Exercise 3.4

```
#include <xc.h>          // Load the proper header for the processor

void delay(void);

int main(void) {
  TRISF = 0xFFFC;        // Pins 0 and 1 of Port F are LED1 and LED2.  Clear
                         // bits 0 and 1 to zero, for output.  Others are inputs.
  LATFbits.LATF0 = 1;    // Turn LED1 and LED2 off.  These pins sink current
  LATFbits.LATF1 = 1;    // on the NU32, so "high" (1) = "off" and "low" (0) = "on"

  while(1) {
    delay();
    LATFINV = 0x0003;    // toggle LED1 and LED2; same as LATFINV = 0x3;
  }
  return 0;
}

void delay(void) {
  int j;
  for (j = 0; j < 1000000; j++) { // number is 1 million
    while(!PORTDbits.RD7) {
        ;   // Pin D7 is the USER switch, low (FALSE) if pressed.
    }
  }
}
```

Exercise 3.5

```
#include <xc.h>          // Load the proper header for the processor
#define MAXCYCLES 1000000
#define DELTACYCLES  100000

void delay(int cycles);

int main(void) {
       int cycles=MAXCYCLES;
       TRISF = 0xFFFC;        // Pins 0 and 1 of Port F are LED1 and LED2.  Clear
                                          // bits 0 and 1 to zero, for output.  Others
are inputs.
       LATFbits.LATF0 = 0;    // Turn LED1 on and LED2 off.  These pins sink current
       LATFbits.LATF1 = 0;    // on the NU32, so "high" (1) = "off" and "low" (0) = "on"

       while (1) {

              if (cycles < 0) {
                     cycles = MAXCYCLES;
              }

              else {
                     delay(cycles);
                     LATFINV = 0x3;    // toggle LED1 and LED2; same as LATFINV = 0x3;
                     cycles -= DELTACYCLES;
              }
       }
       return 0;
```

```
}

void delay(int cycles) {
	int j;
	for (j = 0; j < cycles; j++) { // number is 1 million
		while (!PORTDbits.RD7) {
			;   // Pin D7 is the USER switch, low (FALSE) if pressed.
		}
	}
}
```

Exercise 4.1

Private to NU32.C

Functions

```
// disable interrupts
__builtin_disable_interrupts();

__builtin_mtc0(_CP0_CONFIG, _CP0_CONFIG_SELECT, 0xa4210583);

__builtin_enable_interrupts();
```

Constants

```
char data = 0;
int complete = 0, num_bytes = 0;

data = U3RXREG;      // read the data

U3TXREG = *string;
```

Global variables

```
#define NU32_DESIRED_BAUD 230400     // Baudrate for RS232
```

To be used in other C files

Functions

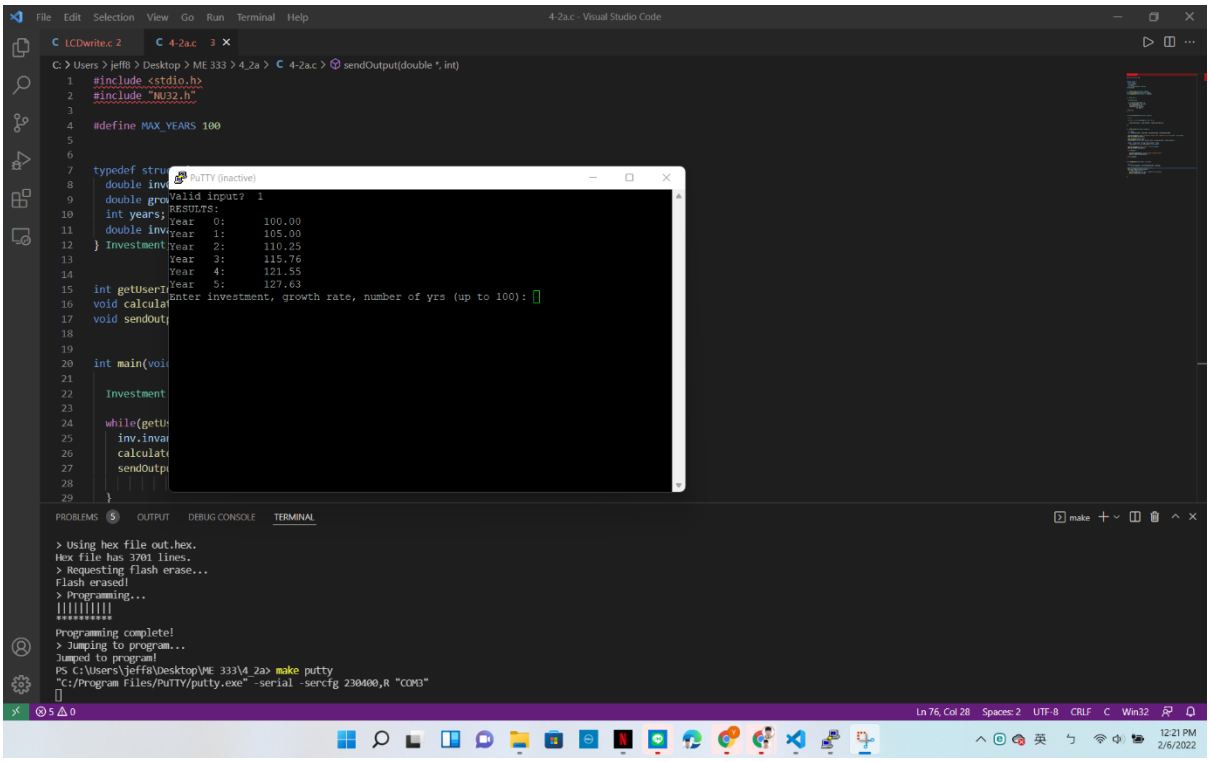
```
void NU32_Startup()

void NU32_ReadUART3(char * message, int maxLength)

void NU32_WriteUART3(const char * string)
```

Exercise 4.2

(a) The code is in 4-2a zip file



(b) The code is in 4-2b zip file

In helper.c file, there are three functions

```
int getUserInput(Investment *invp);
void calculateGrowth(Investment *invp);
void sendOutput(double *arr, int years);
```

(c) The code is in 4-2c zip file

In io.c there are two functions

```
int getUserInput(Investment *invp);
void sendOutput(double *arr, int years);
```

In calculate.c there are one function

```
void calculateGrowth(Investment *invp);
```

Exercise 4.4

```
void LCD_ClearLine(int ln) {
        char c = "";
        LCD_Move(ln, 0);

        for (int i = 1, i <= 16, i++) {
                LCD_WriteChar(c);
        }
}
```