Victor	-2567375
Marco	-2554643
Jannis	-2555689

1)

```
const int SIZE=6;
const unsigned long t_debounce=50;
const int button_left=23;
const int button_right=25;

const int LED[SIZE]={7,6,5,4,3,2};

unsigned long t=0;

int
input_1[SIZE],input_2[SIZE],output[SIZE],ONE[SIZE]={0,0,0,0,0,1},ZERO[SIZE]={0,0,0,0,0,0};
unsigned int state=0;

int is_left_button_pressed=LOW;
int is_right_button_pressed=LOW;

int add(int* output,int* input_1,int* input_2,int size);
void print_array(int* arr,int size);
unsigned long deltat();

void setup() {
  // put your setup code here, to run once:
  for(int i=0;i<SIZE;i++){
    input_1[i]=0;
    input_2[i]=0;
    output[i]=0;
    pinMode(LED[i], OUTPUT);
    digitalWrite(LED[i], LOW);
  }
  pinMode(button_left,INPUT);
  pinMode(button_right,INPUT);
  Serial.begin(9600);
  t=millis();
}

void loop() {
  // put your main code here, to run repeatedly:
  //Serial.println(state);
  switch (state) {
      case 0:
        is_left_button_pressed=digitalRead(button_left);
        is_right_button_pressed=digitalRead(button_right);
        if(is_left_button_pressed==HIGH){
          t=millis();
          state=1;
          int input_1_copy[SIZE];
          for(int i=0;i<SIZE;i++){
```

```
      input_1_copy[i]=input_1[i];
     }
     add(input_1,input_1_copy,ONE,SIZE);
   }else if(is_right_button_pressed==HIGH){
     t=millis();
     state=3;
   }
   break;
 case 1:
   Serial.println("INPUT1");
   print_array(input_1,SIZE);
   if(deltat()>=t_debounce ){
     state=2;
   }
   break;
 case 2:
   is_left_button_pressed=digitalRead(button_left);
   if(is_left_button_pressed==LOW){
     state=0;
   }
   break;

  case 3:
   if(deltat()>=t_debounce){
     state=4;
     print_array(ZERO,SIZE);
   }
   break;
  case 4:
   is_right_button_pressed=digitalRead(button_right);
   if(is_right_button_pressed==LOW){
     state=5;
   }
   break;

  case 5:
   is_left_button_pressed=digitalRead(button_left);
   is_right_button_pressed=digitalRead(button_right);
   if(is_left_button_pressed==HIGH){
     t=millis();
     state=6;
     int input_2_copy[SIZE];
     for(int i=0;i<SIZE;i++){
       input_2_copy[i]=input_2[i];
     }
     add(input_2,input_2_copy,ONE,SIZE);
   }else if(is_right_button_pressed==HIGH){
     t=millis();
     state=8;
   }
   break;
 case 6:
```

```
  Serial.println("INPUT2");
  print_array(input_2,SIZE);
  if(deltat()>=t_debounce){
    state=7;

  }
  break;
 case 7:
  is_left_button_pressed=digitalRead(button_left);
  if(is_left_button_pressed==LOW){
    state=5;
  }
  break;

 case 8:
  if(deltat()>=t_debounce){
    state=9;
  }
  break;
 case 9:
  is_right_button_pressed=digitalRead(button_right);
  if(is_right_button_pressed==LOW){
    state=10;
    add(output,input_1,input_2,SIZE);
    Serial.println("OUTPUT");
    print_array(output,SIZE);
  }
  break;

  case 10:
    is_right_button_pressed=digitalRead(button_right);
    if(is_right_button_pressed==HIGH){
      state=11;
      t=millis();
    }
    break;
  case 11:
    for(int i=0;i<SIZE;i++){
      input_1[i]=0;
      input_2[i]=0;
      output[i]=0;
    }
    if(deltat()>2*t_debounce){
      state=12;
    }
    break;
  case 12:
    is_right_button_pressed=digitalRead(button_right);
    if(is_right_button_pressed==LOW){
      state=0;
      print_array(input_1,SIZE);
    }
```

```
        break;
      default:
        state=0;
        Serial.println("RESETING FSM");
  }
}

int add(int* output,int* input_1,int* input_2,int size){
  int aux=0;
  for(int i=size-1;i>=0;i--){
    output[i]=(input_1[i]+input_2[i]+aux)%2;//(input_1[i] XOR input_2[i]) XOR aux;
    aux=(input_1[i] && input_2[i]) || (aux and(input_1[i]+input_2[i]%2));
  }
  return aux;
}

void print_array(int* arr,int size){
  for(int i=0;i<size;i++){
    Serial.print(arr[i]);
    Serial.print(", ");
    digitalWrite(LED[i], arr[i]);
  }
  Serial.println("");
}

unsigned long deltat(){
  return millis()-t;
}
```

Taster nicht richtig entprellt
=> manchmal Tasterdruck mehrmals erkannt

6,5/7

1.6)

```
const int SIZE=6;
const unsigned long t_debounce=50;
const unsigned long t_fast_counter=500;
const int button_left=23;
const int button_right=25;

const int LED[SIZE]={7,6,5,4,3,2};

unsigned long t=0;
int fast_counter=0;

int
input_1[SIZE],input_2[SIZE],output[SIZE],ONE[SIZE]={0,0,0,0,0,1},ZERO[SIZE]={0,0,0,0,0,0},
TEN[SIZE]={0,0,1,0,1,0};
unsigned int state=0;

int is_left_button_pressed=LOW;
int is_right_button_pressed=LOW;

int add(int* output,int* input_1,int* input_2,int size);
void print_array(int* arr,int size);
unsigned long deltat();

void setup() {
  // put your setup code here, to run once:
  for(int i=0;i<SIZE;i++){
    input_1[i]=0;
    input_2[i]=0;
    output[i]=0;
    pinMode(LED[i], OUTPUT);
    digitalWrite(LED[i], LOW);
  }
  pinMode(button_left,INPUT);
  pinMode(button_right,INPUT);
  Serial.begin(9600);
  t=millis();
}

void loop() {
  // put your main code here, to run repeatedly:
  //Serial.println(state);
  switch (state) {
      case 0:
        is_left_button_pressed=digitalRead(button_left);
        is_right_button_pressed=digitalRead(button_right);
        if(is_left_button_pressed==HIGH){
          t=millis();
```

```
    state=1;
   }else if(is_right_button_pressed==HIGH){
    t=millis();
    state=3;
   }
   break;
  case 1:
   if(deltat()>=t_debounce ){
    state=2;
   }
   break;
  case 2:
   is_left_button_pressed=digitalRead(button_left);
   if(is_left_button_pressed==LOW){
    state=0;
    int input_1_copy[SIZE];
    for(int i=0;i<SIZE;i++){
     input_1_copy[i]=input_1[i];
    }
    add(input_1,input_1_copy,ONE,SIZE);
    Serial.println("INPUT1");
    print_array(input_1,SIZE);
   }else if(deltat()>=t_fast_counter){
    state=13;
    t=millis();
    fast_counter=0;
   }
   break;
  case 13:
   if(deltat()>=10){
    state=14;
    fast_counter++;
   }else if(digitalRead(button_left)==LOW){
    state=0;
    print_array(input_1,SIZE);

   }
   break;
  case 14:
    if(fast_counter>=10){
    int input_1_copy[SIZE];
    for(int i=0;i<SIZE;i++){
        input_1_copy[i]=input_1[i];
    }
    add(input_1,input_1_copy,TEN,SIZE);
    fast_counter=0;
   }

   Serial.println("INPUT1");
   print_array(input_1,SIZE);
   if(digitalRead(button_left)==LOW){
    state=0;
```

```
  }else{
    state=13;
  }
  break;
 case 3:
  if(deltat()>=t_debounce){
    state=4;
    print_array(ZERO,SIZE);
  }
  break;
 case 4:
  is_right_button_pressed=digitalRead(button_right);
  if(is_right_button_pressed==LOW){
    state=5;
  }
  break;

 case 5:
  is_left_button_pressed=digitalRead(button_left);
  is_right_button_pressed=digitalRead(button_right);
  if(is_left_button_pressed==HIGH){
    t=millis();
    state=6;
  }else if(is_right_button_pressed==HIGH){
    t=millis();
    state=8;
  }
  break;
case 6:
  if(deltat()>=t_debounce){
    state=7;
  }
  break;
case 7:
  is_left_button_pressed=digitalRead(button_left);
  if(is_left_button_pressed==LOW){
    state=5;
    int input_2_copy[SIZE];
    for(int i=0;i<SIZE;i++){
      input_2_copy[i]=input_2[i];
    }
    add(input_2,input_2_copy,ONE,SIZE);
    Serial.println("INPUT2");
    print_array(input_2,SIZE);
  }else if(deltat()>=t_fast_counter){
    state=15;
    t=millis();
    fast_counter=0;
  }
  break;
 case 15:
  if(deltat()>=10){
```

```
   state=16;
   fast_counter++;
  }else if(digitalRead(button_left)==LOW){
   state=5;
   print_array(input_2,SIZE);

  }
  break;
 case 16:
   if(fast_counter>=10){
    int input_2_copy[SIZE];
    for(int i=0;i<SIZE;i++){
        input_2_copy[i]=input_2[i];
    }
    add(input_2,input_2_copy,TEN,SIZE);
    fast_counter=0;
   }

  Serial.println("INPUT2");
  print_array(input_2,SIZE);
  if(digitalRead(button_left)==LOW){
   state=5;
  }else{
   state=15;
  }
  break;

 case 8:
  if(deltat()>=t_debounce){
   state=9;
  }
  break;
 case 9:
  is_right_button_pressed=digitalRead(button_right);
  if(is_right_button_pressed==LOW){
   state=10;
   add(output,input_1,input_2,SIZE);
   Serial.println("OUTPUT");
   print_array(output,SIZE);
  }
  break;

  case 10:
   is_right_button_pressed=digitalRead(button_right);
   if(is_right_button_pressed==HIGH){
     state=11;
     t=millis();
   }
   break;
  case 11:
   for(int i=0;i<SIZE;i++){
     input_1[i]=0;
```

```
          input_2[i]=0;
          output[i]=0;
         }
         if(deltat()>2*t_debounce){
          state=12;
         }
         break;
       case 12:
         is_right_button_pressed=digitalRead(button_right);
         if(is_right_button_pressed==LOW){
          state=0;
          print_array(input_1,SIZE);
         }
         break;
      default:
       state=0;
       Serial.println("RESETING FSM");
  }
}

int add(int* output,int* input_1,int* input_2,int size){
  int aux=0;
  for(int i=size-1;i>=0;i--){
    output[i]=(input_1[i]+input_2[i]+aux)%2;//(input_1[i] XOR input_2[i]) XOR aux;
    aux=(input_1[i] && input_2[i]) || (aux and(input_1[i]+input_2[i]%2));
  }
  return aux;
}

void print_array(int* arr,int size){
  for(int i=0;i<size;i++){
    Serial.print(arr[i]);
    Serial.print(", ");
    digitalWrite(LED[i], arr[i]);
  }
  Serial.println("");
}

unsigned long deltat(){
  return millis()-t;
}
```

Display aktualisierung funktioniert, zählt aber falsch hoch

+ 1/2 Bonus

2.0)
Listing 1:
12 wird von "Serial.print()" ausgegeben, weil in C der innerste Scope immer zuerst betrachtet wird. In dieses Beispiel steht "int x=12" im innersten Scope. ✓

Listing 2:
12 wird von Serial.print() ausgegeben, da "Serial.print()" erst aufgerufen wird, nachdem die globale Variable "x" mit 12 aktualisiert wurde. ✓

Listing 3:
7 wird von "Serial.print()" ausgegeben, weil in C der innerste Scope immer zuerst betrachtet wird. In diesem Beispiel ist der Scope mit der Anweisung "int x=12" schon geschlossen. ✓

Listing 4:
7 wird von "Serial.print()" ausgegeben, weil in C der innerste Scope immer zuerst betrachtet wird. In diesem Beispiel ist der Scope mit der Anweisung "int x=newX" schon durch die Funktion geschlossen. ✓

Listing 5:
12 wird von Serial.print() ausgegeben, da "Serial.print()" erst aufgerufen wird, nachdem die globale Variable "x" mit 12 aktualisiert wurde. ✓

5/5

12,5/12