Lab 3

Basic

▼ Answer

```
List p = 18f4520
#include<p18f4520.inc>
CONFIG OSC = INTIO67
CONFIG WDT = OFF
org 0x00

    CLRF TRISA
    MOVLW b'01011111' ;
    MOVWF TRISA

    BCF STATUS, C
    RLCF TRISA, F

    BCF STATUS, C
    BTFSC TRISA, 7
    BSF STATUS, C
    RRCF TRISA, F

Over:
    end
```

Advanced

▼ Answer

```
List p = 18f4520
#include<p18f4520.inc>
CONFIG OSC = INTIO67
CONFIG WDT = OFF
org 0x00

    CLRF 0x020
    CLRF 0x021
    CLRF 0x022
    CLRF 0x023

    MOVLW 0x12
    MOVWF 0x000 ; a1

    MOVLW 0xCB
    MOVWF 0x001 ; a0

    MOVLW 0x09
    MOVWF 0x010 ; b1

    MOVLW 0x35
    MOVWF 0x011 ; b0

Multiply:
    ; Block 1
    MOVF 0x011, W ; W = 0x011 (b0)
    MULWF 0x001 ; a0 * b0

    MOVF PRODL, W
    ADDWF 0x023, F ; Save the low bits of a0 * b0
    MOVF PRODH, W
    ADDWF 0x022, F ; Save the high bits of a0 * b0

    ; Block 2
    MOVF 0x011, W ; W = 0x011 (b0)
    MULWF 0x000 ; a1 * b0

    MOVF PRODL, W
    ADDWF 0x022, F ; Save the low bits of a1 * b0
    BTFSC STATUS, C
    INCF 0x021
    MOVF PRODH, W
    ADDWF 0x021, F ; Save the high bits of a1 * b0
    BTFSC STATUS, C
    INCF 0x020

    ; Block 3
    MOVF 0x010, W ; W = 0x010 (b1)
    MULWF 0x001 ; a0 * b1

    MOVF PRODL, W
    ADDWF 0x022, F ; Save the low bits of a0 * b1
    BTFSC STATUS, C
```

```
54      INCF 0x021
55      BTFSC STATUS, C
56      INCF 0x020
57      MOVF PRODH, W
58      ADDWF 0x021, F ; Save the high bits of a0 * b1
59      BTFSC STATUS, C
60      INCF 0x020
61
62      ; Block 4
63      MOVF 0x010, W ; W = 0x010 (b1)
64      MULWF 0x000 ; a1 * b1
65
66      MOVF PRODL, W
67      ADDWF 0x021, F ; Save the low bits of a1 * b1
68      BTFSC STATUS, C
69      INCF 0x020
70      MOVF PRODH, W
71      ADDWF 0x020, F ; Save the high bits of a0 * b1
72
73  Over:
74      end
```

Bonus

▼ Answer

```
List p = 18f4520
#include<p18f4520.inc>
CONFIG OSC = INTIO67
CONFIG WDT = OFF
org 0x00

    CLRF 0x002
    CLRF 0x003
    MOVLW 0xFF
    MOVWF 0x000 ; Save the high bits
    MOVLW 0xF1
    MOVWF 0x001 ; save the low bits

    MOVLW 0x00
    CPFSGT 0x000 ; If 0x000 == 0 goto lower
    GOTO Lower
    MOVLW 0x08
    MOVWF 0x002
    MOVF 0x000, W
    MOVWF 0x004
    GOTO Divide_2

Lower:
    MOVF 0x001, W
    MOVWF 0x004
    GOTO Divide_2

Divide_2:
    RRCF 0x004
    BTFSC STATUS, C
    GOTO Rounding
    BCF STATUS, C
    MOVLW 0x00
    CPFSGT 0x004 ; If 0x004 == 0 goto Over
    GOTO Over
    INCF 0x002
    GOTO Divide_2

Rounding:
    INCF 0x003
    BCF STATUS, C
    MOVLW 0x00
    CPFSGT 0x004 ; If 0x004 == 0 goto Over
    GOTO Over
    INCF 0x002
    GOTO Divide_2

Over:
    MOVLW 0x00
    CPFSEQ 0x001 ; IF 0x001 > 0, 0x003++
    INCF 0x003

    MOVLW 0x03
```

```
54      CPFSLT 0x003 ; If 0x003 > 2 ,.do rounding
55      INCF 0x002
56
57      CLRF 0x03
58      end
```