

Prep

Task 1

① Write

```
.def button = r17
```

```
.def switch = r18
```

```
.def leds = r19
```

```
clr temp ; Already defined as r16
```

```
out DDRA, temp ; Toggle Switch input
```

```
out DDRC, temp ; Push Button input
```

```
ser temp
```

```
out DDRB, temp ; LEDs output
```

```
clr leds
```

```
ButtonWaitPress:
```

```
in button, PINC
```

```
andi button, 0x01 ; only look at LSB
```

```
cpi button, 0x00 ; is button down?
```

```
breq ButtonWaitRel
```

```
rjmp ButtonWaitPress
```

```
Accumulate:
```

```
in switch, PINA
```

```
add leds, switch
```

```
out PORTB, leds
```

```
rjmp ButtonWaitPress
```

```
ButtonWaitRel:
```

```
in button, PINC
```

```
andi button, 0x01 ; only look at LSB?
```

```
cpi button, 0x01 ; is button Released?
```

```
breq Accumulate
```

```
rjmp ButtonWaitRel
```

Prac 6

Prep Task 1 Cont.

② Simulate — As predicted

Task 2 — Combination Lock

① Write

```
.def tens = r17 ; use low nibble
.def units = r18 ; use low nibble
.def input = r19 ; low nibble = SW, LSB (high nibble) = PB
.def status = r20 ; boolean 1 or 0
.equ bmask = 0x10 ; mask for PB
.equ nmask = 0x0F ; mask for low nibbles
rjmp Initialize
```

ButtonWait Press:

```
in input, PINC
andi input, bmask
cpi input, 0x00
breq ButtonWait Rel
rjmp ButtonWait Press
```

ButtonWait Rel:

```
in input, PINC
andi input, bmask
cpi input, 0x01
breq Set Hex
rjmp ButtonWait Rel
```

Initialize:

```
clr temp
out DDRK, temp
ser temp
out DDRA, temp
out DDRB, temp
out DDRD, temp
```

Prac 6

```
clr tens  
clr units  
clr status  
rjmp DisplayHex
```

DisplayHex:

```
out PORTA, tens  
out PORTB, units  
rjmp SetStatus
```

SetStatus:

```
cpi tens, 0x08 ; given tens digit  
brne Close  
cpi units, 0x03 ; given units digit  
breq Open  
rjmp Close
```

Close:

```
ldi status, 0x02 ; pin 1 for Closed  
out PORTD, status  
rjmp ButtonWaitPress
```

Open:

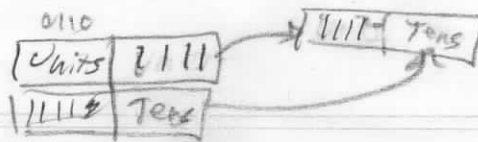
```
ldi status, 0x01 ; pin 0 for open  
out PORTD, status  
rjmp ButtonWaitPress
```

SetHex:

```
mov units, tens  
in tens, PINC  
andi tens, nmask ; only get from switches  
rjmp DisplayHex
```

Prac 6

Procedure



1 Fine mm
1 wk bk mm
1 Tutor mm

① A) Simulate my Accumulator — Success

B) Simulate Partner's Accumulator — N/A

C) Notes — Our program organization is the same.

② Download in Accumulator to the board (Wire, test, document)

Test succeeded as planned. Before starting we should always be sure to wire up mains power.

③ A) Simulate my Combo Lock — Success

B) Simulate partner's Combo Lock — N/A

C) Notes — Remember to test several cases.

④ Download a Combo Lock to the board (Wire, test, document)

Tested with various combinations. Starts with 00, led state shows lock is closed. 83 opens lock as simulation and design description.

Test cases:

Accumulator was tested as per documentation.

After reset, it held 0. Input 64 yields $0+64=64$
Input 31 yields $64+31=95$ and input 21
yields $21+95=116$.

Combination Lock was tested through trial of correct combination. At start the combination is 00 which is wrong so the LED indicated "closed." Entering the first digit yielded 30 which is still wrong with "closed" LED. Finally Entering the next digit yielded 83 which is correct and indicated as such with the "open" LED.

Prag 6

Tutor Task

Prep

① DDRC \rightarrow 00000110 so that
Pin 0 \rightarrow PB input
Pin 1 \rightarrow Open LED output 0x02
Pin 2 \rightarrow Close LED output 0x04
Pins 6-3 \rightarrow SW inputs 01111000

② DDRA \rightarrow 11111111 so that
Pins 3-0 \rightarrow Tens Digit Hex Display output
Pins 7-4 \rightarrow Units Digit Hex Display output

Procedure:

- ① Create new project and copy code from Task 2
- ② Alter code to reflect new specification
- ③ Simulation led to the rectification of several bugs
(so step ② was repeated until simulation was successful.
- ④ After successful simulation we revised the AVR board as per the specification.
- ⑤ Further testing required the input of various combinations and checking the Hex Display and LEDs for correct results.
 - A) After programming the device, it is set at 00 and the lock is closed
 - B) Entering digit 3 on the switches resulted in Hex 30 after the button press. And LED closed
 - C) Entering 8 on the switches resulted in Hex 83 after the button press. And LED opened.
- ⑥ This is the result exactly specified.

Notes: Simulation is a very useful tool as it gives a look inside the chip. Without it, we would be blind.