(a)

```
1   forall i in s..e by n {
2        //affine array expressions
3        A1[a1*i+b1] = A2[a2*i+b2] + 3;
4   }
```

(b)

```
1   for k in 0..((lcm(B,n)/n)-1) {
2        forall i in (s+k*n)..e by lcm(B,n) {
3             //affine array expressions
4             A1[a1*i+b1] = A2[a2*i+b2] + 3;
5   }    }
```

(c)

```
1   for k in 0..((lcm(B,n)/n)-1) {
2      for j in 0..N-1 {
3         if(f(s+k*n+lcm(B,n)*j)/B mod N == $) {
4            //fetch elements from affine array expressions
5           //that are not owning expressions of the loop
6            var buf1 = GET(A2[(s+k*n+lcm(B,n)*j)+b2..e+b2 by N*lcm(B,n)*a2]);
7            var h = 0;
8            forall i in (s+k*n+lcm(B,n)*j)..e by lcm(B,n)*N {
9               //affine array expressions
10              A1[a1*i+b1] = buf1[h] + 3;
11              h++;
12           }
13           //write buffer elements back if written to during loop
14          if(buf1_is_modified)
15             SET(A2[(s+k*n+lcm(B,n)*j)+b2..e+b2 by N*lcm(B,n)*a2]) = buf1;
16  }  }  }
```

(d)

s = starting loop bound
e = ending loop bound
n = loop stride
B = block size
N = number of locales
$ = current locale identifier