```c
// ADC.c
// Runs on LM4F120/TM4C123
// Provide functions that initialize ADC0
// Last Modified: 3/6/2015
// Student names: change this to your names or look very silly
// Last modification date: change this to the last modification date or look very silly

#include <stdint.h>
#include "tm4c123gh6pm.h"

// ADC initialization function
// Input: none
// Output: none
void ADC_Init(void){
      uint32_t delay;
      SYSCTL_RCGCGPIO_R |= 0x10;                  // 1) activate clock on portE
      while((SYSCTL_PRGPIO_R&0x10) != 0x10){};
      delay = SYSCTL_RCGCGPIO_R;                  // 2) extra time for clock to stabilize
      delay = SYSCTL_RCGCGPIO_R;
      //      Ain1 is on PE2
      GPIO_PORTE_DIR_R &= ~0x04;                  // 3) make PE2 input
      GPIO_PORTE_AFSEL_R |= 0x04;                 // 4) enable alternate function on PE2
      GPIO_PORTE_DEN_R &= ~0x04;                  // 5) disable digital I/O on PE2
      GPIO_PORTE_AMSEL_R |= 0x04;                 // 6) enable analog functionality on PE2

      SYSCTL_RCGCADC_R |= 0x0001;                 // 7) activate ADC0
//      while((SYSCTL_PRADC_R&0x0001) != 0x0001){};
      delay = SYSCTL_RCGCADC_R;                   // extra time for clock to stabilize
      delay = SYSCTL_RCGCADC_R;                   // extra time for clock to stabilize
      delay = SYSCTL_RCGCADC_R;                   // extra time for clock to stabilize
      delay = SYSCTL_RCGCADC_R;
      ADC0_PC_R &= ~0xF;                          // 9) clear max sample rate field
      ADC0_PC_R |= 0x1;                           //    configure for 125K samples/sec
      ADC0_SSPRI_R = 0x0123;                      // 10) Sequencer 3 is lowest priority
      ADC0_ACTSS_R &= ~0x0008;                    // 11) disable sample sequencer 3
      ADC0_EMUX_R |= 0xF000;                      // 12) seq3 is continuous trigger
      ADC0_SSMUX3_R &= ~0x000F;                   // 13) clear SS3 field
      ADC0_SSMUX3_R += 0x01;                      // set channel
      ADC0_SSCTL3_R = 0x0006;                     // 14) no TS0 D0, yes IE0 END0
      ADC0_IM_R &= ~0x0008;                       // 15) disable SS3 interrupts
      ADC0_ACTSS_R |= 0x0008;                     // 16) enable sample sequencer 3

}

//------------ADC_In------------
// Busy-wait Analog to digital conversion
// Input: none
// Output: 12-bit result of ADC conversion
uint32_t ADC_In(void){
  uint32_t data;
  ADC0_PSSI_R = 0x0008;               // 1) initiate SS3
  while((ADC0_RIS_R & 0x08) == 0){}   // 2) wait for conversion done
  data = ADC0_SSFIFO3_R & 0xFFF;      // 3) read result
  ADC0_ISC_R = 0x0008;                // 4) acknowledge completion
  return data;
}
```