

## ASSIGNMENT

### วงจรเกมส์ทายเลข

รายวิชา 242-309 MICROCONTROLLER & INTERFACING

ภาคการศึกษา 2/2560

โดย

นายไฟศอล หะยี่ รหัสนักศึกษา 5735512151

นายมาซรัน ยะมิง รหัสนักศึกษา 5835512083

เสนอ

อาจารย์พัชรี เทพนมิตร

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

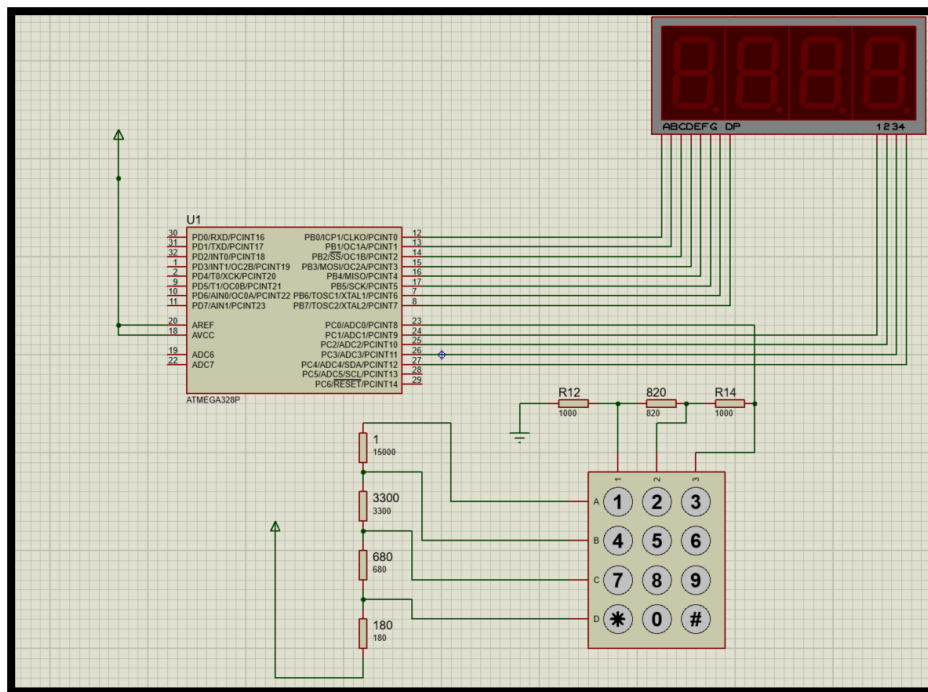
### 1.ชื่อหัวข้อ : วงจรเกมส่ายเลข

### 2.สมาชิกในกลุ่ม :

นายไพศาล หะยี่ รหัสนักศึกษา 57355121151

นายมาซัน ยะมิง รหัสนักศึกษา 58355121083

### 3.โครงสร้างระบบ :



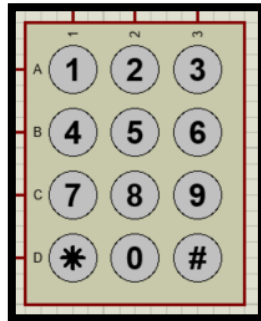
รูปที่ 1 ตัวอย่างวงจร

### อุปกรณ์ที่ใช้

1. Atmega328p 1 ตัว
2. keypad 3x4 1 ตัว
3. ตัวต้านทาน R 7 ตัว
4. Segment multiplexing 4 digit 1 ตัว สีแดง

## INPUT

- Keypad 3x4



## OUTPUT

- 7 segment multiplexing 4 digit



## PORT ที่ใช้งานมีดังนี้

PORT B	
ขาที่	ต่อกับ
PB0	ขา a ของ 7-Segment
PB1	ขา b ของ 7-Segment
PB2	ขา c ของ 7-Segment
PB3	ขา d ของ 7-Segment
PB4	ขา e ของ 7-Segment
PB5	ขา f ของ 7-Segment
PB6	ขา g ของ 7-Segment
PB7	ขา dp ของ 7-Segment

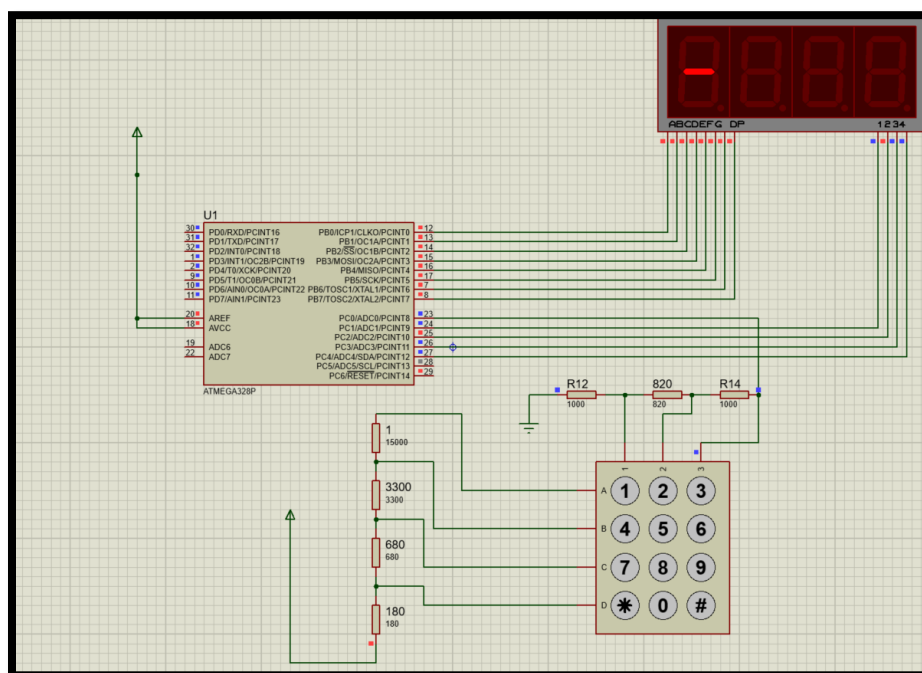
PORT C	
ขาที่	ต่อกับ
PC0	Keypad 3x4
PC1	หลักที่ 1 ของ 7-Segment
PC2	หลักที่ 2 ของ 7-Segment
PC3	หลักที่ 3 ของ 7-Segment
PC4	หลักที่ 4 ของ 7-Segment

### โหมดในการทำงานและวิธีการใช้งาน

1. ป้อนจำนวนตัวเลขไม่เกิน 3 หลัก โดยรับค่าจาก Keypad
2. ปุ่ม # คือ เพื่อทำการยืนยันตัวเลขที่ต้องการทายไว้

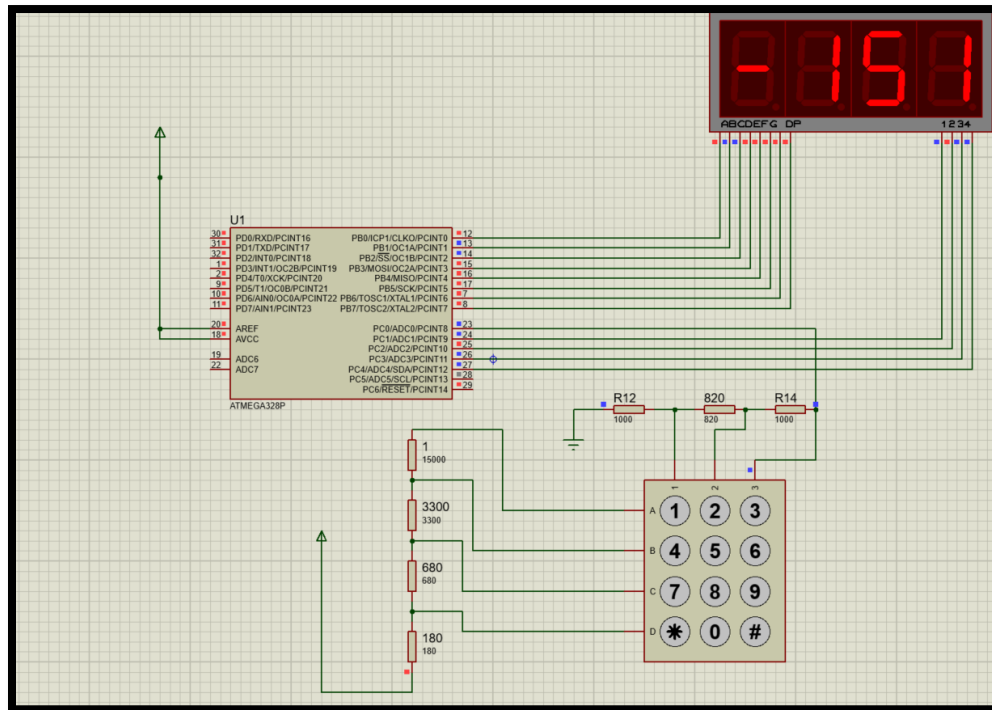
### 4.หลักการทำงาน

1. เมื่อวงจรเริ่มทำงาน วงจรจะทำการสุ่มเลขเพื่อกำหนดเป็นคำตอบที่ถูกต้องสำหรับการเล่นเกมสทายเลขขึ้นมาอัตโนมัติ และแสดงค่าเริ่มต้นออกมาทาง 7 segment multiplexing เป็น - ดังรูป



รูปที่ 2 เมื่อวงจรทำงาน

2. เมื่อผู้ใช้ป้อนค่าลงบน keypad เพื่อกำหนดเลขในการทาย



รูปที่ 3 ตัวอย่างการป้อนค่า

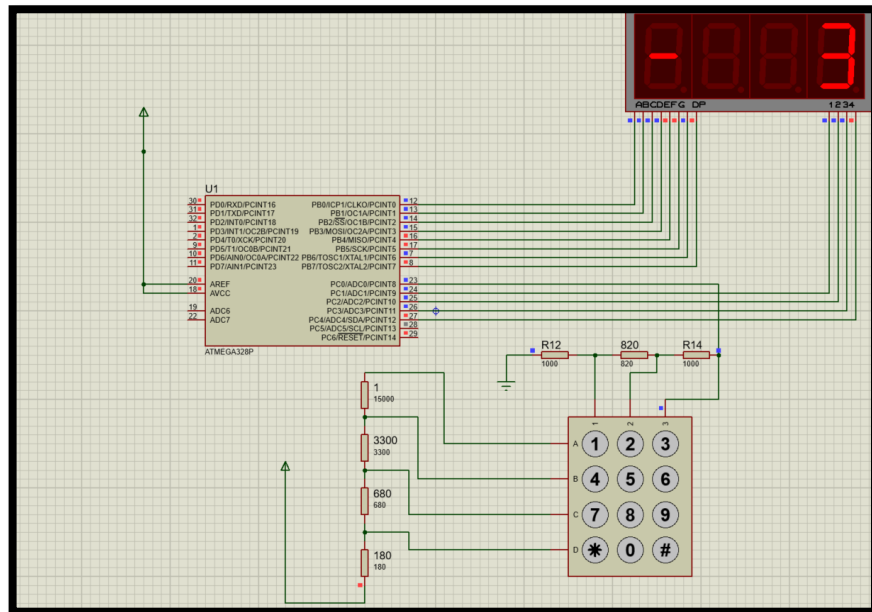
3. ผู้ใช้กดปุ่ม # เพื่อทำการยืนยันคำตอบที่ต้องการทายเพื่อเล่นเกมส์

4. หลังจากผู้ใช้กดปุ่ม # เพื่อทำการทายคำตอบที่ถูกต้องโปรแกรมจะทำการแสดงสถานะของคำตอบที่ทายขั้นต้นนั้นใกล้เคียงกับคำตอบที่ถูกต้องมากน้อยแค่ไหน เพื่อให้การสุ่มครั้งต่อไปใกล้เคียงกับคำตอบที่ถูกต้องมากขึ้น โดยจะทำการแสดงสถานะของคำตอบที่ผู้ใช้ทาย 3 แบบดังนี้

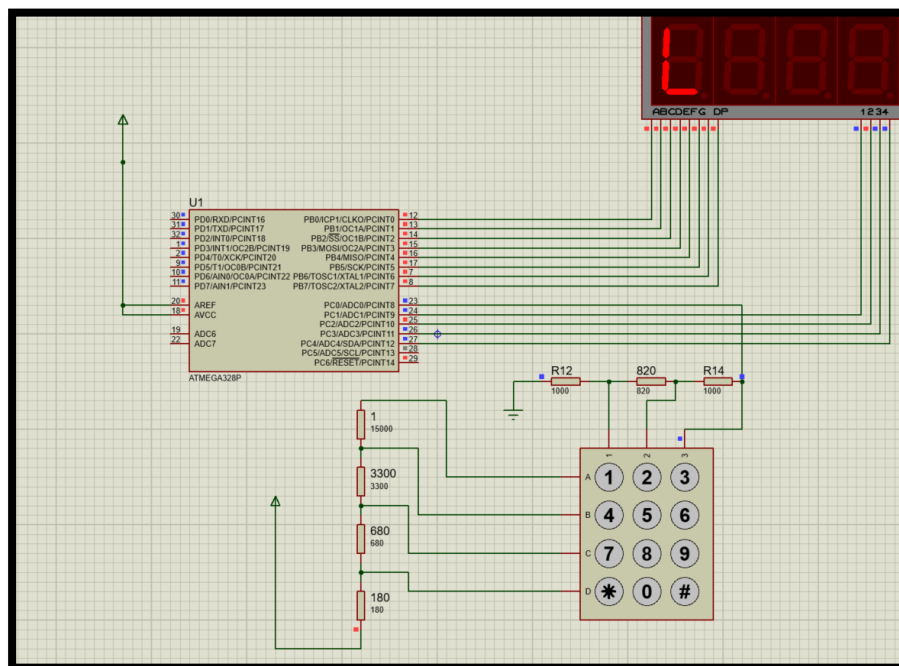
4.1 แสดงตัวอักษร L หมายถึง เลขที่ผู้ใช้ป้อนเพื่อทายคำตอบอยู่ในช่วงต่ำกว่าหรือ Low กว่าคำตอบที่ถูกต้อง

4.2 แสดงตัวอักษร H หมายถึง เลขที่ผู้ใช้ป้อนเพื่อทายคำตอบอยู่ในช่วงสูงกว่าหรือ Hight กว่าคำตอบที่ถูกต้อง

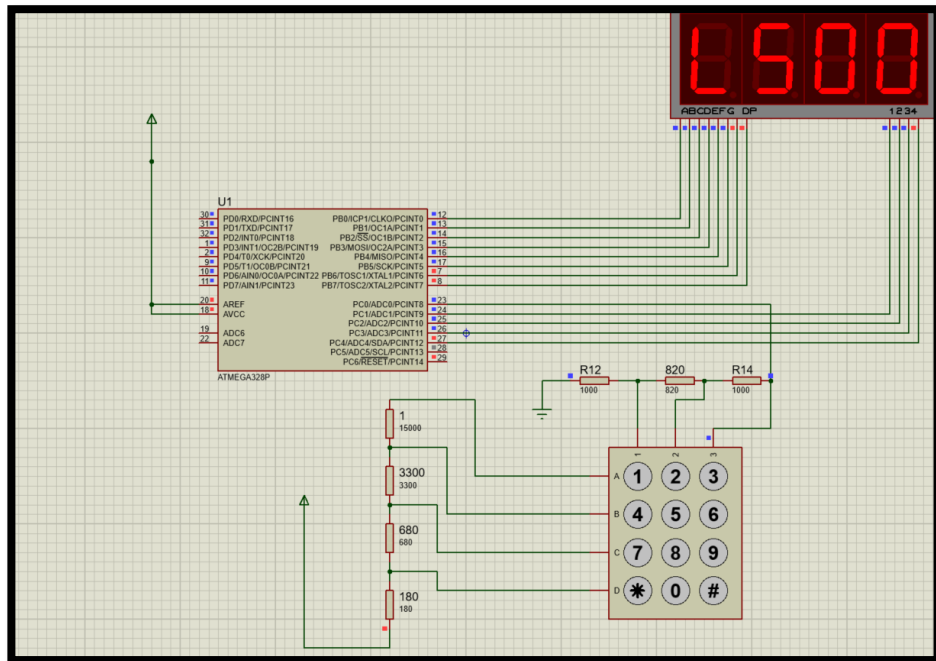
4.2 แสดงตัวอักษร F หมายถึง Finish หรือ เลขที่ผู้ใช้ป้อนเพื่อทายคำตอบนั้นถูกต้อง



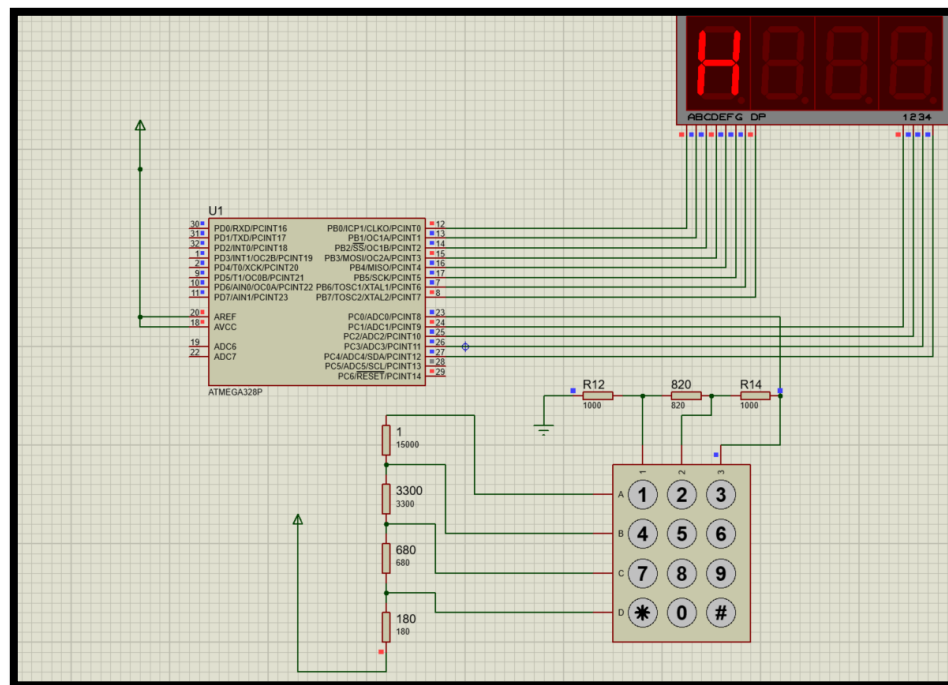
รูปที่ 4.1 เมื่อผู้ใช้ป้อนเลข 3



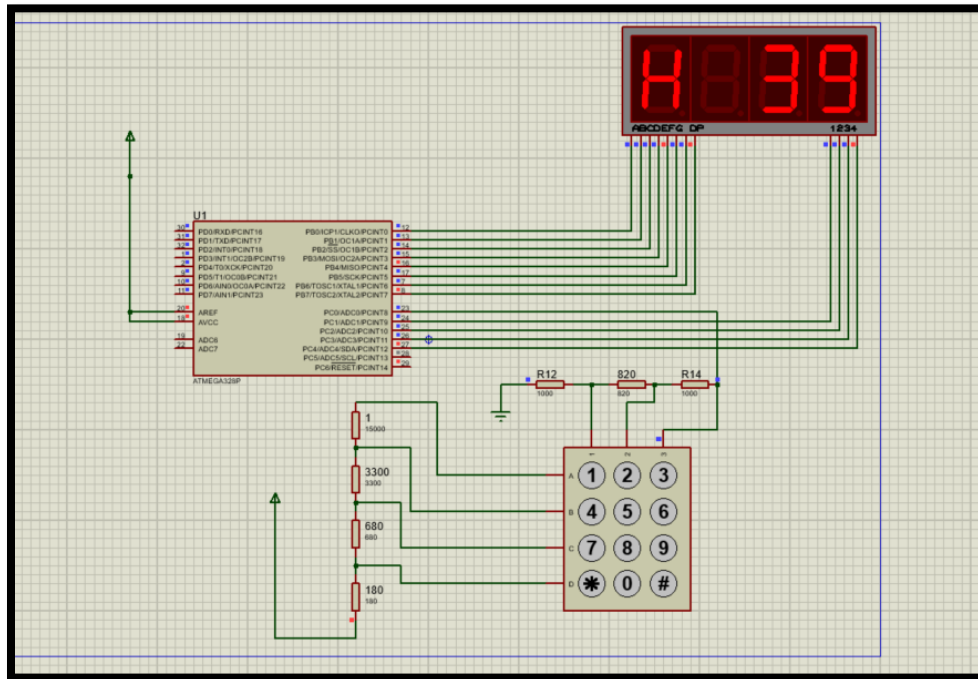
รูปที่ 4.1.1 ตัวอย่างเมื่อเลขต่ำกว่าคำตอบที่ถูกต้อง



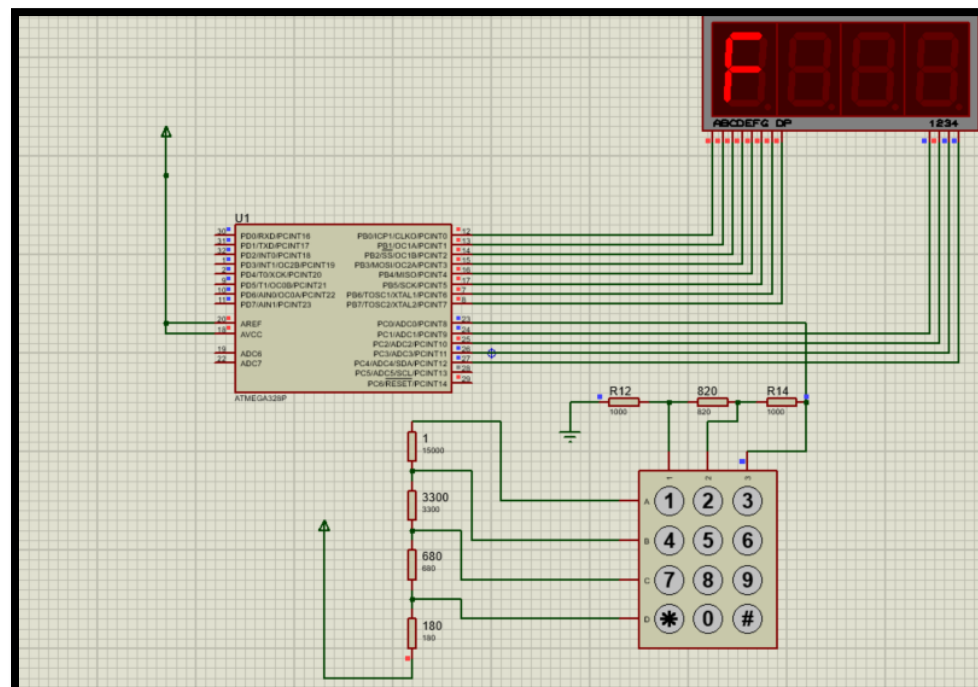
รูปที่ 4.2 เมื่อผู้ใช้ป้อนเลข 500



รูปที่ 4.2.1 ตัวอย่างเมื่อเลขสูงกว่าคำตอบที่ถูกต้อง



รูปที่ 4.3 เมื่อผู้ใช้ป้อนเลข 39



รูปที่ 4.3.1 ตัวอย่างการแสดงผลเมื่อผู้ใช้ป้อนคำตอบถูกต้อง



## 5.สรุปเทคนิคที่ใช้

- ใช้ความรู้เรื่อง Counter/Timer Timer1 และมีการใช้งาน Timer1 Overflow

TCNT1 = 65528; ใช้งาน Timer1 ให้เริ่มจาก 65528 - 65536

TCCR1A = 0x00; ใช้งานโหมด Normal mode

TCCR1B = 0x05; กำหนดให้ Prescale factor = 1024

TIMSK1 = 0x01; Timer/Counter1 Overflow interrupt Enable

- ใช้ความรู้เรื่อง Analog to Digital Converter เพื่อใช้ในการแปลงค่าแรงดันจาร์คีย์แพด โดยแปลงจากอนาลอกเป็นดิจิตอล

ADMUX = 0x20; ใช้แรงดันอ้างอิงจาก aref

ADCSRA = 0b10101111; ADC Enable , ADC Auto Trigger Enable , ADC Interrupt Enable

ADCSRB = 0b00000110; Timer/Counter1 Overflow

## 6.Source code

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <stdio.h>
#include <util/delay.h>
#include <stdlib.h>

unsigned char TB7SEG[] = {
    0b00111111, 0b00000110, 0b01011011, 0b01001111,
    0b01100110, 0b01101101, 0b01111101, 0b00000111,
    0b01111111, 0b01101111, 0b01110111, 0b01111100,
    0b00111001, 0b01011110, 0b01111001, 0b01110001,
    0b01010010, 0b11101110, 0b0111000, 0b1000000, 0b1110001, 0b0000000};

unsigned int accumulator=0;
unsigned char quotient;
unsigned char divisor=0;
unsigned char DECODE = 0, sumstate1=0, sumstate2=0, sumstate3=0;
unsigned char state=0, num=0, sum=0, s=19, M=0, check=1, count=0;
unsigned char value=21, value1=21, value2=21, x=21, x1=21, x2=21;
void display4digits(unsigned char d ,unsigned char d1,unsigned char d2,unsigned char d3){

    //PORTC = 0b00000000;

    PORTC = 0b00000010;
    PORTB = ~TB7SEG[d];
    _delay_ms(15);
```

```

    PORTC = 0b00000100;
    PORTB = ~TB7SEG[d1];
    _delay_ms(15);

    PORTC = 0b00001000;
    PORTB = ~TB7SEG[d2];
    _delay_ms(15);

    PORTC = 0b00010000;
    PORTB = ~TB7SEG[d3];
    _delay_ms(15);
}

```

```

int main()
{
    DDRD = 0xff;
    DDRB = 0xff;
    DDRC = 0b00011110;

    PORTB = 0xff;

    ADMUX = 0x20;
    ADCSRA = 0b10101111;
    ADCSRB = 0b00000110;

    TCNT1 = 65528;
    TCCR1A = 0x00;
    TCCR1B = 0x05;
    TIMSK1 = 0x01;
    sei();
}

```

```
M = rand()%1000;
```

```
while (1){
```

```
    if(state==0 && num==1 ){
```

```
        value=21;value1=21;
```

```
        value2 = DECODE;x2=DECODE;
```

```
        state=1;check=1;count=1;
```

```
    }
```

```
    else if(state==1 && num==2 ){
```

```
        value1 = value2;value2=DECODE;x1=x2;x2=DECODE;
```

```
        state=2;check=1;count=2;
```

```
    }
```

```
    else if(state==2 && num ==3){
```

```
        value= value1;value1=value2;value2 = DECODE;x=x1;x1=x2;x2=DECODE;
```

```
        state=0;num=0;check=1;count=3;
```

```
    }
```

```
    if(DECODE == 11){
```

```
        // x=21;x1=21;x2=21;value=21;value1=21;value2=21;
```

```
        if(count > 0){
```

```
            if(count == 1)sum=value2;
```

```
            else if (count == 2)sum=((value1*10)+value2);
```

```
            else if(count ==3 )sum=((value*100)+(value1*10)+value2);
```

```
        }
```

```
    if(check == 1){
```

```
        if(sum == M){s=20;
```

```

        M = rand()%1000;
        x=21;x1=21;x2=21;
        check=0;
    }
    else if(sum > M){s=17;
        check=0;x=21;x1=21;x2=21;
    }
    else if(sum < M){s=18;
        check=0;x=21;x1=21;x2=21;
    }
    }
    state=0;num=0;
}
    display4digits(s,x,x1,x2);
}
return 0;
}
ISR(TIMER1_OVF_vect)
{
    TCNT1 = 65528; //à»ÇÒÂˆˆˆÇèÒá»ÅšàÊÃÇˆ
}
ISR(ADC_vect)
{
    PORTD = ~PORTD;
    if(ADCH>9) //ifàÃÒjˆ
    {
        accumulator += ADCH;
        divisor++;
    }
    else

```

```

{
    accumulator = 0;
    divisor = 0;
}
if(divisor == 10)
{
    quotient = accumulator/divisor;
    if(quotient>9 && quotient<16)      DECODE = 1;
    else if(quotient>18 && quotient<26)  DECODE = 2;
    else if(quotient>29 && quotient<37)  DECODE = 3;
    else if(quotient>46 && quotient<53)  DECODE = 4;
    else if(quotient>74 && quotient<81)  DECODE = 5;
    else if(quotient>99 && quotient<107) DECODE = 6;
    else if(quotient>133 && quotient<141) DECODE = 7;
    else if(quotient>169 && quotient<177) DECODE = 8;
    else if(quotient>192 && quotient<199) DECODE = 9;
    else if(quotient>212 && quotient<220) DECODE = 10;
    else if(quotient>228 && quotient<236) DECODE = 0;
    else if(quotient>236 && quotient<244) DECODE = 11;

    num+=1;
    if(DECODE == 11){
        num-=1;
    }
}
}

```