

```
//***** glob_header.h *****
```

```
#ifndef GLOB_HEADER_H_  
#define GLOB_HEADER_H_
```

```
#define F_CPU 8000000UL  
#include <util/delay.h>  
#include <avr/io.h>  
#include <avr/interrupt.h>  
#include <avr/eeprom.h>
```

```
#endif /* GLOB_HEADER_H_ */
```

```
//***** main.c *****
```

```
#include "glob_header.h"  
#include "display.h"  
#include "gameplay.h"  
#include "button.h"  
#include "LEDs.h"  
#include "highscore.h"
```

```
volatile uint8_t count;  
extern volatile uint8_t val;  
uint8_t crnt_game_state = GAME_STATE_START;  
uint8_t crnt_lvl = 1;  
char score[2];
```

```
volatile uint8_t seed_index;  
volatile uint8_t seed_row;  
volatile uint8_t crnt_index;  
char player1_name[3];  
char player2_name[3];
```

```
int main(void) {  
    init_button();  
    init_gameplay();  
    all_lights_init();  
    sei();  
    count = 0;  
    while (1) {  
        switch(crnt_game_state){  
            case GAME_STATE_START:
```

```

all_lights_off();
writeString(" PRESS TO START");
    while(crnt_game_state == GAME_STATE_START){
        light_toggle(5);
        _delay_ms(100);
        if(val == 5) crnt_game_state = GAME_STATE_PLAYING;
    }
break;
case GAME_STATE_PLAYING:
    while (crnt_game_state == GAME_STATE_PLAYING){

        start_level(crnt_lvl);
        _delay_ms(10);
        crnt_index = 0;

        while(crnt_index < crnt_lvl+3){
            val = 0;
            while(val == 0){ }

            if (val == temp_array[crnt_index]){
                if(crnt_index < crnt_lvl+2){

                    writeString("OK");
                    _delay_ms(200);
                    crnt_index++;
                }else{
                    writeString("OK, next level");
                    _delay_ms(1000);
                    crnt_index++;
                    crnt_lvl++;
                    break;
                }
            } else {

                writeString("GAME OVER!");
                _delay_ms(1000);
                crnt_game_state = GAME_STATE_OSER;
                break;
            }
        }
    }
break;
case GAME_STATE_OSER:

```

```

while (crnt_game_state == GAME_STATE_OSER)
{
    if (crnt_lvl >= eeprom_read_byte(11)){
        crnt_game_state = GAME_STATE_HIGHSCORE;
    }
    else{
        itoa((crnt_lvl-1), score, 10);
        writeString("OOOZER!");
        _delay_ms(1000);
        writeString("You got ");
        writeString_no_reset(score);
        writeString_no_reset(" points");
        _delay_ms(700);
        clearDisplay();
        highscore_display();
        crnt_game_state = GAME_STATE_START;
        crnt_lvl = 1;
    }
}
break;
case GAME_STATE_HIGHSCORE:
    for (uint8_t i = 0; i<5; i++){

        _delay_ms(200);
        all_lights_on();
        writeString(" HIGH SCORE!");
        _delay_ms(300);
        all_lights_off();
    }

    itoa((crnt_lvl-1), score, 10);
    writeString(" NEW HIGH SCORE");
    _delay_ms(600);
    writeString("You got ");
    writeString_no_reset(score);
    writeString_no_reset(" points");
    _delay_ms(700);
    highscore_add((crnt_lvl-1);
    _delay_ms(1000);
    highscore_display();
    _delay_ms(2500);

```



```

#define btn_4 0b11110111 //PD3
#define btn_5 0b11101111 //PD4
#define btn_6 0b11011111 //PD5
#define btn_7 0b10111111 //PD6
#define btn_8 0b01111111 //PD7

uint8_t btn_state;

void init_button(void);
void timer0_start(void);
uint8_t check_button(void);

#endif /* BUTTON_H_ */

//***** button.c *****

#include "button.h"

void init_button() {

    timer0_start();
    DDRD &= ~(1<<0);
    DDRD &= ~(1<<1);
    DDRD &= ~(1<<2);
    DDRD &= ~(1<<3);
    DDRD &= ~(1<<4);
    DDRD &= ~(1<<5);
    DDRD &= ~(1<<6);
    DDRD &= ~(1<<7);
}

void timer0_start() {
    TIMSK0 |= 1<<TOIE0;
    TCCR0B |= (1<<CS02) | (1<<CS00);
    TCNT0 = 0;
}

uint8_t check_button(){
    btn_state = PIND;
    uint8_t pressed_btn;
    switch(btn_state){
        case btn_1:
            PORTA = 0b00000001;

```

```

        pressed_btn = 1;
    break;
    case btn_2:
        PORTA = 0b00000010;
        pressed_btn = 2;
    break;
    case btn_3:
        PORTA = 0b0000100;
        pressed_btn = 3;
    break;
    case btn_4:
        PORTA = 0b00001000;
        pressed_btn = 4;
    break;
    case btn_5:
        PORTA = 0b00010000;
        pressed_btn = 5;
    break;
    case btn_6:
        PORTA = 0b00100000;
        pressed_btn = 6;
    break;
    case btn_7:
        PORTA = 0b01000000;
        pressed_btn = 7;
    break;
    case btn_8:
        PORTA = 0b10000000;
        pressed_btn = 8;
        break;
    default:
        pressed_btn = 0;
    break;
}
return pressed_btn;
}

```

```

//***** display.h *****

```

```

#ifndef DISPLAY_H_
#define DISPLAY_H_

```

```

#include "glob_header.h"

```

```
#define DISPLAY_CNTRL_PORT    PORTC
#define DISPLAY_CNTRL_DDR    DDRC
```

```
#define RS_PIN    (0)
#define RW_PIN    (6)
#define E_PIN     (7)
```

```
#define FUNCTION_SET    0b00111000
```

```
void rs_high(void);
void rs_low(void);
void rw_high(void);
void rw_low(void);
void e_high(void);
void e_low(void);
void displayOn(void);
void clearDisplay(void);
void returnHome(void);
void writeChar(char);
void writeString(char*);
void writeString_no_reset(char*);
void displayCommand(char);
void start(void);
```

```
#endif /* DISPLAY_H_ */
```

```
/** ***** display.c *****
```

```
#include "display.h"
```

```
void rs_high(void){
    DISPLAY_CNTRL_PORT |= (1<<RS_PIN);
}
void rs_low(void){
    DISPLAY_CNTRL_PORT &= ~(1<<RS_PIN);
}
void rw_high(void){
    DISPLAY_CNTRL_PORT |= (1<<RW_PIN);
}
void rw_low(void){
    DISPLAY_CNTRL_PORT &= ~(1<<RW_PIN);
}
```

```

void e_high(void){
    DISPLAY_CNTRL_PORT |= (1<<E_PIN);
}
void e_low(void){
    DISPLAY_CNTRL_PORT &= ~(1<<E_PIN);
}

```

```

void displayOn(void){
    DDRB = 0xff;
    DISPLAY_CNTRL_DDR |= (1 << RS_PIN);
    DISPLAY_CNTRL_DDR |= (1 << RW_PIN);
    DISPLAY_CNTRL_DDR |= (1 << E_PIN);
    rw_low();
    rs_low();
    e_high();
    displayCommand(0b00001100);
    displayCommand(FUNCTION_SET);
}

```

```

}
void clearDisplay(void) {
    _delay_ms(1);
    displayCommand(0b00000001);
}

```

```

void returnHome(void) {
    _delay_ms(1);
    displayCommand(0b00000010);
}

```

```

void writeChar(char c) {
    _delay_ms(1);
    rs_high();
    rw_low();
    e_high();
    PORTB = c;
    e_low();
    e_high();
}

```

```

}
void writeString(char string[]) {
    clearDisplay();
    returnHome();
    _delay_ms(1);
    int i = 0;
}

```



```

        while(string[i] !='\0'){
            writeChar(string[i]);
            i++;
        }
    }
}

```

```

void writeString_no_reset(char string[]) {
    _delay_ms(1);
    int i = 0;
    while(string[i] !='\0') {
        writeChar(string[i]);
        i++;
    }
}

```

```

void displayCommand(char c) {
    rs_low();
    rw_low();
    e_high();
    PORTB = c;
    e_low();
    e_high();
}

```

```

void start(void) {
    displayOn();
    clearDisplay();
    writeString("test");
}

```

```

//***** game_matrix.h *****

```

```

#ifndef GAME_MATRIX_H_
#define GAME_MATRIX_H_

```

```

const uint8_t game_matrix[30][200] =

```

```

{{8, 4, 6, 2, 1, 3, 7, 2, 4, 3, 4, 4, 1, 4, 5, 5, 4, 6, 7, 1, 6, 6, 7, 1, 1, 6, 4, 6, 6, 5, 7, 8, 1, 8, 7, 4, 2, 2, 8, 3, 8, 2, 3, 4, 7, 1, 7, 6, 3, 6, 2, 7,
1, 3, 8, 3, 1, 3, 3, 4, 6, 5, 4, 7, 2, 1, 4, 3, 2, 8, 3, 3, 2, 6, 4, 6, 5, 8, 2, 2, 8, 6, 1, 3, 2, 8, 5, 8, 4, 1, 1, 3, 2, 2, 5, 8, 8, 6, 4, 4, 3, 7, 5, 3,
3, 2, 6, 2, 5, 1, 5, 8, 6, 6, 7, 1, 3, 4, 7, 8, 7, 1, 2, 6, 4, 6, 1, 5, 3, 5, 4, 3, 3, 1, 5, 8, 1, 7, 6, 7, 8, 1, 6, 6, 3, 6, 3, 2, 2, 7, 4, 5, 4, 4, 1, 1,
6, 3, 7, 5, 2, 7, 2, 2, 5, 6, 5, 8, 7, 5, 7, 7, 1, 2, 6, 7, 3, 5, 5, 3, 3, 3, 4, 3, 1, 4, 8, 2, 1, 2, 4, 5, 2, 8, 1, 1, 4, 5, 5, 7},

```

```

{7, 1, 7, 7, 8, 6, 8, 6, 4, 3, 7, 3, 4, 1, 3, 5, 5, 4, 3, 8, 6, 2, 2, 4, 7, 7, 8, 7, 7, 4, 2, 2, 5, 5, 4, 4, 6, 2, 6, 7, 8, 2, 3, 2, 2, 1, 3, 8, 5, 8, 7, 5,
7, 3, 3, 6, 2, 4, 5, 5, 6, 3, 1, 7, 3, 1, 3, 8, 3, 7, 2, 4, 4, 8, 8, 4, 1, 2, 1, 3, 8, 6, 4, 1, 3, 5, 7, 1, 3, 3, 7, 2, 4, 7, 1, 8, 4, 1, 1, 2, 2, 6, 8, 5,
4, 2, 7, 7, 5, 8, 8, 1, 1, 4, 7, 5, 3, 2, 2, 1, 5, 2, 6, 3, 8, 3, 2, 1, 8, 8, 2, 6, 8, 7, 3, 5, 1, 6, 7, 6, 2, 5, 5, 7, 1, 1, 1, 5, 3, 2, 2, 1, 4, 3, 3, 5,
1, 2, 8, 2, 8, 3, 5, 3, 5, 3, 2, 4, 3, 1, 2, 6, 1, 5, 1, 8, 1, 8, 2, 3, 7, 2, 5, 2, 5, 1, 2, 2, 7, 8, 4, 6, 2, 8, 1, 4, 3, 5, 6, 7},

```

{4, 1, 7, 6, 7, 2, 4, 5, 2, 3, 4, 8, 7, 6, 7, 4, 8, 7, 7, 6, 6, 3, 7, 8, 7, 6, 6, 3, 4, 2, 8, 3, 6, 2, 8, 2, 7, 6, 1, 5, 6, 4, 1, 7, 8, 7, 6, 7, 5, 7, 6, 5, 7, 2, 7, 3, 8, 4, 6, 8, 3, 6, 5, 3, 6, 8, 7, 1, 8, 4, 3, 4, 1, 7, 2, 4, 8, 8, 1, 1, 1, 7, 3, 3, 8, 5, 7, 6, 6, 3, 4, 2, 6, 6, 5, 3, 1, 5, 4, 2, 1, 1, 4, 8, 2, 6, 2, 5, 6, 6, 5, 5, 6, 6, 5, 4, 1, 2, 3, 7, 4, 7, 5, 2, 5, 8, 6, 5, 8, 7, 8, 1, 2, 4, 5, 1, 6, 8, 7, 4, 1, 2, 4, 4, 4, 3, 2, 4, 6, 3, 2, 3, 5, 4, 4, 6, 1, 3, 2, 2, 8, 7, 8, 8, 3, 6, 2, 4, 2, 3, 5, 2, 2, 7, 8, 2, 7, 4, 7, 1, 6, 8, 6, 5, 7, 1, 4, 5, 8, 8, 7, 1, 1, 3, 7, 6, 7, 6, 3},

{6, 1, 3, 2, 1, 1, 3, 7, 1, 2, 2, 8, 2, 1, 7, 1, 2, 8, 2, 4, 7, 7, 6, 1, 3, 8, 8, 6, 1, 2, 3, 4, 1, 4, 1, 7, 5, 1, 3, 4, 4, 5, 5, 5, 6, 7, 8, 1, 2, 5, 1, 5, 3, 5, 1, 6, 2, 5, 7, 3, 3, 7, 7, 8, 8, 3, 6, 2, 6, 1, 4, 2, 1, 7, 2, 7, 8, 8, 4, 3, 4, 1, 2, 6, 4, 1, 8, 8, 6, 7, 4, 7, 3, 3, 7, 6, 5, 2, 3, 2, 1, 6, 6, 4, 3, 5, 7, 5, 8, 5, 4, 5, 1, 1, 7, 6, 7, 2, 7, 5, 7, 5, 5, 1, 1, 7, 2, 6, 7, 4, 2, 5, 4, 6, 8, 1, 8, 2, 2, 6, 5, 5, 1, 1, 7, 4, 5, 1, 2, 8, 1, 2, 1, 6, 1, 2, 8, 4, 6, 6, 1, 7, 6, 4, 6, 1, 3, 6, 4, 5, 4, 3, 2, 2, 4, 2, 8, 1, 7, 1, 3, 8, 7, 1, 6, 1, 3, 2, 6, 6, 3, 4, 8, 1, 8, 1, 8, 2, 6},

{4, 5, 3, 4, 4, 2, 6, 5, 1, 4, 5, 5, 8, 8, 7, 6, 3, 4, 5, 5, 5, 5, 1, 6, 1, 8, 5, 2, 3, 4, 4, 2, 1, 6, 8, 5, 2, 4, 1, 3, 1, 3, 3, 1, 5, 5, 2, 8, 1, 1, 2, 6, 8, 7, 4, 8, 4, 3, 4, 6, 1, 8, 2, 6, 1, 4, 5, 8, 7, 4, 7, 5, 8, 8, 7, 4, 1, 2, 5, 4, 5, 1, 8, 4, 4, 2, 3, 3, 5, 2, 2, 7, 8, 3, 5, 2, 3, 7, 5, 5, 3, 2, 1, 7, 4, 3, 7, 7, 4, 5, 7, 5, 8, 4, 3, 2, 2, 1, 1, 4, 4, 5, 2, 1, 8, 3, 3, 4, 6, 1, 7, 2, 2, 3, 6, 8, 8, 6, 3, 2, 2, 7, 2, 5, 7, 1, 2, 1, 3, 2, 7, 7, 3, 5, 7, 8, 4, 4, 6, 2, 6, 3, 3, 2, 6, 1, 1, 1, 8, 2, 5, 2, 8, 6, 3, 2, 5, 6, 6, 1, 8, 5, 5, 7, 4, 2, 2, 7, 6, 3, 6, 3, 8, 5, 4, 5, 7, 3, 3, 6},

{7, 2, 5, 4, 3, 2, 8, 8, 8, 6, 4, 8, 6, 6, 3, 5, 5, 1, 5, 7, 2, 8, 7, 6, 4, 3, 7, 8, 6, 5, 7, 2, 5, 5, 5, 3, 6, 6, 6, 2, 5, 5, 4, 8, 8, 7, 4, 4, 4, 1, 3, 1, 8, 3, 1, 1, 8, 4, 1, 3, 1, 3, 1, 2, 6, 3, 8, 5, 1, 4, 7, 4, 6, 8, 4, 4, 3, 1, 2, 3, 6, 7, 2, 2, 8, 2, 4, 8, 3, 3, 7, 5, 2, 3, 4, 7, 7, 2, 7, 1, 5, 5, 2, 4, 2, 4, 4, 6, 2, 3, 6, 2, 1, 4, 8, 4, 7, 5, 6, 3, 1, 3, 8, 7, 4, 6, 3, 1, 6, 2, 7, 3, 8, 8, 4, 4, 8, 7, 8, 8, 2, 5, 1, 2, 2, 7, 1, 7, 4, 7, 2, 6, 3, 7, 6, 7, 3, 5, 6, 4, 2, 6, 4, 7, 2, 8, 5, 1, 3, 8, 7, 1, 2, 1, 2, 2, 4, 3, 7, 8, 6, 4, 5, 2, 5, 5, 6, 5, 6, 8, 1, 5, 8, 7, 6, 7, 3, 7, 4, 7},

{3, 7, 3, 5, 6, 5, 5, 4, 2, 2, 8, 4, 4, 8, 1, 4, 4, 5, 6, 1, 4, 1, 8, 5, 7, 5, 4, 6, 7, 3, 6, 6, 7, 5, 3, 6, 8, 1, 1, 3, 5, 7, 8, 4, 5, 5, 1, 8, 7, 5, 4, 4, 8, 7, 3, 4, 2, 1, 3, 8, 7, 7, 2, 4, 1, 5, 7, 8, 3, 8, 6, 1, 4, 6, 5, 7, 4, 3, 6, 8, 8, 6, 6, 7, 1, 3, 4, 8, 4, 7, 2, 2, 6, 2, 1, 6, 8, 8, 1, 5, 5, 8, 8, 4, 6, 2, 4, 1, 7, 1, 4, 6, 6, 4, 6, 5, 4, 7, 3, 1, 7, 3, 8, 1, 8, 1, 8, 2, 1, 1, 2, 4, 8, 6, 2, 4, 4, 3, 3, 8, 4, 3, 6, 3, 7, 7, 4, 2, 3, 5, 2, 3, 3, 6, 7, 1, 2, 1, 8, 6, 3, 1, 8, 4, 7, 6, 3, 7, 8, 7, 1, 2, 1, 2, 5, 1, 3, 1, 4, 1, 7, 7, 8, 4, 7, 7, 2, 7, 7, 8, 4, 6, 5, 8, 7, 3, 2, 6, 8, 7},

{1, 7, 7, 7, 2, 1, 8, 3, 7, 6, 8, 4, 2, 5, 6, 1, 3, 6, 4, 6, 7, 4, 4, 4, 7, 6, 5, 2, 3, 7, 2, 2, 7, 7, 5, 1, 2, 2, 6, 2, 3, 1, 8, 1, 2, 5, 2, 7, 2, 8, 5, 2, 2, 4, 8, 1, 4, 2, 7, 3, 3, 7, 6, 4, 7, 7, 2, 3, 8, 5, 7, 1, 6, 1, 1, 6, 6, 5, 5, 4, 6, 5, 2, 6, 2, 5, 1, 2, 4, 3, 8, 7, 1, 3, 4, 1, 6, 8, 2, 1, 7, 3, 1, 8, 8, 3, 4, 7, 8, 4, 5, 3, 4, 3, 2, 1, 4, 8, 1, 2, 7, 1, 6, 3, 3, 2, 6, 7, 5, 7, 6, 2, 2, 8, 1, 5, 8, 7, 2, 4, 8, 7, 2, 6, 1, 2, 3, 2, 8, 1, 8, 6, 1, 5, 5, 6, 3, 5, 5, 6, 2, 1, 8, 6, 2, 2, 4, 2, 7, 8, 7, 2, 1, 7, 5, 8, 6, 6, 6, 1, 2, 5, 7, 7, 6, 7, 6, 6, 3, 1, 8, 8, 4, 1, 3, 7, 3, 6, 6, 8},

{5, 8, 7, 1, 5, 7, 2, 4, 6, 4, 7, 2, 3, 2, 8, 3, 5, 3, 6, 6, 2, 5, 7, 5, 6, 4, 2, 4, 4, 8, 6, 5, 1, 4, 3, 1, 7, 3, 4, 3, 8, 6, 1, 1, 7, 2, 1, 3, 5, 7, 3, 3, 1, 5, 7, 6, 5, 3, 8, 3, 5, 1, 6, 5, 4, 7, 4, 5, 3, 7, 7, 4, 5, 8, 3, 5, 8, 7, 3, 6, 3, 6, 8, 1, 8, 5, 5, 6, 1, 1, 8, 8, 1, 7, 7, 5, 4, 3, 2, 4, 5, 4, 1, 6, 4, 6, 2, 2, 2, 3, 6, 2, 6, 2, 6, 4, 2, 4, 3, 6, 5, 7, 1, 3, 7, 2, 4, 3, 7, 4, 3, 6, 7, 5, 1, 2, 5, 1, 6, 7, 6, 2, 5, 2, 2, 8, 2, 4, 4, 8, 7, 1, 4, 6, 4, 3, 6, 4, 8, 3, 8, 5, 6, 3, 1, 8, 3, 7, 1, 7, 8, 3, 3, 5, 6, 8, 1, 7, 6, 7, 3, 8, 5, 2, 6, 3, 4, 6, 3, 1, 6, 3, 1, 8, 8, 8, 8, 2, 6, 7},

{8, 4, 5, 7, 5, 6, 1, 3, 7, 4, 6, 2, 7, 3, 2, 2, 8, 4, 8, 4, 3, 5, 2, 4, 8, 7, 6, 6, 7, 6, 2, 3, 6, 6, 2, 5, 2, 2, 3, 1, 5, 6, 5, 8, 7, 5, 1, 6, 2, 8, 7, 5, 2, 8, 2, 8, 5, 6, 5, 8, 8, 4, 6, 5, 1, 7, 7, 8, 1, 6, 1, 8, 6, 4, 5, 1, 4, 5, 6, 6, 8, 7, 5, 7, 1, 4, 5, 2, 5, 8, 5, 2, 2, 2, 8, 8, 1, 5, 3, 3, 1, 1, 5, 7, 2, 6, 7, 5, 7, 7, 5, 7, 1, 6, 1, 7, 1, 1, 1, 5, 5, 2, 5, 6, 1, 7, 2, 4, 2, 4, 1, 8, 1, 1, 8, 6, 7, 3, 3, 5, 6, 3, 1, 7, 1, 6, 5, 3, 2, 8, 7, 7, 8, 7, 7, 8, 1, 5, 8, 7, 5, 6, 2, 5, 3, 4, 5, 7, 2, 4, 7, 8, 7, 6, 7, 7, 3, 2, 2, 3, 1, 2, 5, 6, 7, 7, 3, 7, 3, 2, 4, 2, 7, 2, 6, 6, 2, 2, 8, 8},

{2, 2, 3, 7, 5, 8, 3, 8, 8, 2, 6, 7, 1, 7, 1, 2, 1, 2, 1, 5, 8, 7, 7, 1, 8, 2, 8, 8, 5, 4, 8, 4, 8, 2, 6, 6, 7, 3, 5, 7, 5, 4, 3, 7, 3, 1, 6, 7, 7, 8, 7, 6, 3, 6, 5, 1, 6, 5, 8, 4, 6, 8, 4, 3, 7, 2, 2, 5, 2, 4, 5, 7, 4, 5, 1, 6, 3, 4, 7, 8, 5, 8, 2, 4, 8, 3, 1, 8, 5, 8, 7, 5, 1, 6, 7, 3, 5, 4, 8, 6, 6, 3, 1, 3, 4, 5, 5, 8, 8, 1, 1, 7, 8, 5, 2, 7, 5, 1, 6, 5, 8, 4, 6, 2, 8, 1, 5, 7, 1, 4, 4, 6, 7, 5, 8, 4, 3, 3, 7, 1, 6, 4, 1, 7, 8, 1, 7, 3, 7, 2, 4, 6, 5, 4, 2, 5, 1, 8, 3, 8, 2, 1, 8, 3, 8, 3, 2, 2, 4, 1, 5, 1, 8, 3, 7, 5, 3, 2, 2, 5, 8, 6, 3, 2, 1, 8, 2, 1, 3, 3, 1, 6, 2, 4, 4, 8, 3, 8, 3},

{4, 2, 6, 3, 4, 4, 4, 2, 2, 8, 6, 6, 8, 8, 7, 4, 8, 2, 3, 7, 2, 8, 3, 1, 1, 5, 5, 5, 5, 8, 8, 4, 1, 6, 5, 7, 7, 6, 8, 3, 8, 6, 5, 3, 1, 4, 1, 7, 3, 3, 4, 2, 8, 3, 5, 3, 6, 5, 2, 4, 4, 3, 3, 2, 1, 7, 7, 6, 6, 1, 8, 5, 5, 8, 5, 8, 4, 2, 3, 4, 7, 6, 1, 3, 3, 4, 3, 5, 2, 6, 2, 4, 4, 8, 7, 5, 4, 1, 3, 1, 6, 2, 1, 1, 5, 1, 1, 1, 3, 1, 1, 7, 7, 6, 7, 2, 5, 4, 5, 2, 1, 6, 7, 1, 8, 1, 2, 1, 1, 7, 8, 3, 6, 2, 1, 2, 1, 8, 3, 2, 2, 7, 6, 6, 7, 3, 7, 4, 2, 1, 2, 8, 1, 5, 2, 1, 3, 1, 2, 6, 1, 6, 2, 8, 1, 4, 1, 1, 2, 5, 2, 7, 8, 5, 3, 4, 4, 7, 8, 3, 3, 6, 3, 1, 1, 2, 3, 7, 8, 3, 1, 1, 7, 3, 2, 2, 4, 7},

{6, 5, 2, 4, 1, 6, 4, 3, 4, 5, 1, 5, 5, 3, 4, 4, 1, 4, 7, 2, 2, 3, 7, 5, 6, 1, 4, 6, 3, 6, 5, 2, 1, 3, 7, 8, 5, 3, 7, 2, 1, 1, 5, 1, 7, 3, 6, 1, 5, 3, 5, 5, 7, 1, 8, 5, 4, 7, 8, 1, 6, 3, 6, 6, 1, 8, 8, 2, 7, 4, 1, 3, 5, 5, 6, 4, 4, 7, 2, 7, 6, 6, 4, 6, 1, 6, 2, 3, 3, 4, 7, 4, 4, 1, 2, 6, 1, 3, 4, 4, 3, 7, 8, 5, 1, 3, 3, 8, 4, 3, 8, 4, 6, 1, 4, 7, 4, 3, 1, 7, 2, 5, 4, 3, 6, 2, 4, 8, 4, 8, 1, 7, 4, 7, 1, 6, 4, 8, 5, 1, 8, 7, 6, 7, 8, 4, 2, 1, 5, 5, 6, 1, 3, 5, 8, 4, 4, 4, 8, 6, 6, 5, 3, 8, 2, 3, 8, 7, 8, 7, 3, 3, 2, 7, 3, 6, 5, 6, 8, 7, 4, 8, 3, 3, 8, 7, 3, 4, 7, 1, 7, 1, 6, 5, 8, 8, 4, 4, 7, 8},

{1, 7, 4, 5, 5, 8, 3, 4, 8, 3, 3, 7, 6, 7, 7, 1, 5, 5, 4, 4, 8, 1, 3, 6, 4, 2, 7, 6, 4, 2, 6, 1, 2, 7, 3, 6, 4, 7, 3, 6, 3, 5, 3, 3, 4, 6, 7, 4, 4, 7, 1, 2, 2, 8, 1, 6, 7, 7, 7, 2, 6, 6, 5, 8, 3, 1, 5, 4, 8, 2, 1, 3, 4, 8, 4, 5, 6, 6, 4, 4, 5, 5, 5, 8, 2, 3, 3, 4, 8, 6, 5, 3, 7, 4, 6, 2, 3, 3, 6, 6, 3, 8, 6, 7, 7},

1, 3, 7, 5, 6, 8, 5, 1, 5, 5, 1, 5, 7, 8, 7, 4, 7, 4, 7, 8, 8, 5, 5, 4, 3, 2, 5, 8, 2, 7, 2, 6, 3, 4, 4, 3, 7, 3, 5, 2, 8, 7, 4, 7, 6, 1, 1, 1, 7, 7, 6, 4, 4, 7, 1, 6, 7, 5, 2, 3, 8, 5, 4, 6, 1, 3, 4, 7, 8, 7, 2, 1, 3, 5, 8, 7, 3, 1, 2, 8, 2, 5, 4, 3, 2, 3, 3, 3, 7, 3, 4, 6, 8, 8, 4, 2},

{8, 3, 4, 6, 4, 5, 6, 6, 6, 5, 1, 5, 8, 1, 7, 4, 5, 2, 4, 8, 1, 8, 2, 8, 1, 5, 1, 8, 7, 4, 8, 8, 6, 4, 3, 5, 8, 5, 3, 1, 2, 2, 4, 4, 1, 8, 7, 2, 4, 2, 6, 2, 4, 1, 7, 3, 8, 1, 2, 3, 2, 7, 6, 5, 3, 1, 5, 8, 1, 6, 2, 8, 1, 7, 6, 1, 8, 3, 1, 1, 8, 6, 3, 7, 4, 3, 7, 3, 7, 5, 2, 4, 5, 1, 4, 1, 5, 1, 5, 4, 6, 1, 3, 3, 5, 2, 1, 6, 6, 4, 5, 8, 5, 5, 1, 6, 2, 1, 8, 6, 7, 7, 3, 5, 8, 5, 8, 3, 8, 2, 2, 1, 5, 1, 6, 8, 4, 2, 4, 8, 4, 5, 5, 7, 1, 6, 3, 5, 7, 5, 5, 6, 4, 5, 8, 7, 6, 7, 1, 4, 6, 3, 4, 5, 1, 2, 2, 8, 2, 6, 1, 8, 3, 2, 1, 1, 1, 1, 3, 5, 5, 6, 2, 1, 8, 6, 6, 5, 6, 4, 1, 1, 7, 4, 7, 2, 2, 6, 4},

{7, 4, 5, 6, 1, 3, 1, 4, 8, 8, 7, 5, 4, 5, 2, 7, 7, 1, 8, 2, 4, 7, 5, 8, 8, 5, 8, 2, 3, 3, 2, 6, 1, 8, 3, 3, 1, 5, 3, 5, 3, 1, 7, 1, 7, 6, 8, 1, 8, 8, 3, 3, 3, 1, 2, 5, 4, 6, 5, 8, 7, 7, 1, 5, 5, 6, 7, 6, 7, 4, 2, 7, 8, 4, 1, 5, 7, 5, 3, 7, 4, 8, 5, 5, 3, 3, 1, 3, 7, 4, 8, 3, 6, 6, 5, 4, 2, 4, 3, 2, 1, 7, 4, 7, 5, 7, 6, 4, 7, 4, 8, 6, 6, 2, 6, 3, 2, 7, 4, 1, 6, 8, 1, 5, 5, 8, 8, 5, 8, 2, 1, 3, 7, 4, 2, 8, 8, 7, 6, 2, 5, 7, 2, 8, 8, 1, 5, 5, 3, 7, 6, 2, 7, 2, 3, 2, 2, 5, 1, 6, 7, 3, 6, 6, 4, 2, 8, 8, 1, 6, 7, 1, 8, 4, 2, 5, 4, 8, 8, 8, 1, 1, 3, 6, 8, 2, 6, 2, 1, 1, 5, 5, 4, 1, 5, 8, 7, 3, 1, 8},

{5, 6, 8, 7, 4, 8, 4, 2, 4, 6, 5, 2, 3, 4, 2, 2, 2, 1, 7, 3, 4, 7, 4, 1, 4, 4, 1, 4, 4, 5, 7, 5, 4, 5, 3, 1, 5, 6, 6, 6, 7, 5, 6, 5, 4, 6, 7, 7, 8, 8, 6, 4, 3, 6, 6, 2, 3, 7, 6, 7, 5, 5, 6, 6, 6, 5, 8, 4, 7, 1, 5, 4, 4, 8, 1, 6, 2, 8, 5, 8, 8, 6, 7, 3, 6, 7, 3, 6, 5, 5, 5, 1, 4, 3, 2, 1, 2, 2, 6, 1, 7, 4, 2, 6, 3, 3, 6, 6, 5, 1, 3, 6, 3, 4, 2, 8, 1, 3, 2, 6, 2, 3, 4, 2, 7, 5, 2, 7, 1, 1, 8, 7, 6, 7, 1, 4, 6, 4, 5, 3, 2, 6, 2, 1, 7, 3, 3, 1, 7, 5, 2, 4, 8, 2, 5, 5, 4, 2, 8, 1, 2, 5, 1, 8, 2, 3, 1, 5, 3, 2, 7, 6, 8, 3, 5, 2, 5, 5, 8, 3, 7, 2, 5, 2, 1, 7, 1, 2, 7, 7, 6, 8, 8, 7, 4, 4, 6, 6, 6, 5},

{2, 8, 7, 4, 4, 1, 3, 7, 1, 6, 3, 5, 8, 6, 1, 8, 5, 7, 8, 3, 4, 2, 4, 6, 7, 3, 5, 3, 3, 8, 7, 5, 5, 3, 8, 6, 5, 7, 7, 1, 2, 7, 2, 3, 3, 6, 8, 3, 4, 3, 3, 2, 8, 3, 2, 6, 4, 6, 7, 6, 4, 6, 6, 3, 8, 5, 7, 5, 4, 5, 4, 8, 4, 4, 7, 4, 2, 2, 5, 1, 8, 7, 3, 3, 2, 5, 8, 8, 5, 2, 8, 6, 3, 5, 4, 2, 6, 4, 6, 3, 2, 3, 2, 6, 7, 8, 6, 4, 5, 5, 8, 3, 2, 7, 7, 2, 2, 4, 4, 7, 1, 8, 7, 8, 8, 8, 1, 2, 6, 6, 6, 7, 7, 3, 2, 3, 2, 5, 3, 8, 2, 6, 1, 2, 3, 4, 7, 5, 1, 3, 3, 7, 2, 5, 8, 8, 8, 6, 6, 8, 2, 3, 8, 4, 8, 7, 5, 7, 2, 1, 6, 8, 2, 2, 6, 1, 2, 5, 2, 5, 4, 7, 1, 4, 6, 1, 7, 5, 8, 2, 4, 8, 4, 5, 5, 3, 6, 2, 3, 1},

{4, 2, 8, 4, 5, 3, 3, 5, 5, 8, 2, 2, 7, 3, 5, 3, 4, 7, 2, 1, 1, 3, 8, 5, 1, 1, 6, 6, 6, 4, 7, 1, 8, 8, 7, 2, 6, 8, 7, 1, 2, 5, 2, 4, 3, 7, 6, 7, 5, 3, 6, 3, 1, 3, 2, 6, 3, 5, 5, 7, 6, 6, 8, 4, 8, 5, 3, 1, 1, 3, 5, 4, 1, 1, 3, 2, 5, 7, 1, 5, 4, 2, 3, 4, 8, 5, 4, 1, 3, 7, 4, 1, 3, 6, 3, 4, 5, 6, 2, 7, 2, 1, 8, 8, 6, 1, 1, 4, 3, 2, 5, 4, 7, 3, 4, 7, 3, 1, 7, 3, 6, 1, 3, 2, 8, 3, 1, 1, 4, 1, 8, 8, 3, 2, 1, 4, 2, 6, 4, 1, 6, 2, 8, 2, 1, 2, 8, 4, 3, 4, 7, 3, 5, 4, 7, 6, 1, 1, 5, 8, 3, 1, 4, 3, 6, 3, 2, 7, 5, 8, 4, 4, 3, 3, 2, 7, 5, 5, 8, 2, 2, 6, 2, 6, 6, 5, 3, 5, 3, 4, 5, 5, 6, 1, 5, 4, 2, 7, 2, 8},

{5, 6, 2, 8, 7, 8, 1, 3, 2, 7, 5, 1, 4, 1, 7, 3, 7, 2, 1, 5, 4, 2, 5, 4, 3, 7, 5, 8, 5, 1, 2, 8, 2, 8, 3, 3, 2, 2, 8, 2, 5, 7, 3, 5, 3, 6, 3, 2, 8, 6, 4, 8, 2, 7, 2, 4, 3, 1, 8, 4, 5, 4, 7, 5, 7, 1, 5, 2, 1, 7, 3, 4, 5, 3, 1, 7, 7, 2, 5, 7, 6, 7, 3, 3, 3, 2, 2, 3, 5, 7, 2, 2, 8, 6, 3, 6, 5, 2, 4, 6, 1, 7, 8, 7, 8, 5, 8, 8, 5, 4, 6, 1, 8, 4, 3, 3, 6, 1, 6, 5, 7, 1, 5, 1, 7, 2, 1, 2, 5, 1, 8, 5, 2, 1, 3, 1, 5, 1, 2, 1, 8, 6, 6, 5, 6, 5, 1, 3, 5, 5, 8, 3, 3, 5, 7, 1, 1, 6, 4, 5, 2, 6, 5, 5, 6, 5, 8, 5, 5, 7, 4, 7, 4, 6, 5, 4, 8, 5, 3, 5, 4, 7, 8, 1, 1, 1, 4, 8, 2, 3, 8, 7, 2, 5, 7, 1, 3, 7, 3, 1},

{8, 5, 5, 5, 6, 3, 5, 2, 5, 5, 3, 6, 1, 4, 5, 2, 6, 8, 4, 7, 5, 4, 8, 3, 6, 6, 4, 4, 6, 8, 4, 3, 1, 7, 5, 3, 1, 7, 7, 7, 5, 2, 7, 1, 7, 5, 6, 6, 5, 6, 4, 5, 4, 3, 8, 2, 2, 2, 6, 3, 7, 2, 8, 3, 2, 3, 1, 2, 1, 5, 4, 8, 3, 8, 7, 2, 1, 4, 1, 8, 2, 8, 8, 2, 1, 8, 5, 8, 4, 6, 4, 3, 6, 3, 8, 8, 7, 8, 2, 2, 8, 7, 2, 4, 3, 4, 4, 1, 8, 6, 2, 2, 7, 3, 3, 7, 8, 7, 5, 8, 1, 3, 6, 5, 6, 1, 3, 6, 4, 5, 1, 2, 1, 3, 4, 4, 5, 2, 6, 3, 5, 6, 4, 6, 1, 3, 8, 4, 2, 7, 4, 2, 1, 4, 6, 4, 5, 7, 3, 1, 1, 7, 2, 4, 8, 4, 3, 7, 5, 8, 8, 4, 3, 8, 1, 2, 6, 3, 3, 6, 6, 5, 8, 2, 1, 8, 8, 5, 7, 4, 4, 2, 5, 2, 6, 7, 8, 8},

{7, 4, 1, 7, 7, 3, 3, 2, 6, 8, 1, 5, 5, 6, 1, 7, 3, 6, 5, 8, 8, 7, 4, 4, 2, 4, 3, 7, 4, 8, 1, 6, 6, 3, 7, 5, 2, 2, 2, 3, 2, 5, 6, 1, 1, 3, 6, 8, 7, 7, 2, 8, 7, 5, 4, 4, 2, 3, 3, 4, 1, 4, 8, 4, 8, 1, 5, 7, 7, 2, 5, 4, 3, 4, 1, 3, 3, 1, 6, 1, 8, 4, 1, 1, 5, 6, 1, 7, 4, 7, 3, 3, 7, 3, 1, 3, 6, 8, 5, 8, 2, 7, 2, 1, 5, 5, 5, 1, 6, 6, 2, 7, 4, 4, 6, 5, 2, 1, 4, 5, 1, 8, 6, 8, 4, 5, 4, 2, 7, 1, 8, 3, 1, 5, 6, 7, 7, 8, 1, 5, 5, 3, 3, 6, 4, 7, 4, 8, 4, 2, 6, 2, 2, 8, 7, 7, 3, 5, 5, 2, 8, 4, 2, 4, 4, 7, 4, 3, 7, 2, 5, 1, 8, 5, 6, 1, 3, 1, 5, 6, 3, 2, 8, 2, 2, 2, 3, 6, 7, 8, 4, 1, 1, 5, 3, 5, 4, 8, 7, 4},

{8, 8, 7, 3, 1, 1, 2, 5, 5, 1, 1, 2, 2, 2, 8, 1, 3, 6, 5, 6, 2, 6, 4, 6, 6, 2, 5, 3, 4, 3, 3, 4, 1, 3, 7, 5, 7, 1, 7, 5, 6, 2, 6, 3, 6, 8, 7, 7, 6, 5, 6, 5, 2, 4, 6, 5, 4, 6, 4, 7, 5, 8, 2, 6, 4, 8, 1, 1, 4, 8, 1, 6, 6, 4, 1, 5, 3, 1, 7, 4, 1, 7, 2, 6, 7, 8, 1, 7, 6, 4, 1, 4, 4, 1, 5, 5, 3, 7, 5, 4, 2, 7, 5, 2, 7, 5, 1, 7, 7, 2, 8, 1, 1, 7, 5, 4, 7, 8, 4, 7, 3, 5, 7, 2, 1, 3, 8, 7, 7, 2, 7, 3, 3, 8, 8, 3, 4, 5, 5, 3, 5, 2, 1, 5, 7, 8, 1, 4, 4, 6, 5, 6, 3, 6, 4, 4, 8, 8, 6, 2, 5, 1, 1, 2, 7, 8, 2, 7, 1, 2, 1, 3, 6, 2, 3, 1, 5, 4, 4, 1, 1, 6, 6, 4, 2, 3, 2, 7, 7, 2, 3, 2, 1, 4, 4, 4, 2, 8, 6, 7},

{6, 4, 5, 5, 2, 8, 1, 4, 5, 3, 1, 4, 5, 4, 5, 8, 3, 8, 8, 4, 3, 6, 7, 8, 4, 7, 8, 5, 4, 3, 2, 1, 8, 7, 8, 7, 8, 2, 8, 5, 7, 3, 2, 5, 7, 2, 6, 2, 4, 1, 7, 3, 8, 3, 1, 3, 1, 2, 2, 4, 6, 7, 2, 4, 2, 1, 1, 5, 7, 2, 1, 1, 1, 6, 5, 2, 4, 8, 2, 2, 1, 4, 3, 3, 4, 1, 3, 5, 4, 6, 8, 5, 3, 6, 7, 8, 5, 4, 7, 7, 8, 3, 2, 6, 5, 1, 4, 3, 7, 6, 4, 3, 8, 6, 4, 8, 2, 4, 7, 5, 8, 1, 6, 3, 2, 7, 4, 7, 5, 3, 3, 5, 3, 7, 2, 6, 3, 8, 1, 6, 2, 1, 7, 6, 1, 4, 2, 1, 6, 6, 2, 5, 2, 6, 3, 6, 2, 6, 5, 1, 5, 3, 3, 4, 1, 4, 6, 8, 3, 7, 8, 8, 3, 5, 4, 8, 4, 3, 2, 1, 2, 4, 4, 5, 4, 5, 3, 3, 5, 7, 7, 2, 4, 6, 4, 7, 8, 1, 8, 8},

{4, 2, 7, 5, 4, 2, 3, 6, 4, 8, 4, 6, 3, 6, 3, 2, 8, 1, 4, 3, 3, 6, 6, 1, 7, 7, 6, 4, 1, 7, 4, 5, 2, 6, 5, 7, 4, 2, 7, 5, 7, 1, 1, 5, 3, 5, 3, 3, 2, 5, 8, 1, 8, 7, 2, 7, 8, 1, 2, 7, 5, 2, 1, 5, 6, 1, 6, 1, 2, 3, 1, 6, 7, 3, 4, 2, 2, 8, 8, 8, 5, 1, 3, 3, 8, 2, 7, 1, 6, 2, 6, 1, 5, 3, 3, 2, 7, 6, 2, 5, 3, 2, 2, 2, 7, 4, 2, 3, 1, 3, 1, 7, 7, 1, 3, 3, 6, 4, 8, 8, 7, 8, 3, 5, 1, 7, 7, 8, 1, 3, 8, 5, 5, 6, 5, 6, 3, 8, 7, 3, 7, 8, 5, 1, 4, 4, 1, 6, 7, 6, 6, 5, 1, 4, 1, 6, 5, 8, 2, 6, 5, 2, 7, 1, 7, 1, 7, 5, 8, 7, 4, 5, 1, 7, 2, 8, 8, 6, 1, 2, 4, 8, 6, 8, 3, 6, 8, 2, 5, 4, 4, 5, 3, 2, 3, 7, 6, 5, 6, 4},

```
{8, 6, 8, 8, 8, 3, 5, 1, 2, 7, 1, 5, 3, 7, 2, 3, 3, 4, 4, 6, 3, 6, 4, 7, 6, 7, 8, 8, 4, 5, 5, 5, 5, 4, 1, 7, 8, 2, 8, 3, 3, 6, 8, 2, 6, 6, 5, 3, 5, 1, 5, 5, 4, 5, 7, 1, 2, 8, 2, 1, 5, 2, 3, 1, 1, 6, 5, 5, 6, 3, 2, 2, 7, 8, 1, 7, 6, 1, 5, 8, 1, 2, 6, 6, 4, 1, 8, 6, 2, 5, 8, 8, 1, 8, 4, 7, 8, 4, 6, 1, 1, 4, 6, 2, 8, 2, 3, 4, 8, 3, 1, 5, 6, 8, 5, 1, 1, 3, 6, 4, 6, 7, 4, 6, 4, 6, 6, 1, 5, 6, 1, 4, 4, 4, 7, 8, 7, 3, 7, 7, 8, 1, 2, 2, 4, 2, 4, 4, 3, 3, 3, 4, 6, 6, 2, 5, 7, 4, 3, 4, 6, 8, 4, 7, 1, 4, 2, 7, 3, 1, 7, 2, 1, 4, 2, 6, 8, 2, 2, 2, 8, 4, 1, 6, 8, 7, 7, 5, 5, 4, 1, 1, 8, 8, 5, 6, 7, 2, 1},
```

```
{4, 5, 8, 1, 7, 8, 7, 7, 4, 4, 4, 2, 6, 3, 6, 6, 2, 4, 8, 8, 2, 7, 1, 5, 4, 4, 1, 6, 8, 5, 7, 2, 7, 8, 2, 8, 3, 8, 1, 4, 3, 8, 4, 8, 2, 3, 7, 3, 2, 7, 2, 4, 8, 1, 4, 3, 2, 2, 6, 8, 4, 2, 3, 7, 1, 3, 1, 5, 2, 7, 5, 4, 5, 3, 1, 1, 1, 6, 8, 8, 2, 1, 8, 8, 3, 7, 2, 7, 2, 6, 4, 7, 1, 3, 1, 1, 7, 4, 3, 1, 7, 3, 8, 4, 2, 8, 2, 7, 1, 4, 8, 8, 7, 5, 1, 3, 4, 1, 2, 1, 2, 5, 3, 4, 8, 6, 8, 1, 7, 5, 1, 6, 8, 8, 2, 5, 3, 5, 1, 3, 5, 4, 8, 6, 1, 3, 2, 1, 4, 7, 6, 2, 3, 4, 7, 8, 4, 4, 6, 6, 5, 6, 1, 6, 1, 3, 7, 5, 1, 3, 3, 7, 6, 8, 5, 5, 6, 2, 6, 2, 4, 6, 4, 3, 5, 6, 7, 6, 4, 1, 2, 4, 6, 6, 3, 5, 1, 1, 5, 4},
```

```
{2, 3, 4, 5, 4, 4, 1, 6, 1, 6, 8, 2, 5, 4, 5, 2, 3, 5, 3, 8, 1, 6, 1, 8, 3, 8, 5, 7, 7, 6, 4, 1, 7, 4, 7, 2, 2, 3, 6, 4, 2, 6, 4, 7, 5, 7, 5, 3, 2, 6, 5, 6, 1, 3, 1, 8, 6, 8, 3, 3, 3, 5, 5, 6, 2, 2, 5, 6, 7, 5, 7, 3, 2, 6, 2, 2, 8, 8, 6, 8, 1, 6, 1, 8, 2, 4, 6, 1, 1, 2, 7, 1, 8, 6, 5, 8, 3, 1, 5, 7, 6, 1, 7, 4, 5, 6, 3, 5, 7, 2, 5, 8, 5, 2, 5, 2, 7, 6, 3, 8, 6, 4, 3, 6, 3, 3, 4, 8, 5, 5, 1, 1, 2, 7, 2, 4, 4, 7, 5, 6, 4, 6, 1, 3, 6, 4, 5, 6, 6, 3, 2, 2, 2, 8, 1, 2, 4, 4, 5, 5, 7, 6, 1, 5, 1, 3, 7, 2, 8, 4, 5, 6, 6, 5, 1, 1, 6, 7, 6, 3, 4, 6, 3, 3, 6, 6, 1, 8, 7, 4, 1, 5, 3, 8, 5, 5, 6, 2, 4, 6},
```

```
{4, 7, 1, 5, 3, 5, 8, 2, 5, 4, 3, 1, 7, 8, 7, 5, 6, 1, 6, 1, 4, 2, 7, 1, 3, 7, 1, 1, 6, 1, 5, 7, 4, 8, 4, 8, 4, 8, 3, 8, 4, 7, 4, 4, 1, 5, 8, 1, 7, 5, 2, 7, 7, 2, 4, 6, 6, 5, 7, 1, 4, 1, 4, 2, 2, 6, 5, 3, 5, 7, 3, 5, 8, 8, 2, 8, 1, 2, 2, 6, 7, 3, 1, 8, 8, 4, 7, 3, 6, 1, 7, 2, 8, 3, 5, 5, 6, 8, 6, 4, 5, 3, 7, 1, 4, 1, 6, 1, 5, 3, 4, 2, 3, 1, 8, 7, 5, 3, 3, 4, 1, 1, 7, 2, 7, 1, 4, 5, 2, 5, 3, 2, 7, 3, 2, 5, 2, 1, 6, 2, 5, 6, 6, 6, 8, 8, 4, 4, 6, 3, 4, 7, 5, 1, 2, 7, 2, 5, 2, 3, 4, 3, 6, 6, 1, 8, 5, 4, 4, 4, 5, 8, 5, 8, 2, 6, 1, 6, 7, 2, 5, 3, 8, 2, 1, 4, 2, 7, 2, 6, 3, 7, 7, 4, 2, 5, 1, 4, 4, 5},
```

```
{6, 8, 5, 1, 1, 5, 2, 7, 8, 4, 1, 4, 2, 6, 2, 6, 6, 6, 8, 4, 2, 5, 6, 3, 8, 1, 5, 3, 7, 7, 8, 1, 7, 4, 6, 1, 5, 6, 6, 1, 3, 5, 5, 8, 8, 8, 2, 4, 2, 2, 4, 4, 5, 5, 7, 7, 3, 2, 5, 2, 7, 4, 2, 1, 6, 6, 7, 8, 8, 1, 2, 2, 2, 5, 6, 5, 7, 1, 1, 6, 2, 4, 1, 1, 2, 8, 8, 3, 7, 3, 3, 1, 7, 3, 3, 8, 8, 1, 4, 3, 3, 2, 2, 5, 5, 3, 5, 2, 1, 3, 6, 8, 2, 1, 4, 6, 8, 6, 2, 5, 5, 4, 5, 1, 2, 2, 5, 7, 3, 1, 3, 8, 2, 5, 8, 5, 5, 6, 7, 8, 3, 2, 7, 7, 4, 4, 6, 5, 6, 3, 7, 4, 6, 4, 8, 4, 3, 2, 7, 8, 1, 5, 5, 3, 4, 5, 7, 2, 5, 1, 7, 6, 7, 7, 2, 5, 6, 5, 3, 1, 5, 6, 8, 6, 3, 3, 5, 8, 1, 5, 8, 6, 2, 7, 4, 4, 6, 4, 8, 5});
```

```
#endif /* GAME_MATRIX_H_ */
```

```
/** game.h */
```

```
#ifndef GAMEPLAY_H_  
#define GAMEPLAY_H_
```

```
#include "glob_header.h"  
#include "display.h"
```

```
#define GAME_STATE_START 0  
#define GAME_STATE_PLAYING 1  
#define GAME_STATE_OSER 2  
#define GAME_STATE_HIGHSCORE 3  
#define GAME_STATE_DEBUG 4
```

```
void init_gameplay(void);  
void timer1_start(void);  
void start_level(uint8_t);
```

```
uint8_t temp_array[50];
```

```
#endif /* GAMEPLAY_H_ */
```

```
/** ***** gameplay.c *****
```

```
#include "gameplay.h"  
#include "game_matrix.h"  
#include "LEDs.h"
```

```
extern volatile uint8_t seed_index;  
extern volatile uint8_t seed_row;  
extern volatile uint8_t level;  
extern char player1_name[3];  
extern char player2_name[3];
```

```
void init_gameplay() {  
    timer1_start();  
    displayOn();  
  
    _delay_ms(500);  
  
    writeString("Okay...");  
    _delay_ms(1000);  
  
    _delay_ms(100);  
    writeString("Time to play... ");  
    _delay_ms(1000);  
  
    writeString(" MEMORY GAME!");  
    _delay_ms(1000);  
}
```

```
void timer1_start() {  
    TCCR1B |= (1 << CS10);  
}
```

```
void start_level(uint8_t lvl) {  
    seed_row = TCNT1 % 30; /*"slumpar" en rad i spelmatrisen  
    seed_index = TCNT1 % 150; /*"slumpar" startindex på vald rad  
    _delay_ms(2);  
    all_lights_off();  
    _delay_ms(2);
```

```

writeString("Get ready...");
_delay_ms(800);

writeString("3... ");
_delay_ms(800);

writeString("2... ");
_delay_ms(800);

writeString("1... ");
_delay_ms(1200);

for(uint8_t i = 0; i<lvl+3; i++) {
    temp_array[i] = game_matrix[seed_row][seed_index + i];
    light_on(game_matrix[seed_row][seed_index + i]);
    _delay_ms(500);
    all_lights_off();
    _delay_ms(100);
}

writeString("Your turn!");
}

```

```

//***** highscore.h *****

```

```

#ifndef HIGHSCORE_H_
#define HIGHSCORE_H_

```

```

#include "glob_header.h"

```

```

uint8_t current_name[3];
char name[3];
char nr1_points[2];
char nr2_points[2];
char nr3_points[2];

```

```

extern

```

```

void highscore_add(uint8_t);
void highscore_display(void);

```

```
#endif /* HIGHSCORE_H_ */
```

```
/** ***** highscore.c *****
```

```
#include "highscore.h"
```

```
char letters [26] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',  
'V', 'W', 'X', 'Y', 'Z'};
```

```
uint8_t name_index = 0;
```

```
uint8_t val;
```

```
void highscore_add(uint8_t lvl) {
```

```
    current_name[0] = 0;
```

```
    current_name[1] = 0;
```

```
    current_name[2] = 0;
```

```
    while (name_index < 3) {
```

```
        switch (val){
```

```
            case 2:
```

```
                switch (name_index) {
```

```
                    case 0:
```

```
                        if (current_name[0] == 0){
```

```
                            current_name[0] = 25;
```

```
                        } else {
```

```
                            current_name[0]--;
```

```
                        }
```

```
                    break;
```

```
                    case 1:
```

```
                        if (current_name[1] == 0){
```

```
                            current_name[1] = 25;
```

```
                        } else {
```

```
                            current_name[1]--;
```

```
                        }
```

```
                    break;
```

```
                    case 2:
```

```
                        if (current_name[2] == 0){
```

```
                            current_name[2] = 25;
```

```
                        } else {
```

```
                            current_name[2]--;
```

```
                        }
```

```
                    break;
```

```
                }
```

```
            break;
```

```

case 8:
    switch (name_index) {
        case 0:
            if (current_name[0] == 25){
                current_name[0] = 0;
            } else {
                current_name[0]++;
            }
            break;
        case 1:
            if (current_name[1] == 25){
                current_name[1] = 0;
            } else {
                current_name[1]++;
            }
            break;
        case 2:
            if (current_name[2] == 25) {
                current_name[2] = 0;
            } else {
                current_name[2]++;
            }
            break;
    }
    break;
case 6: if (name_index < 2) name_index++;
break;
case 4: if(name_index > 0) name_index--;
break;
case 5: name_index = 3;
break;
}
clearDisplay();
returnHome();

writeChar(letters[current_name[0]]);
writeChar(letters[current_name[1]]);
writeChar(letters[current_name[2]]);
_delay_ms(200);

clearDisplay();
returnHome();
switch(name_index){

```



```

    case 0:
        writeChar(' ');
        writeChar(letters[current_name[1]]);
        writeChar(letters[current_name[2]]);
    break;
    case 1:
        writeChar(letters[current_name[0]]);
        writeChar(' ');
        writeChar(letters[current_name[2]]);
    break;
    case 2:
        writeChar(letters[current_name[0]]);
        writeChar(letters[current_name[1]]);
        writeChar(' ');
    break;
}
_delay_ms(180);

if (name_index == 3){
    if(lvl >= eeprom_read_byte(3)){
        eeprom_update_byte(8, eeprom_read_byte(4));
        eeprom_update_byte(9, eeprom_read_byte(5));
        eeprom_update_byte(10, eeprom_read_byte(6));
        eeprom_update_byte(11, eeprom_read_byte(7));

        eeprom_update_byte(4, eeprom_read_byte(0));
        eeprom_update_byte(5, eeprom_read_byte(1));
        eeprom_update_byte(6, eeprom_read_byte(2));
        eeprom_update_byte(7, eeprom_read_byte(3));

        eeprom_update_byte(0, letters[current_name[0]]);
        eeprom_update_byte(1, letters[current_name[1]]);
        eeprom_update_byte(2, letters[current_name[2]]);
        eeprom_update_byte(3, lvl);
    } else if(lvl >= eeprom_read_byte(7)){

        eeprom_update_byte(8, eeprom_read_byte(4));
        eeprom_update_byte(9, eeprom_read_byte(5));
        eeprom_update_byte(10, eeprom_read_byte(6));
        eeprom_update_byte(11, eeprom_read_byte(7));

        eeprom_update_byte(4, letters[current_name[0]]);

```

```

        eeprom_update_byte(5, letters[current_name[1]]);
        eeprom_update_byte(6, letters[current_name[2]]);
        eeprom_update_byte(7, lvl);

    } else {

        eeprom_update_byte(8, letters[current_name[0]]);
        eeprom_update_byte(9, letters[current_name[1]]);
        eeprom_update_byte(10, letters[current_name[2]]);
        eeprom_update_byte(11, lvl);

    }

    name_index = 0;
    break;
}
}
}

```

```

void highscore_display(){

    writeString(" TOP PLAYERS:");
    _delay_ms(1000);

    itoa(eeprom_read_byte(3), nr1_points, 10);
    itoa(eeprom_read_byte(7), nr2_points, 10);
    itoa(eeprom_read_byte(11), nr3_points, 10);

    writeString("PLAYER 1: ");
    writeChar(eeprom_read_byte(0));
    writeChar(eeprom_read_byte(1));
    writeChar(eeprom_read_byte(2));
    writeChar(' ');
    writeString_no_reset(nr1_points);
    _delay_ms(1100);

    writeString("PLAYER 2: ");
    writeChar(eeprom_read_byte(4));
    writeChar(eeprom_read_byte(5));
    writeChar(eeprom_read_byte(6));
    writeChar(' ');
    writeString_no_reset(nr2_points);
    _delay_ms(1100);
}

```

```

        writeString("PLAYER 3: ");
        writeChar(eeprom_read_byte(8));
        writeChar(eeprom_read_byte(9));
        writeChar(eeprom_read_byte(10));
        writeChar(' ');
        writeString_no_reset(nr3_points);
        _delay_ms(1100);
    }

//***** LEDs.h *****

#ifndef LEDS_H_
#define LEDS_H_

#include "glob_header.h"

#define led_1 0b00000001 //PA0
#define led_2 0b00000010 //PA1
#define led_3 0b00000100 //PA2
#define led_4 0b00001000 //PA3
#define led_5 0b00010000 //PA4
#define led_6 0b00100000 //PA5
#define led_7 0b01000000 //PA6
#define led_8 0b10000000 //PA7

void all_lights_off(void);
void all_lights_on(void);
void light_on(uint8_t);
void all_lights_init(void);
void light_off(uint8_t);
void light_toggle(uint8_t);

#endif /* LEDS_H_ */

//***** LEDs.c *****

#include "LEDs.h"

void all_lights_off(){
    PORTA = 0;
    PORTC &= ~(1<<1);
}

```

```
void all_lights_on(){
    PORTA |= 0b11111111;
    PORTC |= (1<<1);
}
```

```
void all_lights_init(){
    DDRA |= 0b11111111;
    DDRC |= (1<<1);
}
```

```
void light_on(uint8_t lamp){
    PORTA = (1 << (lamp-1));
}
```

```
void light_off(uint8_t lamp){
    PORTA &= ~(1<<(lamp-1));
}
```

```
void light_toggle(uint8_t lamp){
    PORTA ^= (1<<(lamp-1));
}
```