

problem 1

Start SQL*Plus, log onto the database, and set up the PL/SQL output buffer using the SET SERVEROUTPUT ON command (Note: if the SET SERVEROUTPUT ON command is not used, the output will not be displayed on the screen).

```
SQL> SET SERVEROUTPUT ON;
SQL> michael lundquist 00737340
```

problem 2

Write a PL/SQL program to display your name.

```
begin
  dbms_output.put_line('MICHAEL LUNDQUIST');
end;
/
```

```
SQL> begin
  dbms_output.put_line('MICHAEL LUNDQUIST');
end;
/ 2      3      4
MICHAEL LUNDQUIST

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340
```

problem 3

Write a PL/SQL program that declares two variables (variable1 and variable2), sets their values to zero, and performs the following calculation: variable2=variable1+1 and displays the value of variable2.

```
declare
  variable1 NUMBER(2) := 0;
  variable2 NUMBER(2) := 0;
begin
  variable2 := variable1 + 1;
  dbms_output.put_line('the value of var2 is now ' || variable2);
end;
/
```

![[problem3](../pics/problem 3.png)]

Problem 4

Write a PL/SQL program to display the current date.

```
begin
  dbms_output.put_line('today is ' || sysdate);
end;
/
```

```
SQL> begin
  dbms_output.put_line('today is ' || sysdate);
end;
/
 2      3      4 today is 27-MAR-19

PL/SQL procedure successfully completed.

SQL> michael lundquist 00737340
```

problem 5

Write a PL/SQL program to declare a variable V_NUMBER, assign a value to it (any number), and display its value if the number is positive.

```
declare
V_NUMBER NUMBER(10) := 46290;
begin
  IF V_NUMBER >= 0 THEN
    dbms_output.put_line('V_NUMBER IS ' || V_NUMBER);
  END IF;
end;
/
```

```
SQL> declare
V_NUMBER NUMBER(10) := 46290;
begin
  IF V_NUMBER >= 0 THEN
    dbms_output.put_line('V_NUMBER IS ' || V_NUMBER);
  END IF;
end;
/ 2      3      4      5      6      7      8
V_NUMBER IS 46290

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340
```

problem 6

Write a PL/SQL program to declare a variable V_NUMBER, assign a value to it (any number), and test if it is a positive number or a negative number. Display the result.

```

declare
  V_NUMBER NUMBER(10) := 46290;
begin
  IF V_NUMBER > 0 OR V_NUMBER < 0 THEN
    dbms_output.put_line('V_NUMBER IS ' || V_NUMBER);
  END IF;
end;
/

```

```

SQL> declare
V_NUMBER NUMBER(10) := 46290;
begin
  IF V_NUMBER > 0 OR V_NUMBER < 0 THEN
    dbms_output.put_line('V_NUMBER IS ' || V_NUMBER);
  END IF;
end;
/ 2      3      4      5      6      7      8
V_NUMBER IS 46290

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340

```

problem 7

Write a PL/SQL program to test whether today is Friday or not. Display the result.

```

BEGIN
  IF to_char (SYSDATE, 'D') = '6' THEN
    dbms_output.put_line('TODAY IS FRIDAY');
  ELSE
    dbms_output.put_line('TODAY IS NOT FRIDAY');
  END IF;
END;
/

```

```

SQL> BEGIN
  IF to_char (SYSDATE, 'D') = '6' THEN
    dbms_output.put_line('TODAY IS FRIDAY');
  ELSE
    dbms_output.put_line('TODAY IS NOT FRIDAY');
  END IF;
END;
/ 2      3      4      5      6      7      8
TODAY IS NOT FRIDAY

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340

```

problem 8

Write a PL/SQL program to test if your date of birth falls on Friday, Saturday, or Sunday. Display the result if it's Friday, Saturday, or Sunday, otherwise print "It falls on a week-day".

```

declare
  lv_birthday_date DATE := '07-jul-1993';
BEGIN
  IF to_char (lv_birthday_date, 'D') = '6' THEN
    dbms_output.put_line('Birthday on FRIDAY');
  ELSE
    dbms_output.put_line('birthday NOT on FRIDAY');
  END IF;
END;
/

```

```

SQL> declare
  lv_birthday_date DATE := '07-jul-1993';
BEGIN
  IF to_char (lv_birthday_date, 'D') = '6' THEN
    dbms_output.put_line('Birthday on FRIDAY');
  ELSE
    dbms_output.put_line('birthday NOT on FRIDAY');
  END IF;
END;
/ 2   3   4   5   6   7   8   9  10
birthday NOT on FRIDAY

PL/SQL procedure successfully completed.

SQL> michael lundquist 00737340

```

problem 9

Write a PL/SQL program that declares a variable V_COUNTER, assigns 1 to it, and then adds 1 to the variable 5 times. Display the value of the variable every time it changes. Use the Basic LOOP structure.

```

declare
  lv_counter_num number(2) := 0;
begin
  loop
    lv_counter_num := lv_counter_num + 1;
    exit when lv_counter_num >= 5;
  end loop;
  dbms_output.put_line('result val ' || lv_counter_num);
end;
/


```

![[problem9]]("./pics/problem 9.png")

problem 10

Write a PL/SQL program that declares a variable V_COUNTER, assigns 1 to it, and then adds 1 to the variable 5 times. Display the value of the variable every time it changes. Use the WHILE ... LOOP structure

```
declare
  V_COUNTER number(2) := 1;
begin
  WHILE V_COUNTER <= 5 LOOP
    V_COUNTER := V_COUNTER + 1;
    dbms_output.put_line('CUR VAL ' || V_COUNTER);
  end loop;
end;
/
```

problem10

problem 11

Write a PL/SQL program that declares a variable V_COUNTER, assigns 1 to it, and then adds 1 to the variable 5 times. Display the value of the variable every time it changes. Use the FOR LOOP structure.

```
begin
  FOR V_COUNTER IN 1..6 LOOP
    dbms_output.put_line('CUR VAL ' || V_COUNTER);
  end loop;
end;
/
```

```
SQL> begin
  FOR V_COUNTER IN 1..6 LOOP
    dbms_output.put_line('CUR VAL ' || V_COUNTER);
  end loop;
end;
/ 2    3    4    5    6
CUR VAL 1
CUR VAL 2
CUR VAL 3
CUR VAL 4
CUR VAL 5
CUR VAL 6

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340
```

problem 12

Write a PL/SQL program that declares a variable, assigns to it your birthday, and checks what day you were born on. Use the CASE expression. Display the result.

```

declare
  lv_birthday_date DATE := '07-jul-1993';
  LV_DAY_CHAR CHAR(1) := to_char (lv_birthday_date, 'D');
  LV_TEXTDAY_VCHAR VARCHAR(10);
BEGIN
  CASE LV_DAY_CHAR
    WHEN '1' THEN LV_TEXTDAY_VCHAR := 'SUNDAY';
    WHEN '2' THEN LV_TEXTDAY_VCHAR := 'MONDAY';
    WHEN '3' THEN LV_TEXTDAY_VCHAR := 'TUESDAY';
    WHEN '4' THEN LV_TEXTDAY_VCHAR := 'WEDNESDAY';
    WHEN '5' THEN LV_TEXTDAY_VCHAR := 'THURSDAY';
    WHEN '6' THEN LV_TEXTDAY_VCHAR := 'FRIDAY';
    WHEN '7' THEN LV_TEXTDAY_VCHAR := 'SATURDAY';
  END CASE;
  dbms_output.put_line('YOU WERE BORN ON A ' || LV_TEXTDAY_VCHAR);
END;

```

```

SQL>
declare
  lv_birthday_date DATE := '07-jul-1993';
  LV_DAY_CHAR CHAR(1) := to_char (lv_birthday_date, 'D');
  LV_TEXTDAY_VCHAR VARCHAR(10);
BEGIN
  CASE LV_DAY_CHAR
    WHEN '1' THEN LV_TEXTDAY_VCHAR := 'SUNDAY';
    WHEN '2' THEN LV_TEXTDAY_VCHAR := 'MONDAY';
    WHEN '3' THEN LV_TEXTDAY_VCHAR := 'TUESDAY';
    WHEN '4' THEN LV_TEXTDAY_VCHAR := 'WEDNESDAY';
    WHEN '5' THEN LV_TEXTDAY_VCHAR := 'THURSDAY';
    WHEN '6' THEN LV_TEXTDAY_VCHAR := 'FRIDAY';
    WHEN '7' THEN LV_TEXTDAY_VCHAR := 'SATURDAY';
  END CASE;
  dbms_output.put_line('YOU WERE BORN ON A ' || LV_TEXTDAY_VCHAR);
END;
/SQL> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
YOU WERE BORN ON A WEDNESDAY

PL/SQL procedure successfully completed.

SQL> MICHAEL LUNDQUIST 00737340

```