Esercizi relativi al linguaggio ISA IJVM

1. Descrivere in linguaggio IJVM un algoritmo che scambia fra di loro due interi X e Y.

   Soluzione:

```
        ILOAD   X                   / carica X sullo stack
        ILOAD   Y                   / carica Y sullo stack
        ISTORE  X                   / mette il valore di Y in X
        ISTORE  Y                   / mette il valore di X in Y
```

2. Descrivere in linguaggio IJVM un algoritmo che dati 2 interi A e B, pone in A il $min[A, B]$ e in B il $max[A, B]$.

   Soluzione:

```
        ILOAD   A                   / carica A sullo stack
        ILOAD   B                   / carica B sullo stack
        ISUB                        / A-B
        IFLT    L1                  / se A<B vai a L1
        ILOAD   A                   / carica A sullo stack
        ILOAD   B                   / carica B sullo stack
        ISTORE  A                   / mette il valore di B in A
        ISTORE  B                   / mette il valore di A in B
 L1:    .......
```

3. Scrivere uno spezzone di programma in linguaggio IJVM che somma i numeri interi da 1 a 32.

   Soluzione: Con ciclo a condizione finale:

```
        BIPUSH    0
        ISTORE    sum                 / azzera somma
        BIPUSH    0
        ISTORE    cont                / azzera contatore

 INI:   IINC      cont 1              / incrementa contatore
        ILOAD     cont                / carica cont sullo stack
        ILOAD     sum                 / carica sum sullo stack
        IADD                         / sum+cont
        ISTORE    sum                 / memorizza somma
        ILOAD     cont                / carica cont sullo stack
        BIPUSH    32                  / carica la costante 32
        IF_ICMPEQ L2                  / se cont=32 va a L2
        GOTO      INI                 / va a INI
 L2:    HALT
```

   Soluzione: Con ciclo a condizione iniziale:

```
      BIPUSH   0
      ISTORE   sum                    / azzera somma
      BIPUSH   0
      ISTORE   cont                   / azzera contatore

INI:  IINC     cont 1                 / incrementa contatore
      ILOAD    cont                   / carica cont sullo stack
      BIPUSH   32                     / carica la costante 32
      ISUB                            / cont-32
      IFLT     LL                     / se cont-32<0 vai a LL
      ILOAD    cont                   / carica cont sullo stack
      BIPUSH   32                     / carica la costante 32
      ISUB                            / cont-32
      IFEQ     LL                     / se cont-32=0 vai a LL
      GOTO     FIN                    / vai a FIN (cont>32)
LL:   ILOAD    cont                   / carica cont sullo stack
      ILOAD    sum                    / carica sum sullo stack
      IADD                            / sum+cont
      ISTORE   sum                    / memorizza somma
      GOTO     INI
FIN:  HALT
```

4. Scrivere uno spezzone di programma in linguaggio IJVM che data una parola X (di 32 bit) calcola un intero C che rappresenta il numero di bit uguali a 1 nella stringa binaria di X.

   Soluzione 1: Un test sul valore del bit più significativo di una stringa binaria interpretata come numero intero in complemento a 2 può essere realizzato come test sul segno. E' possibile far scorrere ad uno ad uno i bit di una stringa binaria nella posizione più significativa tramite uno shift a sinistra (realizzato come somma del numero con se stesso).

```
      BIPUSH  0
      ISTORE  C                /1/ azzera contatore C
LOOP: ILOAD   X                /2/ carica X sullo stack
      IFEQ    FIN              /3/ se X=0 vai a FIN
      ILOAD   X                /4/ carica X sullo stack
      IFLT    NEG              /5/ se X<0 vai a NEG
      GOTO    L1               /6/ vai a L1
NEG:  IINC    C 1              /7/ incrementa C
L1:   ILOAD   X                /8/ carica X sullo stack
      DUP                      /9/ copia X sullo stack
      IADD                    /10/ carica X+X sullo stack
      ISTORE  X               /11/ memorizza il nuovo valore di x
      GOTO    LOOP            /12/ vai a LOOP
FIN:  ILOAD   C               /14/ carica C sullo stack
      HALT                    /15/
```

5. Disegnare la struttura dello stack (variabili + operandi) dopo ogni istruzione dell'esercizio 4.

6. Descrivere (come nell'esercizio 2) in linguaggio IJVM un programma che dati 2 interi A e B, pone in A il $min[A, B]$. Il programma deve essere strutturato con programma principale e una procedura che operano nel seguente modo. Il programma principale trasferisce i parametri interi A e B a un metodo, che restituisce

il minimo fra i due interi. Il programma principale carica il minimo ottenuto dalla procedura in A. Questo esercizio è una versione modularizzata dell'esercizio 2.

Soluzione:

```
        .....
        LDC_W OBJREF                / prepara lo stack per la chiamata al metodo
        ILOAD A                     / carica il parametro A sullo stack
        ILOAD B                     / carica il parametro B sullo stack
        INVOKEVIRTUAL minimo        / invoca il metodo minimo
        ISTORE A                    / carica il valore di ritorno [min] in A
        .....

.method minimo(locA,locB)
        ILOAD   locA                / carica locA sullo stack
        ILOAD   locB                / carica locB sullo stack
        ISUB                        / A-B
        IFLT    L1                  / se A<B vai a L1
        ILOAD   locB                / carica locB sullo stack
  L1:   ILOAD   locA                / carica sullo stack il valore di ritorno
        IRETURN                     / restituisce il controllo al chiamante
.end-method
```

7. Scrivete un programma il linguaggio ISA-IJVM per trasferire i due byte più significativi di una variabile locale A nei due byte meno significativi di una variabile locale Y.

   (Nota per caricare sullo stack una variabile a 32 bit, bisogna creare una nuova istruzione chiamata $BISH8PU < byte >$, che shifta la parola affiorante sullo stack e carica $< byte >$ nel byte meno significativo. Per il μ-interprete MIC-1 di questa nuova sitruzione vedere gli esercizi relativi a MIC-1)

   Soluzione:

```
        .constant
OBJREF 0x40 .end-constant

.main

.var a y c .end-var

       //devo creare "a", parola di 4 byte

       bipush   0xb2    //  byte + significativo
       bish8pu  0xd3
       bish8pu  0x3f
       bish8pu  0x10    //  byte - significativo
       istore a         //  a=0xb2d33f10

       //devo creare "y", parola di 4 byte

       bipush 0x26      //  byte + significativo
       bi8pu  0xf4
       bi8pu  0x10
       bi8pu  0xba      //  byte - significativo
       istore y         //  y=0x26f410ba
```

```
       bipush 0
       istore c

 ciclo:bipush 16
       iload c
       if_icmpeq fine
       iload y          // shifto y
       dup              // di un
       iadd             // bit (bisogna farlo comunque)
       iload a          // e
       iflt l1          // se a<0 (bit + significativo=1)
       goto l2
    l1:bipush 1         // trasferisco 1 come bit
       iadd             // meno significativo di y

     l2:istore y

       iload a           // shifto a
       dup               // di un
       iadd              // bit
       istore a          // (bisogna farlo comunque)
       iinc c 1
       goto ciclo

  fine:iload y           // coi dati caricati y in cima allo stack
       halt              // sia: TOS=10bab2d3
.end-main
```