Tuning Lab

1 Identify the tuning tips in the following PL/SQL block:

DECLARE

CURSOR C IS

SELECT ENAME, SAL, COMM

FROM EMPLOYEES;

L\_COMM NUMBER;

BEGIN

FOR I IN C

LOOP

L\_COMM := I.SAL + ((I.COMM/100) \* (I.SAL \* 12));

DBMS\_OUTPUT.PUT\_LINE (I.ENAME||' earns '||L\_COMM||' as

commission');

END LOOP;

END;

/

a. Use BULK COLLECT to select employee data.

b. Declare L\_COMM as NOT NULL.

c. Use PLS\_INTEGER for L\_COMM.

d. No tuning required.

1. Examine the following code and determine the output:

DECLARE

FUNCTION F\_ADD (P\_NUM NUMBER)

RETURN NUMBER

IS

BEGIN

RETURN P\_NUM + 10;

END;

BEGIN

FOR I IN 122..382

LOOP

PRAGMA INLINE (F\_ADD,'YES');

L\_SUM := L\_SUM + F\_ADD (I);

END LOOP;

END;

/

PLSQL\_OPTIMIZE\_LEVEL is set as 2.

a. The local function F\_ADD would not be called inline unless

PLSQL\_OPTIMIZE\_LEVEL is set as 3.

b. The local function F\_ADD may be called inline because

PLSQL\_OPTIMIZE\_LEVEL is set as 2.

c. The local function F\_ADD would be called inline because PRAGMA INLINE

marks it for inline.

d. Inlining cannot be done for locally declared subprograms.

1. Suggest the tuning considerations in the following PL/SQL block:

DECLARE

L\_SUM NATURALN := 0;

L\_ID VARCHAR2(10);

BEGIN

L\_ID := 256;

L\_SUM := L\_ID \* 1.5;

END;