Practico 6

1)

a)

1.1)

```
ADD X0, X1, X2
ADD X0, X0, X3
ADD X0, X0, X4
```

b)

1.1)

```
ADDI X0, X2, #5
ADD X0, X0, X1
```

c)

1.1)

```
  ADD X0, X1, X2
  ADD X0, X0, X0
```

2)

a)

2.1)

```
  f = g + h
```

2)

a)

2.1)

```
  f = g + h
```

3)

a)

3.1)

```
  ADD X0, X0, X1
  SUB X0, XZR, X0
```

3)

b)

3.1)

```
ADD X0, X0, X1
SUB X0, XZR, X0
```

4)

a)

4.1)

```
SUB X1, XZR, X1 // g = -g
ADD X0, X1, X2  // f = g + h

f = -g + h
```

4)

b)

4.1)

```
ADDI X2, X0, #1 // h = f + 1
SUB X0, X1, X2  // f = g + h

f = g + (f + 1)
```

5)

a)

5.1)

```
LDUR X0, [X6, #32]
ADD X0, X0, X1
SUB X0, XZR, X0
```

5.2) Se utilizaron 4 registros

5)

b)

5.1)

```
SUB X8, X2, X3
LSL X8, X8, #3 //2^3 = 8
```

```
ADD X9, X6, X8
LDUR X10, [X9, #0]
STUR X10, [X7, #64]
```

5.2) Se usaron 7 registros

6)

a)

1)

```
h = j * 2
f = h + j + j
```

2)

```
LSL X0, X4, #2
```

b)

1)

```
LSL X9, X3, #3      // k  = i * 8
ADD X9, X6, X9      // k  = &A + (i * 8) = &A[i]
LSL X10, X4, #3     // l = j * 8
ADD X10, X7, X10    // l = &B + (j * 8) = &B[j]
LDUR X12, [X9, #0]  // n = A[i]
ADDI X11, X9, #8    // m = &A[i] + 8 = &A[i+1]
LDUR X9, [X11, #0]  // k  = A[i+1]
ADD X9, X9, X12     // k  = A[i+1] + A[i]
STUR X9, [X10, #0]  // B[j] = A[i+1] + A[i]
```

Sentencia minima en C:

```
B[j] = A[i+1] + A[i]
```

2)

```
LSL X9, X3, #3      // k  = i * 8
ADD X9, X6, X9      // k  = &A + (i * 8) = &A[i]
LSL X10, X4, #3     // l = j * 8
ADD X10, X7, X10    // l = &B + (j * 8) = &B[j]
LDUR X12, [X9, #0]  // n = A[i]

LDUR X9, [X11, #8]  // k  = A[i+1]
ADD X9, X9, X12     // k  = A[i+1] + A[i]
STUR X9, [X10, #0]  // B[j] = A[i+1] + A[i]
```

7)

1)

```
ADDI X9, X6, #8     // x9   = &A[0]+ 8 = &A[1]
ADD  X10, X6, XZR   // x10  = &A[0]+ 0 = &A[0]
STUR X10, [X9, #0]  // A[1] = &A[0]
LDUR X9, [X9, #0]   // x9   = A[1]     = &A[0]
ADD  X0, X9, X10    // f    = &A[0] + &A[0]
```

Sentencia minima en C:

```
f = &A[0] + &A[0]
```

2)

8)

a)

1)

```
lsl x11, x9, #4   // x11 = 0x555555550
orr x11, x11, x10 // x11 = x11 | x10

x11     = 0101 0101 0101 0101 0101 0101 0101 0101 0000
x10     = 0000 0001 0010 0011 0100 0101 0110 0111 1000
x11|x10 = 0101 0101 0111 0111 0101 0101 0111 0111 1000
```

2)

```
lsl  x11, x10, #4     // x11 = 0x123456780
andi x11, x11, #0xfff // x11 = x11 & 0xfff

x11   = 0001 0010 0011 0100 0101 0110 0111 1000 0000
0xfff = 0000 0000 0000 0000 0000 0000 1111 1111 1111
x11   = 0000 0000 0000 0000 0000 0000 0111 1000 0000
```

3)

9)

```
LSR X10, X9, 24
```

10)

```
LSR X10, X9, 30
```

11)

```
MOVZ X0, 0x1234, LSL 48 // X0 = 0x1234000000000000

MOVZ X1, 0xBBB, LSL 52
MOVK X1, 0xAAA, LSL 0   // X1 = 0xBBB0000000000AAA

MOVZ X2, 0xA0A0, LSL 48
MOVK X2, 0xB1B1, LSL 32
MOVK X2, 0xC2C2, LSL 0  // X2 = 0xA0A0B1B10000C2C2

MOVZ X3, 0x1234, LSL 44
MOVK X3, 0x5678, LSL 28
MOVK X3, 0x9ABC, LSL 12
MOVK X3, 0xDEF, LSL 0   // X3 = 0x0123456789ABCDEF
```