**Appendix C – Code for Lab 6 Part 1**

// Lab 7.c

// Chris Pybus and Jeff Pistacchio

//

//-------------------------------------------------------------------------------------------

// Includes

//-------------------------------------------------------------------------------------------

#include <c8051f120.h>

#include <stdio.h>

#include <string.h>

#include "putget.h"

#include <stdlib.h>

#include "LCD.c"

char interrupt **=** '~'**;**

unsigned char \_\_xdata myBoard**[**16**][**16**];**//store all of my board's values (first value denotes row, second value denotes column in said row)

unsigned char \_\_xdata opBoard**[**16**][**16**];**//store all of the values that I try to strike

\_\_sbit \_\_at **(**0xB0**)** P3\_0 **;**

\_\_sbit \_\_at **(**0xB1**)** P3\_1 **;**

\_\_sbit \_\_at **(**0xB2**)** P3\_2 **;**

\_\_sbit \_\_at **(**0xB3**)** P3\_3 **;**

\_\_sbit \_\_at **(**0xB4**)** P3\_4 **;**

\_\_sbit \_\_at **(**0xB5**)** P3\_5 **;**

\_\_sbit \_\_at **(**0xB6**)** P3\_6 **;**

\_\_sbit \_\_at **(**0xB7**)** P3\_7 **;**

char ptboatStamina **=** 2**;**

char cruiserStamina **=** 3**;**

char battleshipStamina **=** 4**;**

char carrierStamina **=** 5**;**

char getCharUART1**(**void**)** //will simply check if flag is high.

**{**

char c **=** ' '**;**

//recieved a character on uart 1 (other computer sent something from their uart 1)

SFRPAGE **=** UART1\_PAGE**;**

**if(**RI1 **==** 1**)**

**{**

RI1 **=** 0**;** //reset flag to 0

c **=** SBUF1**;** //get character from buffer

**return** c**;**

**}**

**return** ' '**;**

**}**

char getCharUART1Persistant**(**void**)** //will not return until a key is pressed

**{**

char c **=** ' '**;**

//recieved a character on uart 1 (other computer sent something from their uart 1)

SFRPAGE **=** UART1\_PAGE**;**

**while(**RI1 **!=** 1**){}** //wait for RI to go high

RI1 **=** 0**;** //reset flag to 0

c **=** SBUF1**;** //get character from buffer

lcd\_dat**(**c**);**

SFRPAGE **=** UART0\_PAGE**;**

**return** c**;**

**}**

char getKeypadPersistant**(**void**)** //will not return until a key is pressed, and only if the key is up/down/left/right/\*/#

**{**

char c **=** ' '**;**

**while(**interrupt **==** '~'**){}**

c **=** interrupt**;**

interrupt **=** '~'**;**

**return** c**;**

**}**

char getKeypadPersistantNav**(**void**)** //will not return until a key is pressed, and only if the key is up/down/left/right/\*/#

**{**

char c **=** ' '**;**

**while(**1**)**

**{**

**while(**interrupt **==** '~'**){}**

c **=** interrupt**;**

interrupt **=** '~'**;**

**if(**c **==** '2' **||** c **==** '6' **||** c **==** '8' **||** c **==** '4' **||** c **==** '\*' **||** c **==** '#'**)**

**{**

**break;**

**}**

**}**

**return** c**;**

**}**

void TIMER\_Init**(**void**)**

**{**

//Timer1, used for UART1, baud 115200

SFRPAGE **=** TIMER01\_PAGE**;**

TCON **=** 0x40**;**

TMOD **=** 0x20**;**

CKCON **=** 0x10**;**

TH1 **=** 0xA0**;**

//Timer2, used for UART0, baud 115200

SFRPAGE **=** TMR2\_PAGE**;**

TMR2CN **=** 0x04**;**

TMR2CF **=** 0x08**;**

RCAP2L **=** 0xF4**;**

RCAP2H **=** 0xFF**;**

**}**

void PORT\_Init**()**

**{**

SFRPAGE **=** CONFIG\_PAGE**;**

XBR0 **=** 0x04**;** //eanble UART0

XBR1 **=** 0x04**;** //enable INT0 on P0.4

XBR2 **=** 0x44**;** //enable UART1

P0MDOUT **=** 0x15**;**

P7MDOUT **=** 0x07**;** // Set E, RW, RS controls to push-pull

P6MDOUT **=** 0x00**;** // P6 must be open-drain to be bidirectional:

// used for both read & write operations

P3MDOUT**=**0xF0**;** // hi nibble to push-pull, lo nibble to open-drain

P3**=**0x0F**;** // write 0's to Port 3 hi nibble, lo nibble set for input

TCON**=**TCON **&** 0xFC**;** // Clear INT0 flag and set for level triggered

IE**=**IE**|**0x81**;** // Enable all interrupts & enable INT0

**}**

void SYSCLK\_Init**(**void**)**

**{**

int i2 **=** 0**;**

SFRPAGE **=** CONFIG\_PAGE**;**

OSCXCN **=** 0x67**;**

**for** **(**i2 **=** 0**;** i2 **<** 3000**;** i2**++);** // Wait 1ms for initialization

**while** **((**OSCXCN **&** 0x80**)** **==** 0**);**

CLKSEL **=** 0x01**;**

OSCICN **&=** **~**0x80**;**

**}**

void UART\_Init**()**

**{**

//UART0

SFRPAGE **=** UART0\_PAGE**;**

SCON0 **=** 0x50**;**

SSTA0 **=** 0x05**;**

TI0 **=** 1**;**

//UART1

SFRPAGE **=** UART1\_PAGE**;**

SCON1 **=** 0x30**;**

TI1 **=** 1**;**

**}**

void sendMessage**(**char **\***str**){**

size\_t size **=** 0**;**

int index **=** 0**;**

int i4 **=** 0**;**

SFRPAGE **=** UART1\_PAGE**;**

size **=** 4**;**//strlen(str);

**for(**index **=** 0**;** index **<** size**;** index**++)** //need to replace 7 with sometjing useful

**{**

printf**(**"%c"**,** str**[**index**]);**

**for(**i4 **=** 0**;** i4 **<** 10000**;** i4**++){}**

**}**

SFRPAGE **=** UART0\_PAGE**;**

**}**

void SW2\_ISR **(**void**)** \_\_interrupt 0

**{**

unsigned int i **=** 0**;**

char portvalue**;**

int column**;**

char output **=** ' '**;**

**for(**i **=** 0**;** i**<**30000**;** i**++);** // wait for the output and input pins to stabilize

**if(**P3\_3 **==** 0**)**

**{**

column **=** 1**;**

**}**

**else** **if(**P3\_2 **==** 0**)**

**{**

column **=** 2**;**

**}**

**else** **if(**P3\_1 **==** 0**)**

**{**

column **=** 3**;**

**}**

**else** **if(**P3\_0 **==** 0**)**

**{**

column **=** 4**;**

**}**

**else**

**{**

**return;**

**}**

//check if row one (will make all 4 least sig digits of P3 go high)

P3 **=** 0x8F**;**

**for(**i **=** 0**;** i**<**3000**;** i**++);**

portvalue **=** P3 **&** 0x0F**;**

**if(**portvalue **==** 0x0F**)**

**{**

**if(**column **==** 1**)**

output **=** '1'**;**

**else** **if(**column **==** 2**)**

output **=** '2'**;**

**else** **if(**column **==** 3**)**

output **=** '3'**;**

**else** **if(**column **==** 4**)**

output **=** 'A'**;**

**}**

**else**

**{**

P3 **=** 0x4F**;**

**for(**i **=** 0**;** i**<**3000**;** i**++);**

portvalue **=** P3 **&** 0x0F**;**

**if(**portvalue **==** 0x0F**)**

**{**

**if(**column **==** 1**)**

output **=** '4'**;**

**else** **if(**column **==** 2**)**

output **=** '5'**;**

**else** **if(**column **==** 3**)**

output **=** '6'**;**

**else** **if(**column **==** 4**)**

output **=** 'B'**;**

**}**

**else**

**{**

P3 **=** 0x2F**;**

**for(**i **=** 0**;** i**<**3000**;** i**++);**

portvalue **=** P3 **&** 0x0F**;**

**if(**portvalue **==** 0x0F**)**

**{**

**if(**column **==** 1**)**

output **=** '7'**;**

**else** **if(**column **==** 2**)**

output **=** '8'**;**

**else** **if(**column **==** 3**)**

output **=** '9'**;**

**else** **if(**column **==** 4**)**

output **=** 'C'**;**

**}**

**else**

**{**

P3 **=** 0x1F**;**

**for(**i **=** 0**;** i**<**3000**;** i**++);**

portvalue **=** P3 **&** 0x0F**;**

**if(**portvalue **==** 0x0F**)**

**{**

**if(**column **==** 1**)**

output **=** '\*'**;**

**else** **if(**column **==** 2**)**

output **=** '0'**;**

**else** **if(**column **==** 3**)**

output **=** '#'**;**

**else** **if(**column **==** 4**)**

output **=** 'D'**;**

**}**

**}**

**}**

**}**

P3 **=** 0x0F**;**

**for(**i **=** 0**;** i**<**3000**;** i**++);**

lcd\_dat**(**output**);**

interrupt **=** output**;**

**while(**P3\_3 **==** 0 **||** P3\_2 **==** 0 **||** P3\_1 **==** 0 **||** P3\_0 **==** 0**){}**

**}**

void displayBoardOutline**()**

**{**

int i **=** 0**;**

char cords**[**16**]** **=** **{**'0'**,** '1'**,** '2'**,** '3'**,** '4'**,** '5'**,** '6'**,** '7'**,** '8'**,** '9'**,** 'A'**,** 'B'**,** 'C'**,** 'D'**,** 'E'**,** 'F'**};**

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[5;17H"**);**

printf**(**"Your board Shots taken"**);** //print out character to console

printf**(**"\033[6;5H"**);**

printf**(**"+- - - - - - - - - - - - - - - - + +- - - - - - - - - - - - - - - - +\n\r"**);** //print out character to console

**for(**i **=** 0**;** i **<** 16**;** i**++)**

**{**

printf**(**"\033[%i;4H"**,(**i**+**7**));**

printf**(**"%c| | | |%c\n\r"**,** cords**[**i**],** cords**[**i**]);** //print out character to console

**}**

printf**(**"\033[%i;5H"**,(**i**+**7**));**

printf**(**"+- - - - - - - - - - - - - - - - + +- - - - - - - - - - - - - - - - +\n\r"**);** //print out character to console

printf**(**"\033[%i;6H"**,(**i**+**8**));**

**for(**i **=** 0**;** i **<** 16**;** i**++)**

**{**

printf**(**"%c "**,** cords**[**i**]);** //print out character to console

**}**

printf**(**" "**);**

**for(**i **=** 0**;** i **<** 16**;** i**++)**

**{**

printf**(**"%c "**,** cords**[**i**]);** //print out character to console

**}**

**}**

void printBoard**()**

**{**

int x **=** 0**;**

int y **=** 0**;**

**for(**y **=** 0**;** y **<** 16**;** y**++)**

**{**

**for(**x **=** 0**;** x **<** 16**;** x**++)**

**{**

**if(**myBoard**[**y**][**x**]** **!=** 'w'**)**

**{**

printf**(**"\033[%i;%iH"**,(**y**+**7**),(**x**\***2 **+** 6**));**

printf**(**"%c"**,** myBoard**[**y**][**x**]);** //use this if you want to print the specific ship

//printf("x");

**}**

**}**

**}**

printf**(**"\033[0;0H"**);**

**}**

void printOpBoard**()**

**{**

int x **=** 0**;**

int y **=** 0**;**

**for(**y **=** 0**;** y **<** 16**;** y**++)**

**{**

**for(**x **=** 0**;** x **<** 16**;** x**++)**

**{**

**if(**opBoard**[**y**][**x**]** **!=** 'w'**)**

**{**

printf**(**"\033[%i;%iH"**,(**y**+**7**),(**x**\***2 **+** 43**));**

printf**(**"%c"**,** opBoard**[**y**][**x**]);** //use this if you want to print the specific ship

//printf("x");

**}**

**}**

**}**

printf**(**"\033[0;0H"**);**

**}**

char canMove**(**unsigned int x\_next**,** unsigned int y\_next**,** unsigned int orientation\_next**,** unsigned int size**)**

**{**

//bounds: min x = 6 min y = 7 (inclusive)

// max x = 37 max y = 22 (inclusive)

unsigned int x\_relative **=** **(**x\_next **-** 6**)** **/** 2**;**

unsigned int y\_relative **=** **(**y\_next **-** 7**);**

unsigned int i **=** 0**;**

//first check if the cursor itself has moved outside the bounds

**if(**x\_next **>** 37 **||** x\_next **<** 6 **||** y\_next **>** 22 **||** y\_next **<** 7**)**

**{**

**return** 0**;** //false, cannot move

**}**

**if(**orientation\_next **==** 0**)** //horizontal

**{**

**if((**x\_next **+** **(**size **\*** 2**)** **-** 1 **)** **>** 37**)**

**return** 0**;**

**}**

**else**

**{**

**if((**y\_next **+** size**)** **-** 1 **>** 22**)**

**return** 0**;**

**}**

**if(**orientation\_next **==** 0**)** //horizontal

**{**

**for(**i **=** 0**;** i **<** size**;** i**++)**

**{**

**if(**myBoard**[**y\_relative**][**x\_relative**+**i**]** **!=** 'w'**)**

**return** 0**;**

**}**

**}**

**else**

**{**

**for(**i **=** 0**;** i **<** size**;** i**++)**

**{**

**if(**myBoard**[**y\_relative**+**i**][**x\_relative**]** **!=** 'w'**)**

**return** 0**;**

**}**

**}**

**return** 1**;** //true, can move

**}**

void addToBoard**(**unsigned int x**,** unsigned int y**,** unsigned int orientation**,** unsigned int size**)**

**{**

unsigned int x\_relative **=** **(**x **-** 6**)** **/** 2**;**

unsigned int y\_relative **=** **(**y **-** 7**);**

unsigned int i **=** 0**;**

**if(**orientation **==** 1**)** //vertical

**{**

**for(**i **=** 0**;** i **<** size**;** i**++)**

**{**

myBoard**[**y\_relative**+**i**][**x\_relative**]** **=** size **+** '0'**;** //increment the row while keeping column static

**}**

**}**

**else** //horizontal

**{**

**for(**i **=** 0**;** i **<** size**;** i**++)**

**{**

myBoard**[**y\_relative**][**x\_relative**+**i**]** **=** size **+** '0'**;** //increment column while staying on the same row

**}**

**}**

**}**

void placeShip**(**unsigned int ship**)** //ship 5 = carrier, 4 = battleship, 3 = cruiser, 2 = pt boat (persistant function)

**{**

unsigned char input **=** ' '**;**

unsigned int i **=** 0**;**

unsigned int orientation **=** 0**;** //0 == horizontal, 1 == vertical

unsigned int orientation\_next**;**

unsigned int x **=** 6**;** //horizontal

unsigned int y **=** 7**;** //vertical

unsigned int x\_next **=** x**;**

unsigned int y\_next **=** y**;**

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[%i;%iH"**,** y**,** x**);**

**for(**i **=** 0**;** i **<** ship**;** i**++)**

printf**(**"x "**);**

printf**(**"\033[%i;%iH"**,** y**,** x**);**

**while(**1**)**

**{**

input **=** getKeypadPersistantNav**();** //this function ensures input char is correct

**if(**input **==** '\*'**)**

**{**

addToBoard**(**x**,** y**,** orientation**,** ship **);** //convert to char

**break;**

**}**

**else**

**{**

x\_next **=** x**;**

y\_next **=** y**;**

orientation\_next **=** orientation**;**

**if(**input **==** '2'**)** //up

y\_next **=** y **-** 1**;**

**else** **if(**input **==** '6'**)** //right

x\_next **=** x **+** 2**;**

**else** **if(**input **==** '8'**)** //down

y\_next **=** y **+** 1**;**

**else** **if(**input **==** '4'**)** //left

x\_next **=** x **-** 2**;**

**else** **if(**input **==** '#'**)** //rotate

**{**

**if(**orientation **==** 0**)**

orientation\_next **=** 1**;**

**else**

orientation\_next **=** 0**;**

**}**

**if(**canMove**(**x\_next**,** y\_next**,** orientation\_next**,** ship**))**

**{**

//erase current ship

printf**(**"\033[%i;%iH"**,** y**,** x**);**

**if(**orientation**==**0**)** //ship was previously horizontal

**{**

**for(**i **=** 0**;** i **<** ship**;** i**++)** //erase the current ship

printf**(**" "**);**

**}**

**else** //ship was previously vertical

**{**

**for(**i **=** 0**;** i **<** ship**+**1**;** i**++)** //erase the current ship

**{**

printf**(**" "**);**

printf**(**"\033[%i;%iH"**,** y**+**i**,** x**);**

**}**

**}**

//print new ship

printf**(**"\033[%i;%iH"**,** y\_next**,** x\_next**);**

**if(**orientation\_next**==**0**)** //ship is now horizontal

**{**

**for(**i **=** 0**;** i **<** ship**;** i**++)** //print new ship

printf**(**"x "**);**

printf**(**"\033[%i;%iH"**,** y\_next**,** x\_next**);** //move the cursor to new position

**}**

**else** //ship is now vertical

**{**

**for(**i **=** 0**;** i **<** ship**+**1**;** i**++)** //print new ship

**{**

printf**(**"x "**);**

printf**(**"\033[%i;%iH"**,** y\_next**+**i**,** x\_next**);**

**}**

printf**(**"\033[%i;%iH"**,** y\_next**,** x\_next**);** //move the cursor to new position

**}**

x **=** x\_next**;**

y **=** y\_next**;**

orientation **=** orientation\_next**;**

**}**

**}**

**}**

**}**

void fillBoards**(**void**)**

**{**

int x **=** 0**;**

int y **=** 0**;**

**for(**y **=** 0**;** y **<** 16**;** y**++)**

**{**

**for(**x **=** 0**;** x **<** 16**;** x**++)**

**{**

myBoard**[**y**][**x**]** **=** 'w'**;**

opBoard**[**y**][**x**]** **=** 'w'**;**

**}**

**}**

**}**

int checkIfAlreadyShotThere**(**char row**,** char col**)**

**{**

unsigned int numRow **=** row **-** '0'**;**

unsigned int numCol **=** col **-** '0'**;**

**if(**row **==** 'A' **||** row **==** 'B' **||** row **==** 'C' **||** row **==** 'D' **||** row **==** 'E' **||** row **==** 'F'**)**

numRow **=** **(**row **-** '0'**)** **-** 7**;**

**else**

numRow **=** row **-** '0'**;**

**if(**col **==** 'A' **||** col **==** 'B' **||** col **==** 'C' **||** col **==** 'D' **||** col **==** 'E' **||** col **==** 'F'**)**

numCol **=** **(**col **-** '0'**)** **-** 7**;**

**else**

numCol **=** col **-** '0'**;**

**if(**opBoard**[**numRow**][**numCol**]** **==** 'x' **||** opBoard**[**numRow**][**numCol**]** **==** 'o'**)**

**return** 0**;** //already shot there

**return** 1**;** //good to go

**}**

int checkIfHit**(**char row**,** char col**)**

**{**

unsigned int numRow **=** row **-** '0'**;**

unsigned int numCol **=** col **-** '0'**;**

char hit**;**

**if(**row **==** 'A' **||** row **==** 'B' **||** row **==** 'C' **||** row **==** 'D' **||** row **==** 'E' **||** row **==** 'F'**)**

numRow **=** **(**row **-** '0'**)** **-** 7**;**

**else**

numRow **=** row **-** '0'**;**

**if(**col **==** 'A' **||** col **==** 'B' **||** col **==** 'C' **||** col **==** 'D' **||** col **==** 'E' **||** col **==** 'F'**)**

numCol **=** **(**col **-** '0'**)** **-** 7**;**

**else**

numCol **=** col **-** '0'**;**

hit **=** myBoard**[**numRow**][**numCol**];**

**if(**hit **==** '2'**)**

**{**

ptboatStamina**--;**

myBoard**[**numRow**][**numCol**]** **=** 'x'**;**

**return** 1**;**

**}**

**else** **if(**hit **==** '3'**)**

**{**

cruiserStamina**--;**

myBoard**[**numRow**][**numCol**]** **=** 'x'**;**

**return** 1**;**

**}**

**else** **if(**hit **==** '4'**)**

**{**

battleshipStamina**--;**

myBoard**[**numRow**][**numCol**]** **=** 'x'**;**

**return** 1**;**

**}**

**else** **if(**hit **==** '5'**)**

**{**

carrierStamina**--;**

myBoard**[**numRow**][**numCol**]** **=** 'x'**;**

**return** 1**;**

**}**

myBoard**[**numRow**][**numCol**]** **=** 'o'**;**

**return** 0**;**

**}**

void addHitOrMiss**(**char row**,** char col**,** int hitMiss**)**

**{**

unsigned int numRow **=** row **-** '0'**;**

unsigned int numCol **=** col **-** '0'**;**

**if(**row **==** 'A' **||** row **==** 'B' **||** row **==** 'C' **||** row **==** 'D' **||** row **==** 'E' **||** row **==** 'F'**)**

numRow **=** **(**row **-** '0'**)** **-** 7**;**

**else**

numRow **=** row **-** '0'**;**

**if(**col **==** 'A' **||** col **==** 'B' **||** col **==** 'C' **||** col **==** 'D' **||** col **==** 'E' **||** col **==** 'F'**)**

numCol **=** **(**col **-** '0'**)** **-** 7**;**

**else**

numCol **=** col **-** '0'**;**

**if(**hitMiss **==** 1**)**

**{**

opBoard**[**numRow**][**numCol**]** **=** 'x'**;** //hit

**}**

**else**

**{**

opBoard**[**numRow**][**numCol**]** **=** 'o'**;** //miss

**}**

**}**

void main **(**void**)**

**{**

char response**[**4**]** **=** **{**'R'**,** ' '**,** ' '**,** ' '**};**

int first **=** 0**;**

int index **=** 0**;**

char row **=** ' '**;**

char col **=** ' '**;**

char c**;**

WDTCN **=** 0xDE**;** // Disable watchdog timer.

WDTCN **=** 0xAD**;**

SFRPAGE **=** CONFIG\_PAGE**;**

SYSCLK\_Init**();** // Initialize the oscillator.

TIMER\_Init**();** // Initialize timers

UART\_Init**();** // Initialize UARTs.

PORT\_Init**();** // Configure the Crossbar and GPIO.

lcd\_init**();** // initialize the LCD screen

lcd\_cmd**(**0x3F**);** // set display to 2 lines 5x8

lcd\_cmd**(**0x0C**);** // turn on display and cursor

lcd\_cmd**(**0x01**);** // clear display

lcd\_dat**(**'#'**);**

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[2J"**);** //Erase screen and move cursor to the home position.

/\*

while(1)

{

printf("%c", getKeypadPersistantNav());

}

\*/

fillBoards**();**

displayBoardOutline**();**

**while(**1**)**

**{**

printf**(**"\033[1;5H"**);**

printf**(**"Setup your own board\n\r"**);** //print out character to console

printf**(**"\033[2;5H"**);**

printf**(**"Use 2/6/8/7 to move up/right/down/left\n\r"**);** //print out character to console

printf**(**"\033[3;5H"**);**

printf**(**"Place carrier with \*, rotate with # "**);** //print out character to console

placeShip**(**5**);**

printf**(**"\033[3;5H"**);**

printf**(**"Place battleship with \*, rotate with # "**);** //print out character to console

placeShip**(**4**);**

printf**(**"\033[3;5H"**);**

printf**(**"Place cruiser with \*, rotate with # "**);** //print out character to console

placeShip**(**3**);**

printf**(**"\033[3;5H"**);**

printf**(**"Place pt boat with \*, rotate with # "**);** //print out character to console

placeShip**(**2**);**

printBoard**();**

//MAYBE ADD CODE TO PROMPT USER TO REDO ALL PLACEMENTS

printf**(**"\033[1;5H"**);**

printf**(**" "**);**

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

printf**(**"\033[3;5H"**);**

printf**(**" "**);**

**if(**getCharUART1**()** **==** 'R'**)** //other player is ready

**{**

//send response that we are ready too

SFRPAGE **=** UART1\_PAGE**;**

printf**(**"R"**);** //signal that we are ready

SFRPAGE **=** UART0\_PAGE**;**

//other player goes first because they were ready first

**}**

**else**

**{**

//send initial ready signal, wait for return

SFRPAGE **=** UART1\_PAGE**;**

printf**(**"R"**);** //signal that we are ready

first **=** 1**;** //we go first

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[2;5H"**);**

printf**(**"Waiting for other player to finish placing ships..."**);**

**while(**1**)**

**{**

c **=** getCharUART1Persistant**();**

**if(**c **==** 'R'**)**

**{**

//other player is ready, we go first

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

**break;**

**}**

**}**

**}**

**if(**first **==** 1**)**

**{**

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

printf**(**"\033[2;5H"**);**

printf**(**"Type in coordinates of your strike (row,col): "**);**

**while(**1**)**

**{**

row **=** getKeypadPersistant**();**

**if(**row **==** '\*'**)**

row **=** 'F'**;**

**if(**row **==** '#'**)**

row **=** 'E'**;**

printf**(**"(%c,"**,** row**);**

col **=** getKeypadPersistant**();**

**if(**col **==** '\*'**)**

col **=** 'F'**;**

**if(**col **==** '#'**)**

col **=** 'E'**;**

printf**(**"%c)"**,** col**);**

**if(**checkIfAlreadyShotThere**(**row**,** col**)** **==** 1**)**

**{**

**break;**

**}**

**else**

**{**

printf**(**"\033[2;5H"**);**

printf**(**"You've already shot there. Try another (row,col): "**);**

**}**

**}**

printf**(**"\033[1;5H"**);**

printf**(**" "**);**

response**[**0**]** **=** row**;**

response**[**1**]** **=** ':'**;**

response**[**2**]** **=** col**;**

response**[**3**]** **=** '!'**;**

sendMessage**(**response**);**

index **=** 0**;**

lcd\_cmd**(**0x01**);** // clear display

**for(**index **=** 0**;** index **<** 4**;** index**++)**

**{**

response**[**index**]** **=** getCharUART1Persistant**();**

**}**

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

**if(**response**[**0**]** **==** 'K'**)**

**{**

printf**(**"\033[1;5H"**);**

printf**(**"Your salvo to (%c,%c) was a HIT! "**,** row**,** col**);**

addHitOrMiss**(**row**,** col**,** 1**);**

**if(**response**[**2**]** **!=** 'A'**)**

**{**

**if(**response**[**2**]** **==** '2'**)**

**{**

printf**(**" You sunk their PT Boat! "**);**

**}**

**else** **if(**response**[**2**]** **==** '3'**)**

**{**

printf**(**" You sunk their Cruiser! "**);**

**}**

**else** **if(**response**[**2**]** **==** '4'**)**

**{**

printf**(**" You sunk their Battleship! "**);**

**}**

**else** **if(**response**[**2**]** **==** '5'**)**

**{**

printf**(**" You sunk their Carrier! "**);**

**}**

**}**

**else**

**{**

printf**(**" "**);**

**}**

**}**

**else** **if(**response**[**0**]** **==** 'M' **&&** response**[**1**]** **==** 'I' **&&** response**[**2**]** **==** 'S' **&&** response**[**3**]** **==** '!'**)**

**{**

printf**(**"\033[1;5H"**);**

printf**(**"Your salvo to (%c,%c) was a MISS! "**,** row**,** col**);**

addHitOrMiss**(**row**,** col**,** 0**);**

**}**

//update strike board

printOpBoard**();**

printf**(**"\033[2;5H"**);**

printf**(**"Other player's turn... "**);**

**}**

**while(**1**)**

**{**

printf**(**"\033[2;5H"**);**

printf**(**"Other player's turn... "**);**

//strike cord message: "R:C!"

index **=** 0**;**

lcd\_cmd**(**0x01**);** // clear display

**for(**index **=** 0**;** index **<** 4**;** index**++)**

**{**

response**[**index**]** **=** getCharUART1Persistant**();**

**}**

printf**(**"\033[1;5H"**);**

printf**(**"Other player sent a salvo to (%c,%c)! "**,** response**[**0**],** response**[**2**]);**

**if(**checkIfHit**(**response**[**0**],** response**[**2**])** **==** 1**)**

**{**

response**[**0**]** **=** 'K'**;**

response**[**1**]** **=** 'I'**;**

response**[**2**]** **=** 'A'**;**

response**[**3**]** **=** '!'**;**

printf**(**"It hit one of your ships! "**);**

**if(**ptboatStamina **==** 0**)**

**{**

response**[**2**]** **=** '2'**;**

ptboatStamina**--;**

**}**

**else** **if(**cruiserStamina **==** 0**)**

**{**

response**[**2**]** **=** '3'**;**

cruiserStamina**--;**

**}**

**else** **if(**battleshipStamina **==** 0**)**

**{**

response**[**2**]** **=** '4'**;**

battleshipStamina**--;**

**}**

**else** **if(**carrierStamina **==** 0**)**

**{**

response**[**2**]** **=** '5'**;**

carrierStamina**--;**

**}**

sendMessage**(**response**);**

**}**

**else**

**{**

printf**(**"It did not hit anything. "**);**

response**[**0**]** **=** 'M'**;**

response**[**1**]** **=** 'I'**;**

response**[**2**]** **=** 'S'**;**

response**[**3**]** **=** '!'**;**

sendMessage**(**response**);**

**}**

printBoard**();**

//==================================

//SWITCH CONTROL

//==================================

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

printf**(**"\033[2;5H"**);**

printf**(**"Type in coordinates of your strike (row,col): "**);**

**while(**1**)**

**{**

row **=** getKeypadPersistant**();**

**if(**row **==** '\*'**)**

row **=** 'F'**;**

**if(**row **==** '#'**)**

row **=** 'E'**;**

printf**(**"(%c,"**,** row**);**

col **=** getKeypadPersistant**();**

**if(**col **==** '\*'**)**

col **=** 'F'**;**

**if(**col **==** '#'**)**

col **=** 'E'**;**

printf**(**"%c)"**,** col**);**

**if(**checkIfAlreadyShotThere**(**row**,** col**)** **==** 1**)**

**{**

**break;**

**}**

**else**

**{**

printf**(**"\033[2;5H"**);**

printf**(**"You've already shot there. Try another (row,col): "**);**

**}**

**}**

printf**(**"\033[1;5H"**);**

printf**(**" "**);**

response**[**0**]** **=** row**;**

response**[**1**]** **=** ':'**;**

response**[**2**]** **=** col**;**

response**[**3**]** **=** '!'**;**

sendMessage**(**response**);**

index **=** 0**;**

lcd\_cmd**(**0x01**);** // clear display

**for(**index **=** 0**;** index **<** 4**;** index**++)**

**{**

response**[**index**]** **=** getCharUART1Persistant**();**

**}**

printf**(**"\033[2;5H"**);**

printf**(**" "**);**

**if(**response**[**0**]** **==** 'K'**)**

**{**

printf**(**"\033[1;5H"**);**

printf**(**"Your salvo to (%c,%c) was a HIT! "**,** row**,** col**);**

addHitOrMiss**(**row**,** col**,** 1**);**

**if(**response**[**2**]** **!=** 'A'**)**

**{**

**if(**response**[**2**]** **==** '2'**)**

**{**

printf**(**" You sunk their PT Boat! "**);**

**}**

**else** **if(**response**[**2**]** **==** '3'**)**

**{**

printf**(**" You sunk their Cruiser! "**);**

**}**

**else** **if(**response**[**2**]** **==** '4'**)**

**{**

printf**(**" You sunk their Battleship! "**);**

**}**

**else** **if(**response**[**2**]** **==** '5'**)**

**{**

printf**(**" You sunk their Carrier! "**);**

**}**

**}**

**else**

**{**

printf**(**" "**);**

**}**

**}**

**else** **if(**response**[**0**]** **==** 'M' **&&** response**[**1**]** **==** 'I' **&&** response**[**2**]** **==** 'S' **&&** response**[**3**]** **==** '!'**)**

**{**

printf**(**"\033[1;5H"**);**

printf**(**"Your salvo to (%c,%c) was a MISS! "**,** row**,** col**);**

addHitOrMiss**(**row**,** col**,** 0**);**

**}**

//update strike board

printOpBoard**();**

printf**(**"\033[2;5H"**);**

printf**(**"Other player's turn... "**);**

**}**

**}**

**}**