Program Code

Provide your program code here for each part of the work task (copy-paste your code).

1. **Sum-of-Products (SoP)**

```
        ORG     PROG

    ; Insert your code following the label "Entry"
    Entry:                  ; KEEP THIS LABEL!!
        BSR     INIT
    MAIN:
        LDAA    PTH
        COMA
        CMPA    #0
        BEQ     SET_SOP
        CMPA    #1
        BEQ     SET_SOP
        CMPA    #2
        BEQ     RESET_SOP
        CMPA    #3
        BEQ     RESET_SOP
        CMPA    #4
        BEQ     RESET_SOP
        CMPA    #5
        BEQ     SET_SOP
        CMPA    #6
        BEQ     RESET_SOP
        CMPA    #7
        BEQ     RESET_SOP
        CMPA    #8
        BEQ     SET_SOP
        CMPA    #9
        BEQ     SET_SOP
        CMPA    #10
        BEQ     RESET_SOP
        CMPA    #11
        BEQ     RESET_SOP
        CMPA    #12
        BEQ     RESET_SOP
        CMPA    #13
        BEQ     RESET_SOP
        CMPA    #14
        BEQ     SET_SOP
        CMPA    #15
        BEQ     SET_SOP
    RESET_SOP:
        LDAB    #0
        BRA     LIGHT_SOP
```

```
SET_SOP:
        LDAB    #1
        BRA     LIGHT_SOP
LIGHT_SOP:
        STAB    PORTB
        BRA     MAIN

        ; Branch to end of program
        BRA     FINISH
```

2. Product-of-Sums (PoS)

```
        ORG     PROG

; Insert your code following the label "Entry"
Entry:                  ; KEEP THIS LABEL!!
        BSR     INIT
MAIN:
        LDAA    PTH
        COMA
        CMPA    #0
        BEQ     RESET_POS
        CMPA    #1
        BEQ     SET_POS
        CMPA    #2
        BEQ     RESET_POS
        CMPA    #3
        BEQ     SET_POS
        CMPA    #4
        BEQ     RESET_POS
        CMPA    #5
        BEQ     RESET_POS
        CMPA    #6
        BEQ     RESET_POS
        CMPA    #7
        BEQ     RESET_POS
        CMPA    #8
        BEQ     RESET_POS
        CMPA    #9
        BEQ     SET_POS
        CMPA    #10
        BEQ     RESET_POS
        CMPA    #11
        BEQ     SET_POS
        CMPA    #12
        BEQ     SET_POS
        CMPA    #13
        BEQ     SET_POS
```

```
        CMPA    #14
        BEQ     SET_POS
        CMPA    #15
        BEQ     RESET_POS
RESET_POS:
        LDAB    #0
        BRA     LIGHT_POS
SET_POS:
        LDAB    #1
        BRA     LIGHT_POS
LIGHT_POS:
        STAB    PORTB
        BRA     MAIN

        ; Branch to end of program
        BRA     FINISH
```

Demo

You must demo the function code to the instructor during the activity time.

Questions

1. **Why do the pushbuttons use negative logic? How can you make the pushbutton inputs positive logic in assembly?**

   The pushbutton use negative logic because when there is high voltage from the power supply, there needs to be low voltage at the pushbutton in order to allow the current to flow and light up the leds. This is why we use the COMA instruction to invert the signal it receives to make it positive logic and function as needed.

2. **Why are the LEDs interfaced to the microcontroller through a logic device (e.g., inverter or buffer)?**

   The voltage differences across the components need intermediary devices to regulate the voltages supplied to each component. If this wasn't the case you may get logic discrepancies that would not be an ideal situation.