects(44, 18, 0, 18, 27);

printLCD('A', 1, aspect, "", "SLOW TO", "STOP");

setLCDB(8, "PROCEED AT SLOW",

"SPEED PAST SIGNAL",

"AND THROUGH",

"TURNOUTS,",

"PREPARING TO STOP",

"AT THE NEXT SIGNAL", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 436") == 0) {

initAspects(31, 29, 12, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(4, "PROCEED AT",

"RESTRICTED SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 437") == 0) {

initAspects(0, 11, 12, 0, 0);

printLCD('A', 1, aspect, "", "STOP AND", "PROCEED");

setLCDB(4, "STOP, THEN PROCEED",

"AT RESTRICTED SPEED;",

"OFFSET SIGNAL HEADS",

"REQUIRED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 438") == 0) {

initAspects(77, 78, 0, 78, 79);

printLCD('A', 1, aspect, "", "TAKE/LEAVE", "SIDING");

setLCDB(8, "INDICATIONS WILL BE",

"SPECIFIED IN THE",

"SPECIAL",

"INSTRUCTIONS,",

"FOR EACH SPECIFIC",

"APPLICATION OF THIS",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "CROR 439") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 281") == 0) {

initAspects(1, 2, 3, 4, 0);

printLCD('A', 1, aspect, "", "CLEAR", "");

setLCDB(4, "PROCEED NOT",

"EXCEEDING NORMAL",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 281-A") == 0) { // TYPO

initAspects(53, 54, 51, 0, 0);

printLCD('A', 1, aspect, "", "CAB SPEED", "");

setLCDB(12, "PROCEED IN",

"ACCORDANCE WITH CAB",

"SIGNAL INDICATION;",

" ",

"REDUCE SPEED TO NOT",

"EXCEEDING 60 MPH IF",

"CAB SPEED CAB SIGNAL",

"IS DISPLAYED",

"WITHOUT A SIGNAL",

"SPEED, OR IF CAB",

"SIGNAL IS NOT",

"OPERATIVE.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 281-B") == 0) {

initAspects(13, 14, 0, 14, 0);

printLCD('A', 1, aspect, "", "APPROACH", "LIMITED");

setLCDB(4, "PROCEED APPROACHING",

"NEXT SIGNAL NOT",

"EXCEEDING LIMITED",

"SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 281-C") == 0) {

initAspects(15, 16, 0, 54, 51);

printLCD('A', 1, aspect, "", "LIMITED", "CLEAR");

setLCDB(8, "PROCEED AT LIMITED",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"NORMAL SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 282") == 0) {

initAspects(19, 20, 0, 20, 0);

printLCD('A', 1, aspect, "", "APPROACH", "MEDIUM");

setLCDB(4, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"MEDIUM SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 282-A") == 0) {

initAspects(55, 56, 27, 0, 0);

printLCD('A', 1, aspect, "", "ADVANCED", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT THE SECOND",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"LIMITED SPEED MUST",

"BEGIN REDUCTION TO",

"LIMITED SPEED",

"AS SOON AS ENGINE",

"PASSES THE SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 283") == 0) {

initAspects(5, 6, 0, 61, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "CLEAR");

setLCDB(8, "PROCEED AT MEDIUM",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"NORMAL SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 283-A") == 0) {

initAspects(24, 0, 0, 0, 0);

printLCD('A', 1, aspect, "MEDIUM", "APPROACH", "MEDIUM");

setLCDB(16, "PROCEED AT MEDIUM",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN APPROACH THE",

"NEXT SIGNAL AT",

"MEDIUM SPEED;",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEING REDUCTION TO",

"MEDIUM SPEED AS",

"SOON AS THE SIGNAL",

"IS CLEARLY VISIBLE.", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 284") == 0) {

initAspects(21, 22, 0, 22, 0);

printLCD('A', 1, aspect, "", "APPROACH", "SLOW");

setLCDB(12, "PROCEED APPROACHING",

"THE NEXT SIGNAL AT",

"SLOW SPEED;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED",

"AS SOON AS THE",

"ENGINE PASSES THE",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 285") == 0) {

initAspects(7, 8, 9, 0, 0);

printLCD('A', 1, aspect, "", "APPROACH", "");

setLCDB(12, "PROCEED PREPARED",

"TO STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED",

"AS SOON AS THE",

"ENGINE PASSES THE",

"SIGNAL.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 286") == 0) {

initAspects(28, 18, 0, 43, 0);

printLCD('A', 1, aspect, "", "MEDIUM", "APPROACH");

setLCDB(12, "PROCEED PREPARED",

"TO STOP AT THE NEXT",

"SIGNAL;",

" ",

"TRAINS EXCEEDING",

"MEDIUM SPEED MUST",

"BEGIN REDUCTION TO",

"MEDIUM SPEED",

"AS SOON AS THE",

"MEDIUM APPROACH",

"SIGNAL IS CLEARLY",

"VISIBLE.", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 287") == 0) {

initAspects(30, 0, 0, 2, 3);

printLCD('A', 1, aspect, "", "SLOW CLEAR", "");

setLCDB(8, "PROCEED AT SLOW",

"SPEED UNTIL ENTIRE",

"TRAIN CLEARS ALL",

"INTERLOCKING OR",

"SPRING SWITCHES,",

"THEN PROCEED AT",

"NORMAL SPEED.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 288") == 0) {

initAspects(44, 0, 0, 8, 27);

printLCD('A', 1, aspect, "", "SLOW", "APPROACH");

setLCDB(12, "PROCEED PREPARED TO",

"STOP AT NEXT SIGNAL;",

" ",

" ",

"SLOW SPEED APPLIES",

"UNTIL ENTIRE TRAIN",

"CLEARS ALL",

"INTERLOCKING",

"OR SPRING SWITCHES,",

"THEN MEDIUM SPEED",

"APPLIES.", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 290") == 0) { // TYPO

initAspects(31, 29, 0, 29, 9);

printLCD('A', 1, aspect, "", "RESTRICTING", "");

setLCDB(16, "PROCEED AT",

"RESTRICTED SPEED",

"UNTIL ENTIRE TRAIN",

"HAS CLEARED ALL",

"INTERLOCKING AND",

"SPRING SWITCHES (IF",

"THE SIGNAL IS AN",

"INTERLOCKING OR",

"CONTROLLED POINT",

"SIGNAL) AND THE",

"LEADING WHEELS HAVE",

"EITHER PASSED A MORE",

"FAVORABLE FIXED",

"SIGNAL, OR ENTERED",

"NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 291") == 0) {

initAspects(10, 11, 12, 11, 0);

printLCD('A', 1, aspect, "", "STOP AND", "PROCEED");

setLCDB(20, "No. PLATE REQ’D:",

"STOP, THEN PROCEED",

"AT RESTRICTED SPEED",

"UNTIL THE ENTIRE",

"TRAIN HAS CLEARED",

"ALL INTERLOCKING OR",

"SPRING SWITCHES (IF",

"THE SIGNAL IS",

"AN INTERLOCKING OR",

"CONTROLLED POINT",

"SIGNAL) AND THE",

"LEADING WHEELS",

"HAVE EITHER PASSED A",

"MORE FAVORABLE FIXED",

"SIGNAL, OR ",

" ",

"ENTERED NON-SIGNALED",

"TERRITORY.", "", "", "", "", "", "");

} else if (strcmp(aspect, "NORAC 292") == 0) {

initAspects(10, 11, 12, 11, 12);

printLCD('A', 1, aspect, "", "STOP", "");

setLCDB(4, "STOP.",

" ",

"ABSOLUTE SIGNALS",

"ONLY.", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "");

} else if (strcmp(aspect, "ROTATE") == 0 || strcmp(aspect, "RANDOM") == 0) {

printLCD('A', 1, "RANDOM INDICATIONS", "SELECTED", "", "");

printLCD('B', 1, " ", " ", " ", " ");

rotate = 1;

Serial.print("Enter time to cycle (1-99 sec), send 'C' to cancel ");

Serial.print(ack);

Serial.print(" ");

char cycleDelay[3] = {0, 0, 0};

readIN(cycleDelay);

if (cycleDelay[0] == 'C') {

Serial.print("\nEnter in rule number ");

loop();

}

sscanf(cycleDelay, "%d", &dly);

if (strcmp(aspect, "ROTATE") == 0) {

Serial.print("\nCycling... Send anything to terminate the cycle\n");

while (Serial.available() == 0) { sigCycle(); }

rotate = 0;

serialFlush();

} else if (strcmp(aspect, "RANDOM") == 0) {

Serial.print("\nCycling... Send anything to terminate the cycle\n");

while (Serial.available() == 0) { sigRand(); }

rotate = 0;

serialFlush();

Serial.print("\n");

}

} else { // Otherwise, send non-existant error

Serial.print("ERROR: NON\_EXISTANT\n");

Serial.print("Enter a rule number ");

Serial.print(ack);

loop();

}

pinCall();

}

// Method used to rotate through the different indications

void sigCycle() {

for (int i = 0; i < 136; i++) {

Serial.print("Indication #");

Serial.print(i);

Serial.print(ack);

Serial.print("\n");

initAspects(i, i, i, i, i);

pinCall();

}

}

// Method used to randomly rotate through the different indications

void sigRand() {

int r = random(0, 135);

Serial.print("Indication #");

Serial.print(r);

Serial.print(ack);

Serial.print("\n");

initAspects(r, r, r, r, r);

pinCall();

}

// Loads the aspects into the program to be translated into the pin call arrays

void initAspects(int A, int B, int C, int D, int E) { // Signal A, B, C, D, E - On the physical layout, from left to right: E, D, C, B, A

indications[0] = A; // Sets signal #0 to indication A

indications[1] = B;

indications[2] = C;

indications[3] = D;

indications[4] = E;

}

// Initalizes all the pin calls

void pinCall() {

indicationList(0, pin\_call\_A);

indicationList(1, pin\_call\_B);

indicationList(2, pin\_call\_C);

indicationList(3, pin\_call\_D);

indicationList(4, pin\_call\_E);

pinAssign();

}

// Chooses from the list what aspect the indication needs to be assigned to

void indicationList(int j, int pinMatrix[]) {

if (indications[j] == 0) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 1) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 2) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 3) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 4) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 5) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 6) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 7) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 8) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 9) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 10) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 11) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 12) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 13) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 14) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 15) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 16) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 17) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 18) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 19) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 20) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 21) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 22) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 23) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0);

} else if (indications[j] == 24) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0);

} else if (indications[j] == 25) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0);

} else if (indications[j] == 26) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0);

} else if (indications[j] == 27) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 28) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 29) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 30) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0);

} else if (indications[j] == 31) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0);

} else if (indications[j] == 32) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1);

} else if (indications[j] == 33) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 34) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0);

} else if (indications[j] == 35) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 36) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0);

} else if (indications[j] == 37) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 38) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 39) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 40) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 41) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 42) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 43) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 44) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 2, 0, 0);

} else if (indications[j] == 45) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 46) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 47) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 48) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 49) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 50) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 51) {

initPinCall(pinMatrix, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 52) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 53) {

initPinCall(pinMatrix, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 54) {

initPinCall(pinMatrix, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 55) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 56) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 57) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 58) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 59) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 60) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 61) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 62) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 63) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 64) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 65) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 2, 0, 0, 0);

} else if (indications[j] == 66) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0);

} else if (indications[j] == 67) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2, 0, 0);

} else if (indications[j] == 68) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 1, 0, 0);

} else if (indications[j] == 69) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 70) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 2, 0, 0);

} else if (indications[j] == 71) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 72) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 2, 0);

} else if (indications[j] == 73) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 2, 0, 0, 0);

} else if (indications[j] == 74) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0);

} else if (indications[j] == 75) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0);

} else if (indications[j] == 76) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0, 0, 0);

} else if (indications[j] == 77) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 2, 0);

} else if (indications[j] == 78) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 79) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 80) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 81) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 82) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 83) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 84) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 85) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 86) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 87) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 88) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 89) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 90) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0);

} else if (indications[j] == 91) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 92) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 93) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 94) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 95) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 96) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 97) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 98) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 99) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 100) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 101) {

initPinCall(pinMatrix, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 102) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0);

} else if (indications[j] == 103) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0);

} else if (indications[j] == 104) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0);

} else if (indications[j] == 105) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0);

} else if (indications[j] == 106) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0);

} else if (indications[j] == 107) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0);

} else if (indications[j] == 108) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 109) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0);

} else if (indications[j] == 110) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0);

} else if (indications[j] == 111) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0);

} else if (indications[j] == 112) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 113) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 114) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 115) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 116) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0);

} else if (indications[j] == 117) {

initPinCall(pinMatrix, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 118) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0);

} else if (indications[j] == 119) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 120) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0);

} else if (indications[j] == 121) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 122) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0);

} else if (indications[j] == 123) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0);

} else if (indications[j] == 124) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0);

} else if (indications[j] == 125) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0);

} else if (indications[j] == 126) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 1, 0);

} else if (indications[j] == 127) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 128) {

initPinCall(pinMatrix, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 129) {

initPinCall(pinMatrix, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0);

} else if (indications[j] == 130) {

initPinCall(pinMatrix, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 131) {

initPinCall(pinMatrix, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0);

} else if (indications[j] == 132) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 2, 0, 0, 0);

} else if (indications[j] == 133) {

initPinCall(pinMatrix, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0);

} else if (indications[j] == 134) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 2, 0, 0);

} else if (indications[j] == 135) {

initPinCall(pinMatrix, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 2, 0);

}

}

// Sets up the signal array

void initPinCall(int pinCall[], int a0, int a1, int a2, int a3, int a4, int a5, int a6, int a7, int a8, int a9, int a10, int a11, int a12, int a13) {

pinCall[0] = a0;

pinCall[1] = a1;

pinCall[2] = a2;

pinCall[3] = a3;

pinCall[4] = a4;

pinCall[5] = a5;

pinCall[6] = a6;

pinCall[7] = a7;

pinCall[8] = a8;

pinCall[9] = a9;

pinCall[10] = a10;

pinCall[11] = a11;

pinCall[12] = a12;

pinCall[13] = a13;

}

// Assigns actions to all the pins on the Arduino board

void pinAssign() {

int k = 2; // Sets up the code to light up signal A

for (int i = 2; i < 14; i++) { // Reads through the signal matrix

pins[i] = pin\_call\_A[k];

k++;

}

k = 2; // Signal B

for (int i = 22; i < 32; i++) {

pins[i] = pin\_call\_B[k];

k++;

}

k = 2; // Signal C

for (int i = 32; i < 40; i++) {

pins[i] = pin\_call\_C[k];

k++;

}

k = 2; // Signal D

for (int i = 40; i < 48; i++) {

pins[i] = pin\_call\_D[k];

k++;

}

k = 2; // Signal E

for (int i = 48; i < 54; i++) {

pins[i] = pin\_call\_E[k];

k++;

}

powerOn();

}

// Lights up the signals

void powerOn() {

f = 0;

for (int i = 2; i < 54; i++) { // DO NOT START i AT 0! ARDUINO WILL GO HAYWIRE!

if (pins[i] == 0) { // If 0:

digitalWrite(i, LOW); // Turn light off

} else if (pins[i] == 1) { // If 1:

digitalWrite(i, HIGH); // Turn light on

} else if (pins[i] == 2) { // If 2:

digitalWrite(i, HIGH); // Turn light on

flash[f++] = i; // Mark light for flashing

}

}

if (rotate == 0) { // If the program will NOT rotate through indications

Serial.print("Enter a rule number: ");

Serial.print(ack);

//flash

//serialFlush(); // Flushes any left over Java input (safeguard)

Serial.print("\n");

loop(); // returns to the aspect selector

} else if (rotate == 1) { // This block will hang the program on an indication for a specified period of time

while (Serial.available() == 0) { // Allows the cycle to be interrupted at any given time

int s = dly; // stores the delay specified by the Java output

while (s > 0) { // While time from next indication is greater than 0

delay(1000); // repeat flash section

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

digitalWrite(flash[i], LOW);

}

s--;

delay(1000);

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

digitalWrite(flash[i], HIGH);

}

s--;

}

break;

}

}

}