/\*  
 \* Opdracht 3.1 - ADC Baby!  
 \*   
 \* Eigenaar:  
 \* Maarten Paauw <s1094220@student.hsleiden.nl>  
 \* s1094220  
 \* INF3C  
 \*   
 \* Versie: 1  
 \* Aangemaakt: 15 februari 2018  
 \* Gewijzigd: 27 maart 2018  
 \*/  
  
#include <avr/io.h>  
#include <util/delay.h>  
#include <stdlib.h>  
  
// Genereer een random nummer tussen 1 en 6.  
int dobbel()  
{  
 // Genereer een random nummer tussen 1 en 6.  
 return (rand() % 6) + 1;  
}  
  
// Regel de pinnen op de C bank.  
void handleC(int number)  
{  
 // Regel de pinnen op de C bank.  
 PORTC = ~number;  
}  
  
// Regel de pinnen op de D bank.  
void handleD(int number)  
{  
  
 // Door de RX en TX pinnen begin ik pas op pin D2.  
 // Daarom wordt er even 2x naar links gebitshift.  
 PORTD = (number << 2);  
}  
  
// Zet alle poorten uit.  
void clear()  
{  
 // Zet alle pinnen op de C bank op 0.  
 PORTC = 0b0000000;  
  
 // Zet alle pinnen op de D bank op 0.  
 PORTD = 0b0000000;  
}  
  
// Functie voor een enkel oog.  
void dot(int number)  
{  
 // Het getal min 1 voor de index uit de array.  
 number = number - 1;  
  
 // Alle 9 ogen met de daarbij behorende pinnen.  
 int dots[9][2] = {  
 {0b00000011, 0b00000011}, // LINKS BOVEN  
 {0b00001100, 0b00000011}, // MIDDEN BOVEN  
 {0b00110000, 0b00000011}, // RECHTS BOVEN  
 {0b00000011, 0b00001100}, // LINKS MIDDEN  
 {0b00001100, 0b00001100}, // MIDDEN MIDDEN  
 {0b00110000, 0b00001100}, // RECHTS MIDDEN  
 {0b00000011, 0b00110000}, // LINKS ONDER  
 {0b00001100, 0b00110000}, // MIDDEN ONDER  
 {0b00110000, 0b00110000} // RECHTS ONDER  
 };  
  
 // Zet de C pinnen aan.  
 handleC(dots[number][0]);  
  
 // Zet de D pinnen aan.  
 handleD(dots[number][1]);  
}  
  
// Functie voor een enkele zijde.  
void side(int number)  
{  
  
 // Alle zijdes van een dobbelsteen met welke ogen er nodig zijn.  
 int sides[6][6] = {  
 {5}, // 1  
 {1, 9}, // 2  
 {1, 5, 9}, // 3  
 {1, 3, 7, 9}, // 4  
 {1, 3, 5, 7, 9}, // 5  
 {1, 3, 4, 6, 7, 9} // 6  
 };  
  
 // Loop door de ogen heen.  
 for (int i = 0; i < number; i++)  
 {  
 // Zet alle pinnen uit.  
 clear();  
  
 // Zet de dot aan.  
 dot(sides[number - 1][i]);  
  
 // Wacht voor 1 miliseconden.  
 \_delay\_ms(1);  
 }  
}  
  
// De main functie.  
int main(void)  
{  
 // Zet de B pinnen op input.  
 PORTB = (1 << PB0);  
  
 // Zet de C pinnen op output.  
 DDRC = 0b00111111;  
  
 // Zet de D pinnen op output.  
 DDRD = 0b11111100;  
  
 // Toon standaard 1.  
 int res = 1;  
  
 // Loop voor altijd.  
 while (1)  
 {  
 // Controleer of de knop is ingedrukt.  
 if (bit\_is\_clear(PINB, PB0))  
 {  
  
 // Genereer een random getal.  
 res = dobbel();  
  
 // Leeg het display.  
 clear();  
  
 // Loop net zolang dat de button losgelaten is.  
 loop\_until\_bit\_is\_set(PINB, PB0);  
 }  
  
 // Toon het resultaat.  
 side(res);  
 }  
  
 // Geef een 0 terug.  
 return 0;  
}