#include <avr/io.h>
#include "uart.h"

void init();
//Function prototype for our init() function. The definition follows main().
// Althouh not nessacary, it is good pratice to keep all of your
// initalization calls in this function and call it at the beginning of your
// main() so your code is more readable.

void printf_example();
//Shows how to use printf to output different data types in a couple of
// different formats.

int main(void)
{
  //Initialize the input and output ports.
  init();

  //Initalize the serial port (UART)
  // this function is defined in the uart.c and uart.h files
  uart_init();

  printf_example();

  while (1)
  {
    PORTB = PINA; //Reads the buttons and sets the LEDs accordingly.
  }

  return 0;
}

void init()
{
  DDRA = 0x00; //Configure PORTA to be an input.
  PORTA = 0xFF;
  DDRB = 0xFF; //configure PORTD to be an output.
}

void printf_example()
{
  //A couple of different data types
  char single_letter = 'Z';
  char cstr[] = "This is a string."
  uint8_t the_answer = 42;
  uint16_t days_in_year = 365;
  uint32_t jenny = 8675309;
  double pi = 3.14159265358979323846;

  printf("This is a single characte: %c\r\n", single_letter);
  printf("%s\r\n", cstr);
  printf("An 8bit unsigned int in decimal format: %u\r\n", the_answer);
  printf("An 8bit unsigned int in hexadecimal format: %X\r\n",
         the_answer);
  printf("A 16bit unsigned int: %u\r\n", days_in_year);
printf("A 32bit unsigned int: %lu\n", jenny);
printf("A 32bit floating point number: %f\n", pi);