Pseudo Code Lab 1-2

Ethan Grey

compiler directives

#include <c88051\_SDCC.h>

#include <stdio.h>

declare global variables

sbit SS, PB0, PB1, LED0, LED1, BILED0, BILED1

function prototypes

void Port\_Init(void)

void Set\_Outputs(void)

char Rand\_Num(void)

void Check\_Inputs

main function

declare local variables

char round = 0; // Used to keep track of the rounds

char correct = 0; // Used to keep score

char rand = 0; // Used for random sequence

initialization functions

Sys\_Init();

putchar(' ');

Port\_Init();

Begin infinite loop execute

Set\_Outputs(void) function to read sbit inputs and set sbit outputs

Check\_Inputs(rand) function to read push buttons and check if it was correct

End infinite loop

End main function

Functions

void Port\_Init(void)

Set SFRs P2, P3, P2MDOUT & P3MDOUT so P2.0, P3.0 & P3.1 are inputs,

P3.3, P3.4, P3.6 & P3.7 are outputs

End Port\_Init

void Set\_Outputs(void)

If SS is off then

Print "Game paused"

Else if (round == 10) (user has completed all 10 rounds)

Print number of correct presses

While SS is on

Wait

End while loop

If SS is off

Reset round counter

Else (this means SS is on, game is running)

Char tempRand = rand;

While (tempRand == rand) Make sure the random value isn’t the same

rand = Rand\_Num() (set the random number)

end while loop

switch rand

case 0

LED0 is on

LED1 is off

Case 1

LED0 is off

LED1 is on

Case 2

LED0 is on

LED1 is on

End Set\_Outputs

Void Check\_Inputs(char num)

While timer is less than 1 second

If SS is off

Pause the timer

End while loop

Switch rand

Case 0

If PB0 is on and PB1 is off

Correct++

Turn BILED green

Else

Turn BILED red

Case 1

If PB0 is off and PB1 is on

Correct++

Turn BILED green

Else

Turn BILED red

Case 2

If PB0 is on and PB1 is on

Correct++

Turn BILED green

Else

Turn BILED red

End Check\_Inputs

Char Rand\_Num(void)

Generate random number form 0 to 2

Return the number