

ESERCIZI SUI SISTEMI DI NUMERAZIONE

1°) Convertire in base 10 i seguenti numeri rappresentati nelle basi indicate:

- $(1000101)_2$ [R. 69]
- $(477)_8$ [R. 319]
- $(40F)_{16}$ [R. 1039]
- $(5778)_9$ [R. 4283]
- $(126)_9$ [R. 105]
- $(781)_{16}$ [R. 1921]
- $(3B8)_{13}$ [R. 658]
- $(10010)_8$ [R. 4104]
- $(2EA)_{16}$ [R. 746]
- $(5669)_{11}$ [R. 7456]
- $(889)_{12}$ [R. 1257]
- $(1110)_3$ [R. 1065]
- $(1357)_8$ [R. 751]

2°) Convertire i seguenti numeri decimali nelle basi specificate:

- 345 in base 2 [R. $(101011001)_2$]
- 345 in base 8 [R. $(531)_8$]
- 345 in base 16 [R. $(159)_{16}$]
- 989 in base 5 [R. $(12424)_5$]
- 417 in base 7 [R. $(1134)_7$]
- 615 in base 9 [R. $(753)_9$]
- 426 in base 2 [R. $(110101010)_2$]
- 1042 in base 11 [R. $(868)_{11}$]
- 6666 in base 16 [R. $(1A0A)_{16}$]

- 4596 in base 4 [R. (1013310)₄]
- 687 in base 16 [R. (2AF)₁₆]
- 595 in base 5 [R. (4340)₅]
- 111 in base 2 [R. (1101111)₂]
- 656 in base 5 [R. (10111)₅]
- 811 in base 16 [R. (32B)₁₆]
- 1101 in base 8 [R. (2115)₈]

3°) Convertire i seguenti numeri nelle basi specificate:

- (10010101001010)₂ in base 8 [R. (22512)₈]
- (11110101101000)₂ in base 16 [R. (3D68)₁₆]
- (13277)₈ in base 2 [R. (1011010111111)₂]
- (B0E9)₁₆ in base 2 [R. (1011000011101001)₂]
- (214)₅ in base 2 [R. (111011)₂]
- (354)₇ in base 6 [R. (510)₆]
- (821)₉ in base 12 [R. (477)₁₂]
- (821)₉ in base 8 [R. (1233)₈]
- (821)₉ in base 16 [R. (29B)₁₆]
- (AC29B)₁₆ in base 8 [R. (2541233)₈]
- (34772)₈ in base 16 [R. (39FA)₁₆]
- (1011)₉ in base 13 [R. (44B)₁₃]
- (312)₁₆ in base 4 [R. (30102)₄]
- (1492)₁₁ in base 15 [R. (87B)₁₅]
- (C14)₁₅ in base 16 [R. (A9F)₁₆]
- (C14)₁₅ in base 8 [R. (5237)₈]

ESERCIZI SULLE OPERAZIONI ARITMETICHE

1°) Eseguire le seguenti addizioni in binario:

- $34 + 77$ [R. $(1101111)_2$]
- $225 + 63$ [R. $(100100000)_2$]
- $229 + 111$ [R. $(101010100)_2$]
- $153 + 197$ [R. $(101011110)_2$]

2°) Eseguire le seguenti addizioni in ottale:

- $34 + 77$ [R. $(157)_8$]
- $225 + 63$ [R. $(440)_8$]
- $229 + 111$ [R. $(524)_8$]
- $153 + 197$ [R. $(536)_8$]

3°) Eseguire le seguenti addizioni in esadecimale:

- $34 + 77$ [R. $(6F)_{16}$]
- $225 + 63$ [R. $(120)_{16}$]
- $229 + 111$ [R. $(154)_{16}$]
- $153 + 197$ [R. $(15E)_{16}$]

4°) Eseguire le seguenti sottrazioni in binario:

- $84 - 37$ [R. $(101111)_2$]
- $225 - 63$ [R. $(10100010)_2$]
- $229 - 111$ [R. $(1110110)_2$]
- $321 - 123$ [R. $(11000110)_2$]
- $123 - 321$ [R. $(100111010)_2$]

5°) Eseguire le seguenti sottrazioni in ottale:

- $84 - 37$ [R. $(57)_8$]
- $225 - 63$ [R. $(242)_8$]
- $229 - 111$ [R. $(166)_8$]
- $321 - 123$ [R. $(306)_8$]
- $123 - 321$ [R. $(472)_8$]

6°) Eseguire le seguenti sottrazioni in esadecimale:

- $84 - 37$ [R. $(2F)_{16}$]
- $225 - 63$ [R. $(A2)_{16}$]
- $229 - 111$ [R. $(76)_{16}$]
- $321 - 123$ [R. $(C6)_{16}$]
- $123 - 321$ [R. $(F3A)_{16}$]

7°) Esegui la **moltiplicazione binaria** tra **63** e **30**, convertendo il risultato in decimale.

$$\begin{array}{r} 111111 \text{ X} \\ \underline{11110} = \\ 1233332100 \\ 1111110 + \\ 11111100 + \\ 111111000 + \\ \underline{1111110000} = \\ 11101100010 \end{array}$$

$$(11101100010)_2 = 2+32+64+256+512+1024 = 1890$$

8°) Esegui la **moltiplicazione binaria** tra **61** e **31**, convertendo il risultato in decimale.

$$\begin{array}{r} 111101 \text{ X} \\ \underline{11111} = \\ 1233332100 \\ 111101 + \\ 1111010 + \\ 11110100 + \\ 111101000 + \\ \underline{1111010000} = \\ 11101100011 \end{array}$$

$$(11101100011)_2 = 1+2+32+64+256+512+1024 = 1891$$

9°) Esegui la **moltiplicazione binaria** tra **59** e **31**, convertendo il risultato in decimale.

$$\begin{array}{r} 111011 \text{ X} \\ \underline{11111} = \\ 123333211 \\ 111011 + \\ 1110110 + \\ 11101100 + \\ 111011000 + \\ \underline{1110110000} = \\ 11100100101 \end{array}$$

$$(11100100101)_2 = 1+4+32+256+512+1024 = 1829$$

10°) Esegui la **moltiplicazione binaria** tra **55** e **31**, convertendo il risultato in decimale.

$$\begin{array}{r} 110111 \text{ X} \\ \underline{11111} = \\ 1223332210 \\ 110111 + \\ 1101110 + \\ 11011100 + \\ 110111000 + \\ \underline{1101110000} = \\ 11010101001 \end{array}$$

$$(11010101001)_2 = 1+8+32+128+512+1024 = 1705$$

11°) Esegui la **divisione binaria** tra **1979** e **99**, convertendo il quoziente ed il resto in decimale.

$$\begin{array}{r}
 11110111011 \mid \underline{1100011} \\
 \underline{1100011} \mid 10011 \\
 110001011 \mid \\
 \underline{1100011} \mid \\
 11000101 \mid \\
 \underline{1100011} \mid \\
 1100010 \mid
 \end{array}$$

$$(10011)_2 = 1+2+16 = 19$$

$$(1100010)_2 = 2+32+64 = 98$$

12°) Esegui la **divisione binaria** tra **1999** e **100**, convertendo il quoziente ed il resto in decimale.

$$\begin{array}{r}
 11111001111 \mid \underline{1100100} \\
 \underline{1100100} \mid 10011 \\
 110001111 \mid \\
 \underline{1100100} \mid \\
 11000111 \mid \\
 \underline{1100100} \mid
 \end{array}$$

$$1100011 \mid$$

$$(10011)_2 = 1+2+16 = 19$$

$$(1100011)_2 = 1+2+32+64 = 99$$

13°) Esegui la **divisione binaria** tra **1990** e **91**, convertendo il quoziente ed il resto in decimale.

$$\begin{array}{r}
 11111000110 \mid \underline{1011011} \\
 \underline{1011011} \mid 10101 \\
 1000010110 \mid \\
 \underline{1011011} \mid \\
 10101010 \mid \\
 \underline{1011011} \mid \\
 1001111 \mid
 \end{array}$$

$$(10101)_2 = 1+4+16 = 21$$

$$(1001111)_2 = 1+2+4+8+64 = 79$$

14°) Esegui la **divisione binaria** tra **1937** e **97**, convertendo il quoziente ed il resto in decimale.

$$\begin{array}{r}
 11110010001 \mid \underline{1100001} \\
 \underline{1100001} \mid 10011 \\
 110000001 \mid \\
 \underline{1100001} \mid \\
 10111111 \mid \\
 \underline{1100001} \mid \\
 1011110 \mid
 \end{array}$$

$$(10011)_2 = 1+2+16 = 19$$

$$(1011110)_2 = 2+4+8+16+64 = 94$$

ESERCIZI SUI NUMERI INTERI CON SEGNO

- 1°) Calcola i numeri rappresentati in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 dalla parola binaria **(11110001)₂**.

$$\text{In modulo e segno} = -(1+16+32+64) = -113$$

$$\text{In complemento a 1} = -(00001110)_2 = -(2+4+8) = -14$$

$$\text{In complemento a 2} = -(00001111)_2 = -(1+2+4+8) = -15$$

$$\text{In eccesso 127} = 1+16+32+64+128-127 = 114$$

- 2°) Calcola i numeri rappresentati in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 dalla parola binaria **(11100101)₂**.

$$\text{In modulo e segno} = -(1100101)_2 = -(1+4+32+64) = -101$$

$$\text{In complemento a 1} = -(00011010)_2 = -(2+8+16) = -26$$

$$\text{In complemento a 2} = -(00011011)_2 = -(1+2+8+16) = -27$$

$$\text{In eccesso 127} = 1+4+32+64+128-127 = 102$$

- 3°) Calcola i numeri rappresentati in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 dalla parola binaria **(11101001)₂**.

$$\text{In modulo e segno} = -(1101001)_2 = -(1+8+32+64) = -105$$

$$\text{In complemento a 1} = -(00010110)_2 = -(2+4+16) = -22$$

$$\text{In complemento a 2} = -(00010111)_2 = -(1+2+4+16) = -23$$

$$\text{In eccesso 127} = 1+8+32+64+128-127 = 106$$

- 4°) Calcola i numeri rappresentati in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 dalla parola binaria **(11100011)₂**.

$$\text{In modulo e segno} = -(1100011)_2 = -(1+2+32+64) = -99$$

$$\text{In complemento a 1} = -(00011100)_2 = -(4+8+16) = -28$$

$$\text{In complemento a 2} = -(00011101)_2 = -(1+4+8+16) = -29$$

$$\text{In eccesso 127} = 1+2+32+64+128-127 = 100$$

5°) Rappresenta con 8 bit in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 il numero **-99**.

In modulo e segno = 11100011

In complemento a uno = 10011100

In complemento a due = 10011100 + 1 = 10011101

In eccesso 127 = 01111111 -
01100011 =
00011100

6°) Rappresenta con 8 bit in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 il numero **-105**.

In modulo e segno = 11101001

In complemento a uno = 10010110

In complemento a due = 10010110 + 1 = 10010111

In eccesso 127 = 01111111 -
01101001 =
00010110

7°) Rappresenta con 8 bit in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 il numero **-101**.

In modulo e segno = 11100101

In complemento a uno = 10011010

In complemento a due = 10011010 + 1 = 10011011

In eccesso 127 = 01111111 -
01100101 =
00011010

8°) Rappresenta con 8 bit in modulo e segno, in complemento ad uno, in complemento a due ed in eccesso 127 il numero **-113**.

In modulo e segno = 11110001

In complemento a uno = 10001110

In complemento a due = 10001110 + 1 = 10001111

In eccesso 127 = 01111111 -
01110001 =
00001110

ADDIZIONI TRA NUMERI INTERI RAPPRESENTATI IN COMPLEMENTO A DUE

- 1°) Esegui l'**addizione binaria** tra le rappresentazioni in complemento a due di **-107** e **127**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 107 = 149 = 10010101_2 \\
 127 = 01111111_2 \\
 \hline
 \text{DEC}(00010100) = 4+16 = 20
 \end{array}
 \qquad
 \begin{array}{r}
 11111111 \\
 10010101_2 + \\
 \underline{01111111_2} = \\
 00010100_2
 \end{array}$$

- 2°) Esegui l'**addizione binaria** tra le rappresentazioni in complemento a due di **-99** e **119**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 99 = 157 = 10011101_2 \\
 119 = 01110111_2 \\
 \hline
 \text{DEC}(00010100) = 4+16 = 20
 \end{array}
 \qquad
 \begin{array}{r}
 11111111 \\
 10011101_2 + \\
 \underline{01110111_2} = \\
 00010100_2
 \end{array}$$

- 3°) Esegui l'**addizione binaria** tra le rappresentazioni in complemento a due di **-103** e **123**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 103 = 153 = 10011001_2 \\
 115 = 01111011_2 \\
 \hline
 \text{DEC}(00010100) = 4+16 = 20
 \end{array}
 \qquad
 \begin{array}{r}
 11111 \ 11 \\
 10011001_2 + \\
 \underline{01111011_2} = \\
 00010100_2
 \end{array}$$

- 4°) Esegui l'**addizione binaria** tra le rappresentazioni in complemento a due di **-95** e **115**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 95 = 161 = 10100001_2 \\
 115 = 01110011_2 \\
 \hline
 \text{DEC}(00010100) = 4+16 = 20
 \end{array}
 \qquad
 \begin{array}{r}
 111 \ \ 11 \\
 10100001_2 + \\
 \underline{01110011_2} = \\
 00010100_2
 \end{array}$$

SOTTRAZIONI TRA NUMERI INTERI RAPPRESENTATI IN COMPLEMENTO A DUE

- 1°) Esegui la **sottrazione binaria** tra le rappresentazioni in complemento a due di **-115** e **-95**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 115 = 141 = 10001101_2 \\
 256 - 95 = 161 = 10100001_2 \\
 \text{DEC}(11101100) = 4+8+32+64-128 = -20
 \end{array}
 \qquad
 \begin{array}{r}
 111 \\
 10001101_2 - \\
 \underline{10100001_2} = \\
 11101100_2
 \end{array}$$

- 2°) Esegui la **sottrazione binaria** tra le rappresentazioni in complemento a due di **-123** e **-103**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 123 = 133 = 10000101_2 \\
 256 - 103 = 153 = 10011001_2 \\
 \text{DEC}(11101100) = 4+8+32+64-128 = -20
 \end{array}
 \qquad
 \begin{array}{r}
 11111 \\
 10000101_2 - \\
 \underline{10011001_2} = \\
 11101100_2
 \end{array}$$

- 3°) Esegui la **sottrazione binaria** tra le rappresentazioni in complemento a due di **-119** e **-99**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 119 = 137 = 10001001_2 \\
 256 - 99 = 157 = 10011101_2 \\
 \text{DEC}(11101100) = 4+8+32+64-128 = -20
 \end{array}
 \qquad
 \begin{array}{r}
 111111 \\
 10001001_2 - \\
 \underline{10011101_2} = \\
 11101100_2
 \end{array}$$

- 4°) Esegui la **sottrazione binaria** tra le rappresentazioni in complemento a due di **-127** e **-107**, convertendo il risultato in decimale.

$$\begin{array}{r}
 256 - 127 = 129 = 10000001_2 \\
 256 - 107 = 149 = 10100001_2 \\
 \text{DEC}(11101100) = 4+8+32+64-128 = -20
 \end{array}
 \qquad
 \begin{array}{r}
 111111 \\
 10000001_2 - \\
 \underline{10010101_2} = \\
 11101100_2
 \end{array}$$

ESERCIZI SUI NUMERI IN VIRGOLA MOBILE

- 1°) Calcola il valore del numero rappresentato in virgola mobile dalla parola binaria corrispondente al numero esadecimale **(5E3C)₁₆**.

$$(5E3C)_{16} = (0101111000111100)_2$$

$$S = 0, \quad E = 10111, \quad M = 1000111100$$

$$e = 10111_2 - 15 = 1+2+3+16-15 = 8$$

$$m = (1,10001111)_2$$

$$x = (1,10001111)_2 \times 2^8$$

$$= (110001111)_2 = 1+2+4+8+128+256 = 399$$

- 2°) Calcola il valore del numero rappresentato in virgola mobile dalla parola binaria corrispondente al numero esadecimale **(5FCC)₁₆**.

$$(5FCC)_{16} = (0101111111001100)_2$$

$$S = 0, \quad E = 10111, \quad M = 1111001100$$

$$e = 10111_2 - 15 = 1+2+4+16-15 = 8$$

$$m = (1,11110011)_2$$

$$x = (1,11110011)_2 \times 2^8$$

$$= (111110011)_2 = 1+2+16+32+64+128+256 = 499$$

- 3°) Calcola il valore del numero rappresentato in virgola mobile dalla parola binaria corrispondente al numero esadecimale **(60AC)₁₆**.

$$(60AC)_{16} = (0110000010101100)_2$$

$$S = 0, \quad E = 11000, \quad M = 0010101100$$

$$e = 11000_2 - 15 = 8+16-15 = 9$$

$$m = (1,00101011)_2$$

$$x = (1,00101011)_2 \times 2^9$$

$$= (1001010110)_2 = 2+4+16+64+512 = 598$$

- 4°) Calcola il valore del numero rappresentato in virgola mobile dalla parola binaria corrispondente al numero esadecimale **(6174)₁₆**.

$$(6174)_{16} = (0110000101110100)_2$$

$$S = 0, \quad E = 11000, \quad M = 0101110100$$

$$e = 11000_2 - 15 = 8+16-15 = 9$$

$$m = (1,01011101)_2$$

$$x = (1,01011101)_2 \times 2^9$$

$$= (1010111010)_2 = 2+8+16+32+128+512 = 698$$

- 5°) Rappresenta in virgola mobile con 16 bit il numero **0,1**, convertendo il risultato in esadecimale.

$$0,1 \approx (0,00011001100110)_2 = (1,1001100110)_2 \times 2^{-4}$$

$$(E)_2 = 15 - 4 = 11 \quad E = 01011, \quad M = 1001100110$$

$$(W)_2 = (0010111001100110)_2 = (2E66)_{16}$$

- 6°) Rappresenta in virgola mobile con 16 bit il numero **0,3**, convertendo il risultato in esadecimale.

$$0,3 \approx (0,010011001101)_2 = (1,0011001101)_2 \times 2^{-2}$$

$$(E)_2 = 15 - 2 = 13 \quad E = 01101, \quad M = 0011001101$$

$$(W)_2 = (0011010011001101)_2 = (34CD)_{16}$$

- 7°) Rappresenta in virgola mobile con 16 bit il numero **0,7**, convertendo il risultato in esadecimale.

$$0,7 \approx (0,10110011010)_2 = (1,0110011010)_2 \times 2^{-1}$$

$$(E)_2 = 15 - 1 = 14 \quad E = 01110, \quad M = 0110011001$$

$$(W)_2 = (0011100110011010)_2 = (399A)_{16}$$

- 8°) Rappresenta in virgola mobile con 16 bit il numero **0,9**, convertendo il risultato in esadecimale.

$$0,9 \approx (0,11100110011)_2 = (1,1100110011)_2 \times 2^{-1}$$

$$(E)_2 = 15 - 1 = 14 \quad E = 01110, \quad M = 1100110011$$

$$(W)_2 = (0011101100110011)_2 = (3B33)_{16}$$

ESERCIZI NEL LINGUAGGIO ASSEMBLATIVO DEL MICROCONTROLORE PIC 16F84

- 1°) Calcolare in binario e convertire in decimale il contenuto del registro W dopo l'esecuzione delle seguenti istruzioni:

```

MOVLW    150 ;W = (10010110)2 and
ANDLW    170 ;   (10101010)2 =
           ;   (10000010)2 = 2 + 128 = 130
    
```

- 2°) Calcolare in binario e convertire in decimale il contenuto del registro W dopo l'esecuzione delle seguenti istruzioni:

```

MOVLW    45  ;W = (00101101)2 ior
IORLW    180 ;   (10110100)2 =
           ;   (10111101)2 = 1 + 4 + 8 + 16 +
           ;                       32 + 128 = 189
    
```

- 3°) Calcolare in binario e convertire in decimale il contenuto del registro W dopo l'esecuzione delle seguenti istruzioni:

```

MOVLW    60  ;W = (00111100)2 xor
XORLW    165 ;   (10100101)2 =
           ;   (10011001)2 = 1 + 8 + 16 + 128 =
           ;                       = 153
    
```

- 4°) Calcolare in binario e convertire in decimale il contenuto del registro W ed il valore dei *flag* C, DC e Z dopo l'esecuzione delle seguenti istruzioni:

```

           ;           11011001
MOVLW    205 ;   W = (11001101)2 +
ADDLW    217 ;   (11011001)2 =
           ;   (10100110)2 = 2 + 4 + 32 +
           ;                       + 128 = 166
           ;   C = 1, DC = 1, Z = 0
    
```

- 5°) Calcolare in binario e convertire in decimale il contenuto del registro W ed il valore dei *flag* C, DC e Z dopo l'esecuzione delle seguenti istruzioni:

```

           ;           11001111
MOVLW    237 ;   W = (10111000)2 -
SUBLW    184 ;   (11101101)2 =
           ;   (11001011)2 = 1 + 2 + 8 + 64
           ;                       + 128 = 203
           ;   C = 0, DC = 0, Z = 0
    
```

- 6°) Trasformare il contenuto del registro d'indirizzo (20)₁₆ dal codice ASCII di una lettera in forma maiuscola al codice ASCII della stessa lettera in forma minuscola.

```

MOVLW    20H ;W = 20h
ADDWF    20H,F ; [20h] = [20h] + 20h
    
```

- 7°) Copiare nel registro di lavoro W la rappresentazione binaria della cifra decimale il cui codice ASCII è contenuto nel registro d'indirizzo (20)₁₆.

```
MOVF      20H,W      ;W = [20H]
ANDLW    00001111B ;W = W AND 00001111B (bit a bit)
```

- 8°) Scambiare tra loro i contenuti dei registri d'indirizzo (20)₁₆ e (30)₁₆.

```
MOVF      20H,W      ;W = [20H]
MOVWF    40H         ;[40H] = W
MOVF      30H,W      ;W = [30H]
MOVWF    20H         ;[20H] = W
MOVF      40H,W      ;W = [40H]
MOVWF    30H         ;[30H] = W
```

- 9°) Invertire il livello del segnale presente su ciascuna delle quattro linee d'uscita RB4, RB5, RB6 ed RB7, senza modificare il segnale sulle altre linee della porta B.

```
MOVLW    11110000b ;W = 11110000b
XORWF    PORTB,F   ;PORTB = PORTB XOR W
```

- 10°) Sommare al contenuto del registro d'indirizzo (20)₁₆ quello del registro d'indirizzo (30)₁₆ solo se il risultato è minore di 256.

```
MOVF      20H,W      ;W = [2016].
ADDWF    30H,W      ;W = ([2016] + [3016]) mod 256.
BTFSS    STATUS,C   ;Se il riporto finale è zero,
MOVWF    20H         ;allora [2016] = W.
```

- 11°) Sottrarre al contenuto del registro d'indirizzo (20)₁₆ quello del registro d'indirizzo (30)₁₆ solo se il risultato è maggiore o uguale a zero.

```
MOVF      30H,W      ;W = [3016].
SUBWF    20H,W      ;W = ([2016] - [3016]) mod 256.
BTFSC    STATUS,C   ;Se il prestito finale è zero,
MOVWF    20H         ;allora [2016] = W.
```

- 12°) Traslare di una posizione verso sinistra tutti i bit del codice binario contenuto nel registro d'indirizzo (20)₁₆, copiando il valore del bit più significativo nella posizione di ordine zero.

```
MOVF      20H,W      ;W = [20H]
ADDWF    20H,F      ;[20H] = [20H] * 2
MOVF      STATUS,W   ;W = STATUS
ANDLW    00000001B ;W = Carry
ADDWF    20H,F      ;[20H] = [20H] + Carry
```

- 13°) Incrementare di uno il numero binario di 16 cifre la cui parte bassa è contenuta nel

registro d'indirizzo (20)₁₆ e la cui parte alta è contenuta nel registro d'indirizzo (21)₁₆.

```
MOVLW    1           ;W = 1
ADDWF    20H,F       ;[20h] = [20h] + W = [20h] + 1
MOVF     STATUS,W   ;W = STATUS
ANDLW    00000001B  ;W = STATUS and (00000001)2
ADDWF    21H,F       ;[21h] = [21h] + Carry
```

14°) Moltiplicare per due il numero binario di 16 cifre la cui parte bassa è contenuta nel registro d'indirizzo (20)₁₆ e la cui parte alta è contenuta nel registro d'indirizzo (21)₁₆.

```
MOVF     20H,W       ;W = [20h]
ADDWF    20H,F       ;[20h] = [20h] + W = [20h] * 2
MOVF     STATUS,W   ;W = STATUS
ANDLW    00000001B  ;W = STATUS and (00000001)2
ADDWF    21H,W       ;W = [21h] + Carry
ADDWF    21H,F       ;[21h] = [21h] + W =
                    ; = [21h] * 2 + Carry
```

15°) Configurare le linee RA0, RA1, RA2 ed RA3 in ingresso; la linea RA4 in uscita ed a livello alto.

```
BSF      STATUS,RP0 ;Seleziona TRISA.
MOVLW    00001111B  ;Imposta RA0..3 in ingresso
MOVWF    TRISA      ;Imposta RA4 in uscita.
BCF      STATUS,RP0 ;Seleziona PORTA.
BSF      PORTA,4    ;Poni RA4 a livello alto.
```

16°) Simulare una porta logica NOT con ingresso la linea RA0 ed uscita la linea RA4 (si supponga che RA0 sia già impostata in ingresso ed RA4 in uscita).

```
CICLO:   BTFSS      PORTA,0 ;Se RA0 è a livello basso
         GOTO       RA0L    ;allora vai a RA0L.
RA0H:    BCF        PORTA,4 ;Poni RA4 a livello basso.
         GOTO       CICLO   ;Vai a CICLO.
RA0L:    BSF        PORTA,4 ;Poni RA4 a livello alto.
         GOTO       CICLO   ;Vai a CICLO.
```

17°) Simulare una porta AND con ingressi le linee RA0 ed RA1 ed uscita la linea RA4 (si supponga che RA0 ed RA1 siano già impostate in ingresso ed RA4 in uscita).

```
CICLO:   BTFSC      PORTA,0 ;Se RA4 è a livello alto
         GOTO       RA0H    ;allora vai a RA0H.
RA0L:    BCF        PORTA,4 ;Poni RA4 a livello basso.
         GOTO       CICLO   ;Vai a CICLO.
RA0H:    BTFSS      PORTA,1 ;Se RA4 è a livello basso
         GOTO       RA0L    ;allora vai a RA0L.
         BSF        PORTA,4 ;Poni RA4 a livello alto.
         GOTO       CICLO   ;Vai a CICLO.
```

18°) Simulare una porta OR con ingressi le linee RA0 ed RA1 ed uscita la linea RA4 (si supponga che RA0 ed RA1 siano già impostate in ingresso ed RA4 in uscita).

```
CICLO:    BTFSS    PORTA,0    ;Se RA4 è a livello basso
          GOTO    RA0L      ;allora vai a RA0L.
RA0H:    BSF     PORTA,4    ;Poni RA4 a livello alto.
          GOTO    CICLO     ;Vai a CICLO.
RA0L:    BTFSC   PORTA,1    ;Se RA4 è a livello alto
          GOTO    RA0H      ;allora vai a RA0H.
          BCF     PORTA,4    ;Poni RA4 a livello basso.
          GOTO    CICLO     ;Vai a CICLO.
```

19°) Simulare una porta EXOR con ingressi le linee RA0 ed RA1 ed uscita la linea RA4 (si supponga che RA0 ed RA1 siano già impostate in ingresso ed RA4 in uscita).

```
CICLO:    BTFSS   PORTA,0    ;Se RA0 è a livello basso
          GOTO    RA0L      ;allora vai a RA0L.
RA0H:    BTFSS   PORTA,1    ;Se RA1 è a livello basso
          GOTO    RA1L      ;allora vai a RA1L.
RA1H:    BCF     PORTA,4    ;Poni RA4 a livello basso.
          GOTO    CICLO     ;Vai a CICLO.
RA0L:    BTFSS   PORTA,1    ;Se RA0 è a livello basso
          GOTO    RA1H      ;allora vai a RA1H.
RA1L:    BSF     PORTA,4    ;Poni RA4 a livello alto.
          GOTO    CICLO     ;Vai a CICLO.
```

20°) Simulare una porta NAND con ingressi le linee RA0 ed RA1 ed uscita la linea RA4 (si supponga che RA0 ed RA1 siano già impostate in ingresso ed RA4 in uscita).

```
CICLO:    BTFSS   PORTA,0    ;Se RA0 è a livello basso
          GOTO    HIGH      ;allora vai a HIGH.
          BTFSS   PORTA,1    ;Se RA1 è a livello basso
          GOTO    HIGH      ;allora vai a HIGH.
LOW:      BCF     PORTA,4    ;Poni RA4 a livello basso.
          GOTO    CICLO     ;Vai a CICLO.
HIGH:     BSF     PORTA,4    ;Poni RA4 a livello alto.
          GOTO    CICLO     ;Vai a CICLO.
```

21°) Simulare una porta NOR con ingressi le linee RA0 ed RA1 ed uscita la linea RA4 (si supponga che RA0 ed RA1 siano già impostate in ingresso ed RA4 in uscita).

```
CICLO:    BTFSC   PORTA,0    ;Se RA0 è a livello alto
          GOTO    LOW       ;allora vai a LOW.
          BTFSC   PORTA,1    ;Se RA1 è a livello alto
          GOTO    LOW       ;allora vai a HIGH.
HIGH:     BSF     PORTA,4    ;Poni RA4 a livello alto.
          GOTO    CICLO     ;Vai a CICLO.
LOW:      BCF     PORTA,4    ;Poni RA4 a livello basso.
          GOTO    CICLO     ;Vai a CICLO.
```

- 22°) Supponendo che il tempo d'esecuzione di ogni istruzione del linguaggio macchina del PIC16F84 sia pari ad un microsecondo, scrivere un programma in linguaggio assembler per produrre sulla linea RA4 un segnale periodico con forma d'onda quadra e frequenza pari a 125 Khz.

```

BSF      STATUS,RP0;Seleziona TRISA.
BCF      TRISA,4   ;Imposta RA4 in uscita.
BCF      STATUS,RP0;Seleziona PORTA.

CICLO:   BSF      PORTA,4   ;Poni RA4 a livello alto.
          BSF      PORTA,4   ;Poni RA4 a livello alto.
          BSF      PORTA,4   ;Poni RA4 a livello alto.
          BSF      PORTA,4   ;Poni RA4 a livello alto.

          BCF      PORTA,4   ;Poni RA4 a livello basso.
          BCF      PORTA,4   ;Poni RA4 a livello basso.
          BCF      PORTA,4   ;Poni RA4 a livello basso.
          GOTO     CICLO     ;Ripeti il ciclo.

```

;Poiché il tempo d'esecuzione di ogni istruzione è un microsecondo, il tempo necessario ad eseguire il corpo di questo ciclo è uguale a 8 microsecondi. Nella prima metà di questo periodo il segnale sulla linea RA4 è a livello alto, nella seconda metà è a livello basso. Perciò, sulla linea RA4 viene generato un segnale periodico con forma d'onda quadra e frequenza pari a 125 Khz.

- 23°) Scrivere un programma in linguaggio assembler che faccia funzionare il PIC16F84 come cinque porte NOT con ingressi le linee RA0, RA1, RA2, RA3 ed RA4 ed uscite le linee RB0, RB1, RB2, RB3 ed RB4 rispettivamente (la prima con ingresso RA0 ed uscita RB0, la seconda con ingresso RA1 ed uscita RB1 e così via).

```

BSF      STATUS,RP0   ;Seleziona TRISA e TRISB.
MOVLW   11111B       ;Imposta tutte le linee
MOVWF   TRISA        ;della porta A in ingresso.
MOVLW   00000000B    ;Imposta tutte le linee
MOVWF   TRISB        ;della porta B in uscita.
BCF      STATUS,RP0   ;Seleziona PORTA e PORTB.

CICLO:   MOVF      PORTA,W   ;W = PORTA.
          XORLW   11111111B  ;W = NOT PORTA.
          MOVWF   PORTB      ;PORTB = W = NOT PORTA.
          GOTO     CICLO     ;Ripeti il ciclo.

```

- 24°) Scrivere un programma in linguaggio assembler che faccia funzionare il PIC16F84 come quattro porte OR a due ingressi: la prima porta con ingressi le linee RA0 ed RB0 e con uscita la linea RB4, la seconda porta con ingressi RA1 ed RB1 e con uscita RB5, la terza porta con ingressi RA2 ed RB2 e con uscita RB6, la quarta porta con ingressi RA3 ed RB3 e con uscita RB7.

```

BSF      STATUS,RP0   ;Seleziona TRISA e TRISB.
MOVLW   11111B       ;Imposta tutte le linee
MOVWF   TRISA        ;della porta A in ingresso.
MOVLW   00001111B    ;Imposta RB0..3 in ingresso.

```

```

MOVWF    TRISB           ;Imposta RB4..7 in uscita.
BCF      STATUS,RP0     ;Seleziona PORTA e PORTB.

CICLO:   MOVF            PORTA,W           ;W = PORTA.
         IORWF           PORTB,W         ;W = PORTA or PORTB.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          PORTB           ;PORTB = W.
         GOTO          CICLO

```

- 25°) Scrivere un programma in linguaggio assembler che faccia funzionare il PIC16F84 come quattro porte AND a due ingressi: la prima porta con ingressi le linee RA0 ed RB0 e con uscita la linea RB4, la seconda porta con ingressi RA1 ed RB1 e con uscita RB5, la terza porta con ingressi RA2 ed RB2 e con uscita RB6, la quarta porta con ingressi RA3 ed RB3 e con uscita RB7.

```

BSF      STATUS,RP0     ;Seleziona TRISA e TRISB.
MOVLW   11111B         ;Imposta tutte le linee
MOVWF   TRISA          ;della porta A in ingresso.
MOVLW   00001111B     ;Imposta RB0..3 in ingresso.
MOVWF   TRISB          ;Imposta RB4..7 in uscita.
BCF      STATUS,RP0     ;Seleziona PORTA e PORTB.

CICLO:   MOVF            PORTA,W           ;W = PORTA.
         ANDWF          PORTB,W         ;W = PORTA and PORTB.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          FSR              ;FSR = W.
         ADDWF          FSR,W           ;W = 2*FSR.
         MOVWF          PORTB           ;PORTB = W.
         GOTO          CICLO

```

- 26°) Scrivere un programma in linguaggio assembler che faccia funzionare il PIC16F84 come un convertitore da codice BCD a codice a sette segmenti con ingressi le linee RA0, RA1, RA2 ed RA3 ed uscite le linee RB0 (segmento a), RB1 (segmento b), RB2 (segmento c), RB3 (segmento d), RB4 (segmento e), RB5 (segmento f), RB6 (segmento g) ed RB7 (punto decimale).

```

BSF      STATUS,RP0     ;Seleziona TRISA e TRISB.
MOVLW   11111B         ;Imposta tutte le linee
MOVWF   TRISA          ;della porta A in ingresso.
MOVLW   00000000B     ;Imposta tutte le linee
MOVWF   TRISB          ;della poorta B in uscita.
BCF      STATUS,RP0     ;Seleziona PORTA e PORTB.

```

```

MOVLW    00111111B    ;Copia in Reg[20h] il
MOVWF    20H          ;codice a 7 segmenti di 0.
MOVLW    00000110B    ;Copia in Reg[21h] il
MOVWF    21H          ;codice a 7 segmenti di 1.
MOVLW    01011011B    ;Copia in Reg[22h] il
MOVWF    22H          ;codice a 7 segmenti di 2.
MOVLW    01001111B    ;Copia in Reg[23h] il
MOVWF    23H          ;codice a 7 segmenti di 3.
MOVLW    01100110B    ;Copia in Reg[24h] il
MOVWF    24H          ;codice a 7 segmenti di 4.
MOVLW    01101101B    ;Copia in Reg[25h] il
MOVWF    25H          ;codice a 7 segmenti di 5.
MOVLW    01111101B    ;Copia in Reg[26h] il
MOVWF    26H          ;codice a 7 segmenti di 6.
MOVLW    00000111B    ;Copia in Reg[27h] il
MOVWF    27H          ;codice a 7 segmenti di 7.
MOVLW    01111111B    ;Copia in Reg[28h] il
MOVWF    28H          ;codice a 7 segmenti di 8.
MOVLW    01101111B    ;Copia in Reg[29h] il
MOVWF    29H          ;codice a 7 segmenti di 9.

```

```

CICLO:   MOVF    PORTA,W    ;W = contenuto della porta A
         ANDLW   00001111B  ;W = codice BCD in ingresso.
         ADDLW   20H        ;FSR = 20h + codice BCD.
         MOVWF   FSR
         MOVF    INDF,W     ;W = [20h + codice BCD].
         MOVWF   PORTB     ;PORTB = W.
         GOTO    CICLO     ;Ripeti il ciclo.

```