DATABASE MANAGEMENT SYSTEM

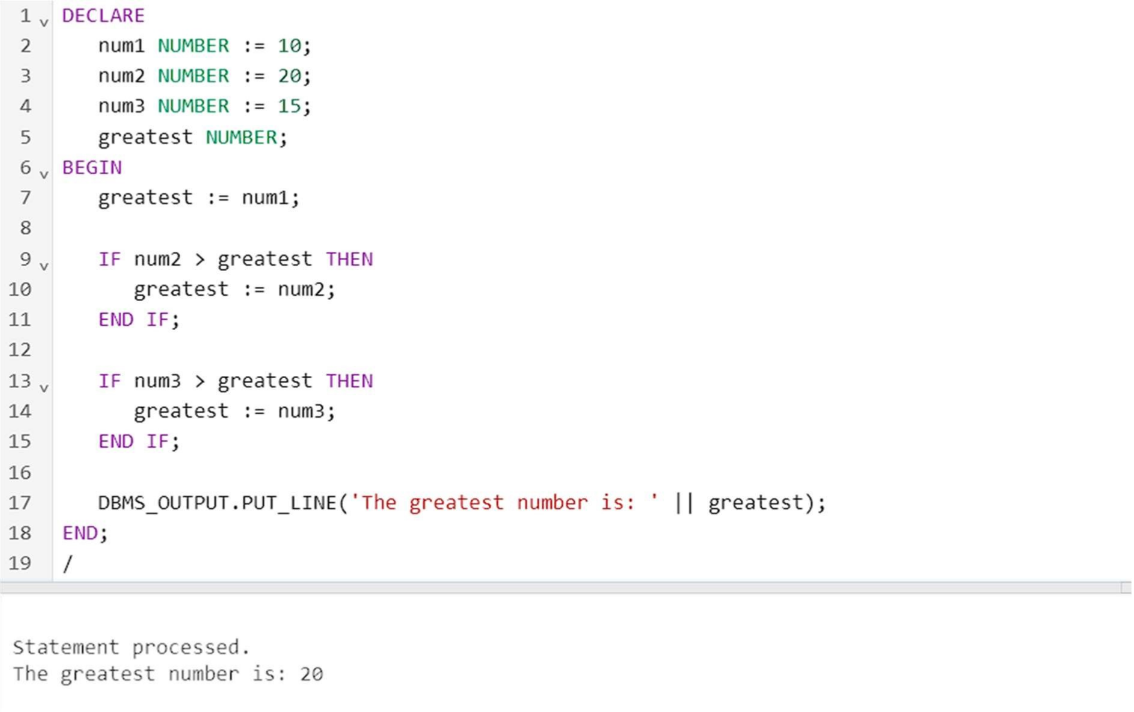
ASSIGNMENT 8

Submitted by:

Prabhnoor Singh

1021150593NC3

1) WAP to find the greatest of three numbers.



```
DECLARE
   num1 NUMBER := 10;
   num2 NUMBER := 20;
   num3 NUMBER := 15;
   greatest NUMBER;
BEGIN
   greatest := num1;

   IF num2 > greatest THEN
      greatest := num2;
   END IF;

   IF num3 > greatest THEN
      greatest := num3;
   END IF;

   DBMS_OUTPUT.PUT_LINE('The greatest number is: ' || greatest);
END;
/
```

```
Statement processed.
The greatest number is: 20
```

2) WAP to find the grade. Consider the following:

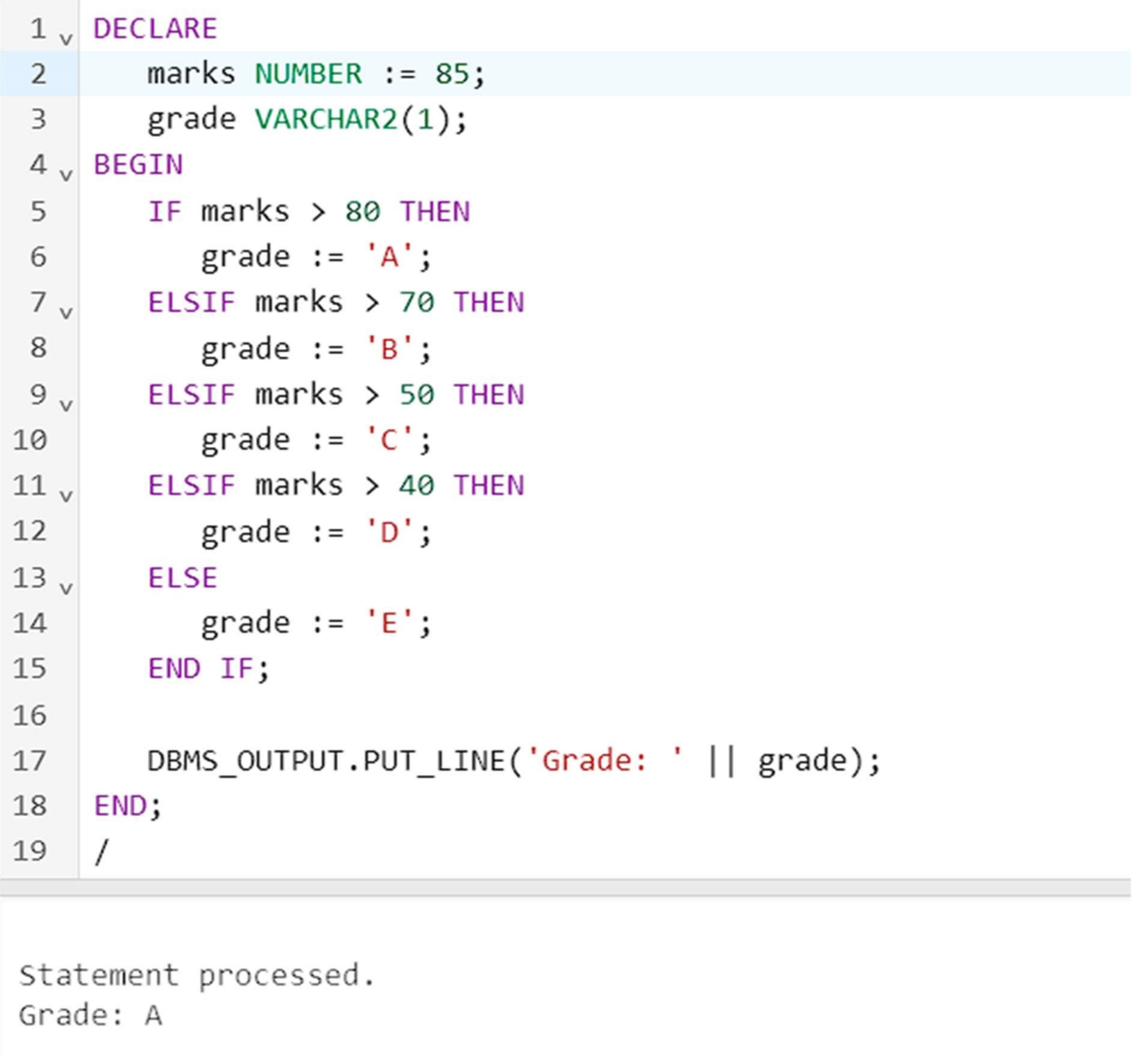
Marks > 80 A grade

Marks >70 B grade

Marks >50 C grade

Marks > 40 D grade

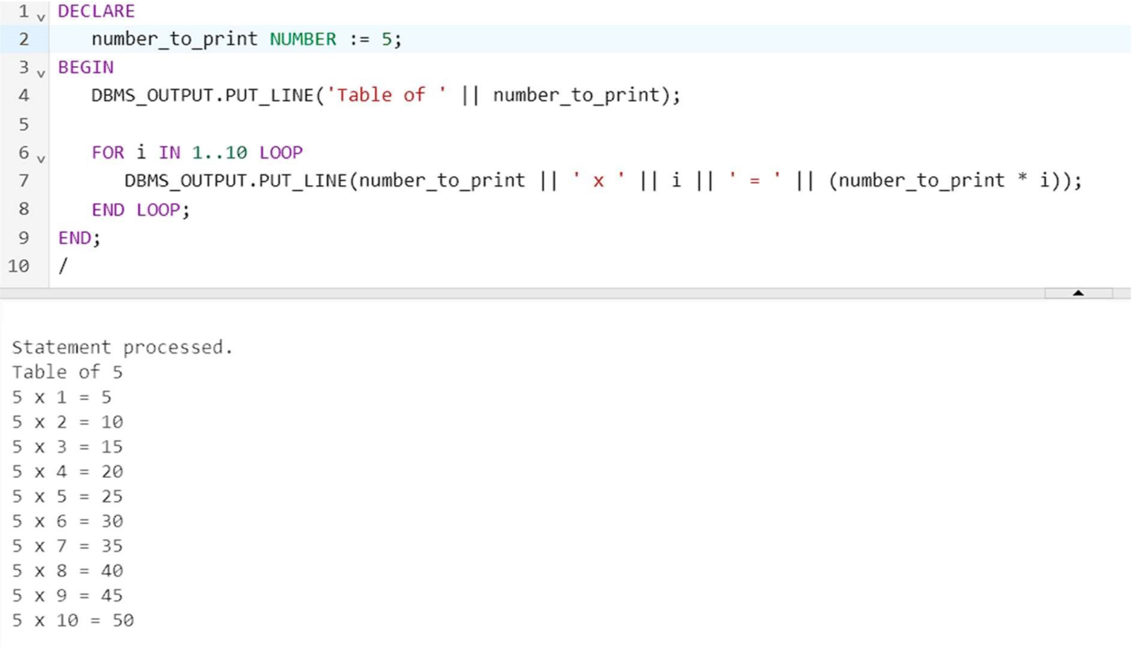
Marks < 40 E grade



```
DECLARE
   marks NUMBER := 85;
   grade VARCHAR2(1);
BEGIN
   IF marks > 80 THEN
      grade := 'A';
   ELSIF marks > 70 THEN
      grade := 'B';
   ELSIF marks > 50 THEN
      grade := 'C';
   ELSIF marks > 40 THEN
      grade := 'D';
   ELSE
      grade := 'E';
   END IF;

   DBMS_OUTPUT.PUT_LINE('Grade: ' || grade);
END;
/
```

```
Statement processed.
Grade: A
```

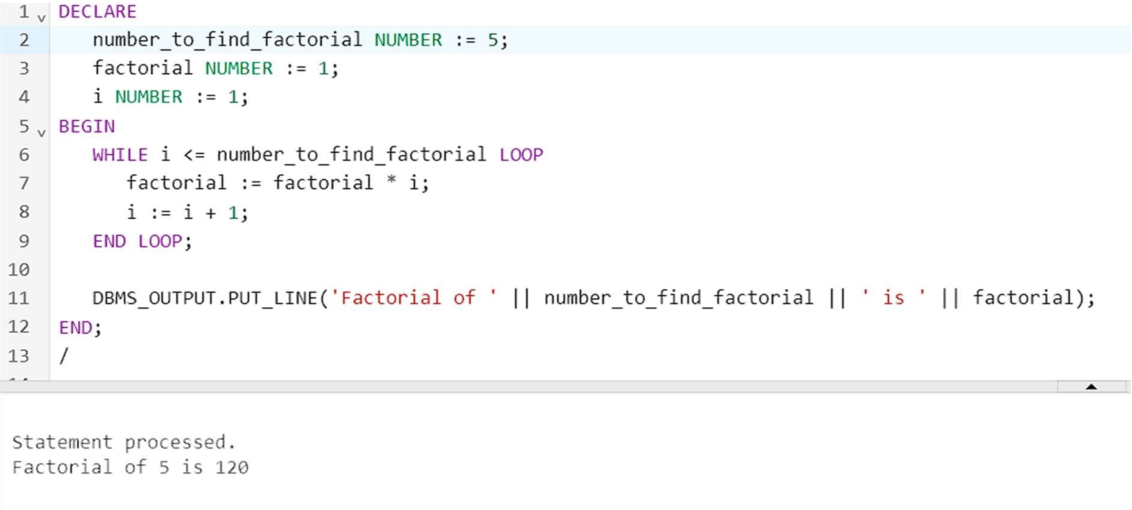
3) WAP to print the table of a given number.(use for loop)



```
DECLARE
   number_to_print NUMBER := 5;
BEGIN
   DBMS_OUTPUT.PUT_LINE('Table of ' || number_to_print);

   FOR i IN 1..10 LOOP
      DBMS_OUTPUT.PUT_LINE(number_to_print || ' x ' || i || ' = ' || (number_to_print * i));
   END LOOP;
END;
/
```

```
Statement processed.
Table of 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

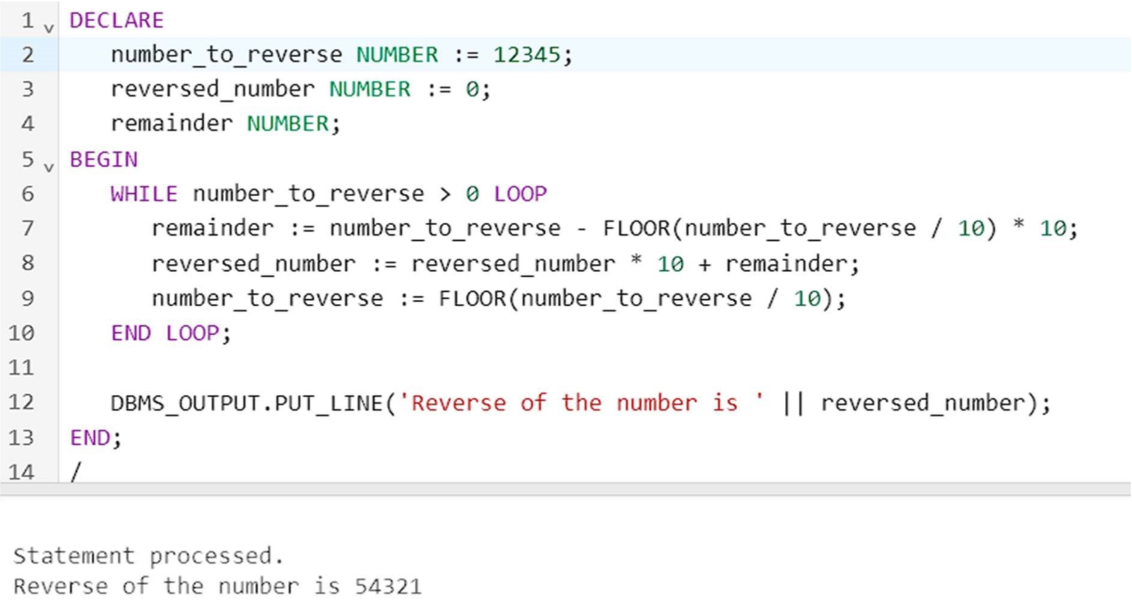
4) WAP to find out the factorial of a given number.(use while loop)



```
DECLARE
   number_to_find_factorial NUMBER := 5;
   factorial NUMBER := 1;
   i NUMBER := 1;
BEGIN
   WHILE i <= number_to_find_factorial LOOP
      factorial := factorial * i;
      i := i + 1;
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Factorial of ' || number_to_find_factorial || ' is ' || factorial);
END;
/
```

```
Statement processed.
Factorial of 5 is 120
```

5) WAP to find the reverse of a number(use exit when stement)



```
DECLARE
   number_to_reverse NUMBER := 12345;
   reversed_number NUMBER := 0;
   remainder NUMBER;
BEGIN
   WHILE number_to_reverse > 0 LOOP
      remainder := number_to_reverse - FLOOR(number_to_reverse / 10) * 10;
      reversed_number := reversed_number * 10 + remainder;
      number_to_reverse := FLOOR(number_to_reverse / 10);
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Reverse of the number is ' || reversed_number);
END;
/
```

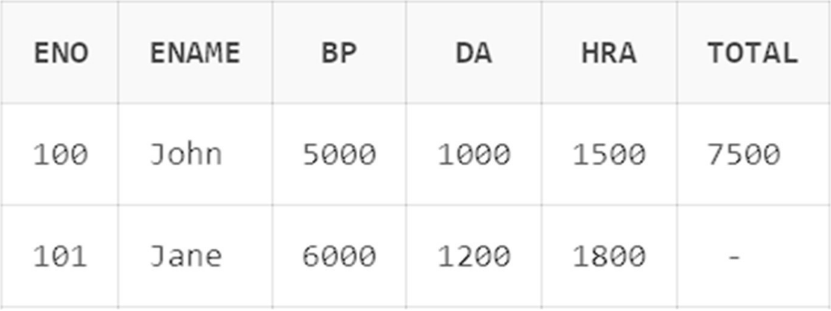
```
Statement processed.
Reverse of the number is 54321
```

6) PL/SQL block to update total sal for empno 100.

Eno,ename, bp,da,hra,total.

```
CREATE TABLE employee (
eno NUMBER,    ename
VARCHAR2(50),    bp
NUMBER,    da NUMBER,
hra NUMBER,    total
NUMBER
);


INSERT INTO employee VALUES (100, 'John', 5000, 1000, 1500, NULL); INSERT INTO
employee VALUES (101, 'Jane', 6000, 1200, 1800, NULL);
 DECLARE    v_empno
NUMBER := 100;    v_bp
NUMBER;    v_da NUMBER;
v_hra NUMBER;    v_total
NUMBER;
BEGIN
  SELECT bp, da, hra


  INTO v_bp, v_da, v_hra
  FROM employee
  WHERE eno = v_empno;
  v_total := v_bp + v_da + v_hra;
  UPDATE employee
  SET total = v_total
  WHERE eno = v_empno;


  COMMIT;
END;
/


Select * from employee;
```

Output:



ENO	ENAME	BP	DA	HRA	TOTAL
100	John	5000	1000	1500	7500
101	Jane	6000	1200	1800	-

7) PL/SQL block to calculate fine for rno 100

Rno, bookno, doi, dor, fine

Fine is rs 1 if days<7

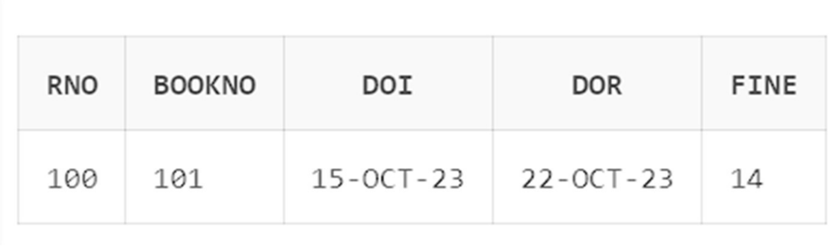
Fine is rs 2 if days<14 and >7

Fine is rs 3 if days>14

Amount mentioned is for each day.

```
CREATE TABLE library_records (
rno NUMBER,    bookno NUMBER,
doi DATE,    dor DATE,    fine
NUMBER
);


INSERT INTO library_records VALUES (100, 101, TO_DATE('2023-10-15', 'YYYY-MM-
DD'), TO_DATE('2023-10-22', 'YYYY-MM-DD'), NULL);
 DECLARE    v_rno
NUMBER := 100;    v_doi
DATE;    v_dor DATE;
v_fine NUMBER;
BEGIN
   SELECT doi, dor
   INTO v_doi, v_dor
   FROM library_records    WHERE
rno = v_rno;    v_fine := v_dor -
v_doi;    IF v_fine < 7 THEN
v_fine := v_fine * 1;
   ELSIF v_fine >= 7 AND v_fine < 14 THEN
v_fine := v_fine * 2;    ELSE
      v_fine := v_fine * 3;
   END IF;


   UPDATE library_records
   SET fine = v_fine
   WHERE rno = v_rno;


   COMMIT;




END;
/
Select * from library_records;
```

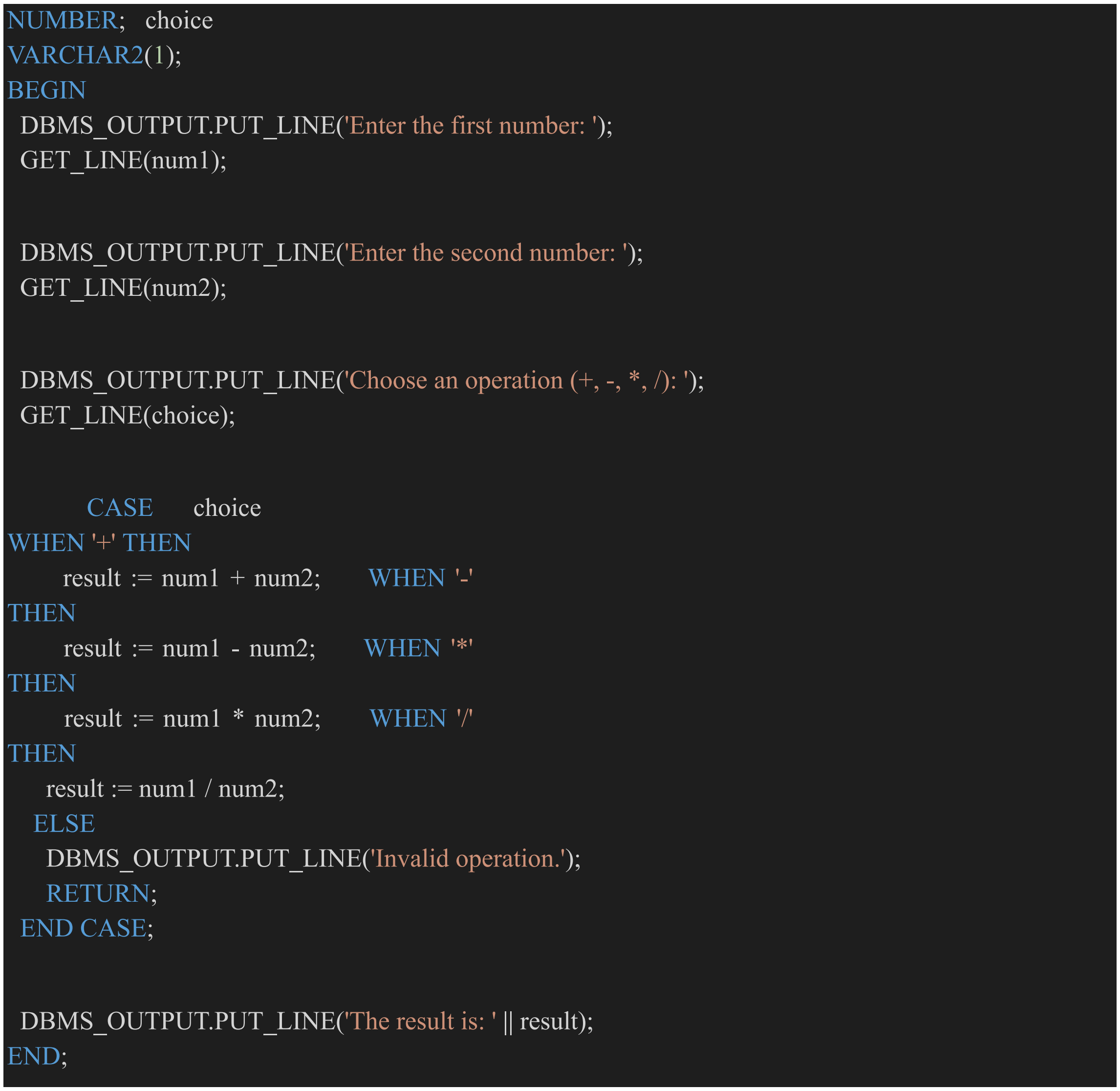
Output:



RNO	BOOKNO	DOI	DOR	FINE
100	101	15-OCT-23	22-OCT-23	14

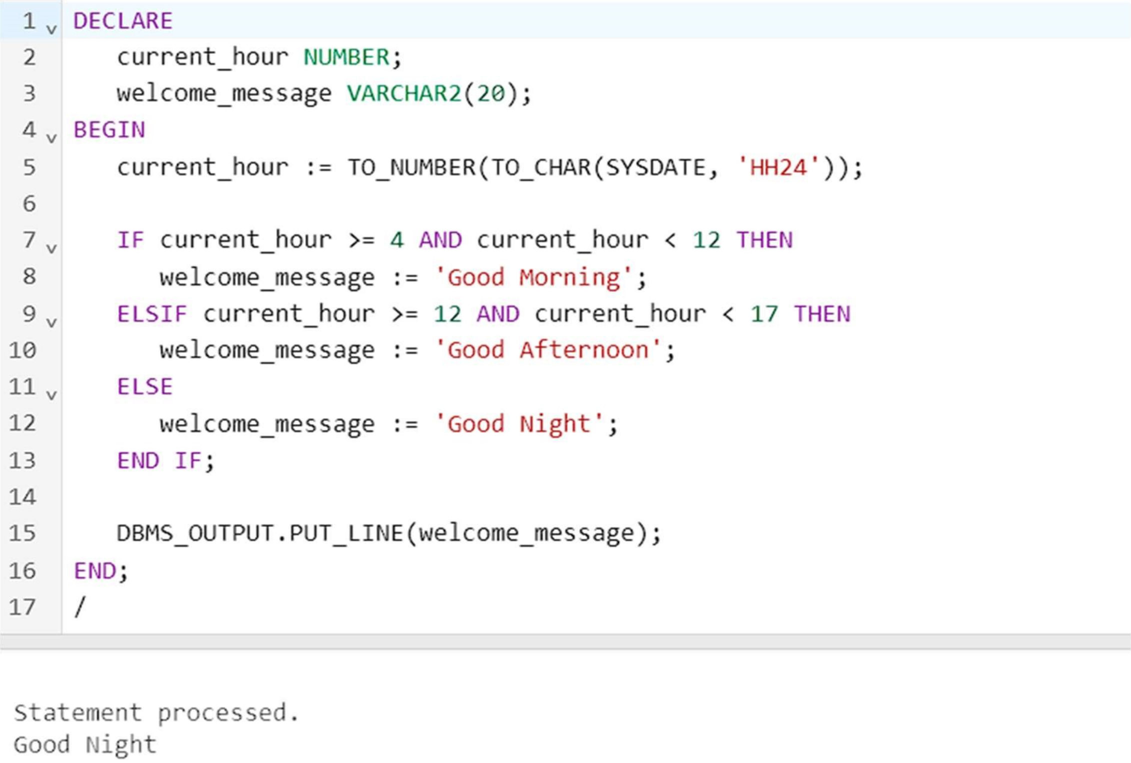
8) PL/SQL block that performs addition (+), subtraction (-), multiplication (*) and division (/) of two numbers as choice by the user.

```
DECLARE   num1
NUMBER;   num2
NUMBER;   result
```



```
NUMBER;   choice
VARCHAR2(1);
BEGIN
  DBMS_OUTPUT.PUT_LINE('Enter the first number: ');
  GET_LINE(num1);


  DBMS_OUTPUT.PUT_LINE('Enter the second number: ');
  GET_LINE(num2);


  DBMS_OUTPUT.PUT_LINE('Choose an operation (+, -, *, /): ');
  GET_LINE(choice);


            CASE      choice
WHEN '+' THEN
         result := num1 + num2;       WHEN '-'
THEN
         result := num1 - num2;       WHEN '*'
THEN
         result := num1 * num2;       WHEN '/'
THEN
      result := num1 / num2;
    ELSE
      DBMS_OUTPUT.PUT_LINE('Invalid operation.');
      RETURN;
  END CASE;


  DBMS_OUTPUT.PUT_LINE('The result is: ' || result);
END;
```

9) Pl/SQL block to display welcome message like good morning, good afternoon, good night depending on system time.



```sql
DECLARE
   current_hour NUMBER;
   welcome_message VARCHAR2(20);
BEGIN
   current_hour := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24'));

   IF current_hour >= 4 AND current_hour < 12 THEN
      welcome_message := 'Good Morning';
   ELSIF current_hour >= 12 AND current_hour < 17 THEN
      welcome_message := 'Good Afternoon';
   ELSE
      welcome_message := 'Good Night';
   END IF;

   DBMS_OUTPUT.PUT_LINE(welcome_message);
END;
/
```

```
Statement processed.
Good Night
```