1.Write a program to print the following format
      WELCOME TO PL/SQL PROGRAMMING

```
BEGIN
      DBMS_OUTPUT.PUT_LINE('WELCOME TO PL/SQL PROGRAMMING');
END;
/
```

2.Write a program to print the numbers from 1 to 100

```
DECLARE
    N NUMBER(3):=1;
    V VARCHAR2(1000);
 BEGIN
    WHILE N <=1000
    LOOP
            V:=V||''||N;
            N:=N+1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(V);
 END;
/
```

3.write a program to print the even numbers from 1 to 100

```
DECLARE
    N NUMBER(3):=0;
 BEGIN
    WHILE N <=100
    LOOP
            N:=N+2;
          DBMS_OUTPUT.PUT_LINE(N);
    END LOOP;
 END;
/
```

4.Write a program to print the odd numbers from 1 to 100

```
DECLARE
    N NUMBER(3):=1;
 BEGIN
    WHILE N <=100
    LOOP
            N:=N+2;
          DBMS_OUTPUT.PUT_LINE(N);
    END LOOP;
 END;
/
```

5.write a program for multiplication table

```
DECLARE
      A NUMBER(2):=&A;
      B NUMBER(2):=1;
      C NUMBER(3);
BEGIN
      WHILE B <=10
```

```
      LOOP
            C:=A*B;
            DBMS_OUTPUT.PUT_LINE(A||'*'||B||'='||C);
            B:=B+1;
      END LOOP;
END;
/
```

6.write a program to find the sum of numbers from 1 to 100

```
DECLARE
      N NUMBER(3):=1;
      S NUMBER(4):=0;
BEGIN
      WHILE N <=100
      LOOP
            S:=S+N;
            N:=N+1;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE SUM OF 1 TO 100 IS '||S);
END;
/
```

7.Write a program to find the sum of all odd numbers from 1 to 100

```
DECLARE
      N NUMBER(3):=1;
      S NUMBER(4):=0;
BEGIN
      WHILE N <=100
      LOOP
            S:=S+N;
            N:=N+2;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE SUM OF 1 TO 100  ODD NUMBERS IS '||S);

END;
/
```

8.Write a program to find the sum of all even numbers from 1 to 100

```
DECLARE
      N NUMBER(3):=0;
      S NUMBER(4):=0;
BEGIN
      WHILE N <=100
      LOOP
            S:=S+N;
            N:=N+2;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE SUM OF 1 TO 100  EVEN NUMBERS IS '||S);

END;
/
```

9.Write a program to accept a number and find how many digits it contain

```
DECLARE
      N NUMBER(5):=&N;
      CNT NUMBER:=0;
      R NUMBER(2):=0;
BEGIN
      WHILE N !=0
      LOOP
            R:=MOD(N,10);
            CNT:=CNT+1;
            N:=TRUNC(N/10);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('NUMBER OF DIGITS OF GIVEN NUMBER IS
'||CNT);
END;
/
```

10.Write a program to accept a number and find the sum of the digits

```
DECLARE
      N NUMBER(5):=&N;
      S NUMBER:=0;
      R NUMBER(2):=0;
BEGIN
      WHILE N !=0
      LOOP
            R:=MOD(N,10);
            S:=S+R;
            N:=TRUNC(N/10);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF DIGITS OF GIVEN NUMBER IS '||S);

END;
/
```

11.Write a program to accept a number and print it in reverse order

```
DECLARE
      N NUMBER(5):=&N;
      REV NUMBER(5):=0;
      R NUMBER(5):=0;
BEGIN
      WHILE N !=0
      LOOP
            R:=MOD(N,10);
            REV:=REV*10+R;
            N:=TRUNC(N/10);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE REVERSE OF A GIVEN NUMBER IS '||REV);

END;
/
```

12.Write a program to accept a no and check whether it is Armstrong number or not

13.Write a porgram to generate all the Armstrong numbers from 1 to 1000
14.Write a program to generate all prime numbers between 1 to 100
15.Write a program to aceept a number and check whether it is prime number or not
16.Write a program to display the fibonacci series from 1 to 10
17.Write a program to aceept a number and print it in binary format
18.Write a program to accept a number and find the factorial of the number
19.Find the factorials of numbers from 1 to 10

```
DECLARE
      FACT NUMBER:=1;
      V VARCHAR2(100);
BEGIN
      FOR I IN 1..10
      LOOP
            FOR J IN 1..I
            LOOP
                  FACT:=FACT*J;
                  V:=J||'*'||V;
            END LOOP;
            DBMS_OUTPUT.PUT_LINE(RTRIM(V,'*')||'='||FACT);
            FACT:=1;
            V:=NULL;
      END LOOP;
END;
/
```

20.Write a program to aceept a number and display it in the Octal format

```
DECLARE
      N NUMBER(2):=&N;
      R NUMBER(2);
      V VARCHAR2(1000);
BEGIN
      WHILE N>0
      LOOP
            R:=MOD(N,8);
            V:=R||V;
            N:=TRUNC(N/8);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('OCTAL OF A GIVEN NUMBER IS '||V);
END;
/
```

21.Write a program to accept a number and print the multiplication tables upto soo

```
DECLARE
      N NUMBER(2):=&N;
      M NUMBER;
BEGIN
      FOR I IN N..N+5
      LOOP
            FOR J IN 1..10
```

```
            LOOP
            M:=I*J;
            DBMS_OUTPUT.PUT_LINE(I||'*'||J||'='||M);
            END LOOP;
      DBMS_OUTPUT.PUT_LINE('**********************');
      END LOOP;
END;
/
```

22.Write a program to accept the temp in Centigrade and convert it into Fahrenheit(c=F-32/1.8)

```
DECLARE
      C NUMBER:=&C;
      F NUMBER;
BEGIN
      F:=C*1.8+32;
      DBMS_OUTPUT.PUT_LINE('THE FARENHETT OF GIVEN OC IS '||F);
END;
/
```

23.Write a program to calculate the area of a triangle by accepting the 3 sides
(s=(a+b+c)/2 area=sqrt(s*(s-a)*(s-b)*(s-c)))

```
DECLARE
      S NUMBER;
      A NUMBER:=&A;
      B NUMBER:=&B;
      C NUMBER:=&C;
      AREA NUMBER(7,2);
BEGIN
      S:=(A+B+C)/2;
      AREA:=SQRT(S*(S-A)*(S-B)*(S-C));
      DBMS_OUTPUT.PUT_LINE('THE AREA OF TRIANGLE IS '||AREA);
END;
/
```

24.Write a program to calculate the area of a circle by accepting the radius and unit of measure Area=PI*r2

```
DECLARE
      R NUMBER:=&R;
      AREA NUMBER(7,2);
BEGIN
      AREA:=(22/7)*R*R;
      DBMS_OUTPUT.PUT_LINE('THE AREA OF CIRCLE IS '||AREA);
END;
/
```

25.Write a program to calculate the perimeter of a circle(perimeter=2*PI*r)

```
DECLARE
      R NUMBER:=&R;
      PERIMETER NUMBER(7,2);
BEGIN
```

```
      PERIMETER:=2*(22/7)*R;
      DBMS_OUTPUT.PUT_LINE('THE PERIMETER OF CIRCLE IS '||PERIMETER);
END;
/
```

26.Write a program to accept the 3 sides of the triangle and display the type of triangle

```
DECLARE
      A NUMBER(4,2):=&A;
      B NUMBER(4,2):=&B;
      C NUMBER(4,2):=&C;
      PERIMETER NUMBER(7,2);
BEGIN
      IF (A=B AND B=C AND C=A) THEN
      DBMS_OUTPUT.PUT_LINE('EQUILATERAL TRIANGLE');
      ELSIF A=B OR A=C OR C=B THEN
      DBMS_OUTPUT.PUT_LINE('ISOSOCELESS TRIANGLE');
      ELSE
        DBMS_OUTPUT.PUT_LINE('SCALEN TRIANGLE');
      END IF;
END;
/
```

27.Write a program accept the value of A,B&C display which is greater

```
DECLARE
      A NUMBER(4,2):=&A;
      B NUMBER(4,2):=&B;
      C NUMBER(4,2):=&C;
BEGIN
      IF (A>B AND A>C) THEN
      DBMS_OUTPUT.PUT_LINE('A IS GREATER '||''||A);
      ELSIF B>C THEN
      DBMS_OUTPUT.PUT_LINE('B IS GREATE '||''||B);
      ELSE
        DBMS_OUTPUT.PUT_LINE('C IS GREATER '||''||C);
      END IF;
END;
/
```

28.Write a program accept a string and check whether it is palindrome or not

```
DECLARE
      S VARCHAR2(10):='&S';
      L VARCHAR2(20);
      TEMP VARCHAR2(10);
BEGIN
      FOR I IN REVERSE 1..LENGTH(S)
      LOOP
      L:=SUBSTR(S,I,1);
      TEMP:=TEMP||''||L;
      END LOOP;
      IF TEMP=S THEN
      DBMS_OUTPUT.PUT_LINE(TEMP ||''||' IS PALINDROME');
      ELSE
```

```
      DBMS_OUTPUT.PUT_LINE(TEMP ||''||' IS NOT PALINDROME');
      END IF;
END;
/
```

29.Write a program aceepts the value of A,B and swap the nos and print the values

```
DECLARE
      A NUMBER(2):=&A;
      B NUMBER(2):=&B;
      FLAG NUMBER(2);
BEGIN
      FLAG:=A;
      A:=B;
      B:=FLAG;
      DBMS_OUTPUT.PUT_LINE('A '||'= '||A||' AND '||''||'B '||'= '||B);
END;
/
```

30.Write a program to accept the values of A , B and swap the numbers and print the values
without using third variable

```
DECLARE
      A NUMBER(2):=&A;
      B NUMBER(2):=&B;
      FLAG NUMBER(2);
BEGIN
      FLAG:=A;
      A:=B;
      B:=FLAG;
      DBMS_OUTPUT.PUT_LINE('A '||'= '||A||' AND '||''||'B '||'= '||B);
END;
/
```

31.Write a program to accept the side of a square and calculate the area area =a2

```
DECLARE
      A NUMBER:=&A;
      AREA NUMBER(5);
BEGIN
      AREA:=A*A;
      DBMS_OUTPUT.PUT_LINE('AREA OF A SQUARE IS '||''||AREA);
END;
/
```

32.Write a program to accept principle amount ,rate,time calculate the simple interest si=(p*t*r)/100

```
DECLARE
      P NUMBER(6,2):=&P;
      R NUMBER(6,2):=&R;
      T NUMBER(6,2):=&T;
      SI NUMBER(6,2);
BEGIN
      SI:=(P*R*T)/100;
```

```
      DBMS_OUTPUT.PUT_LINE('SIMPLE INTEREST IS '||''||SI);
END;
/
```

33.Erite a program to aceept the principle amount,rate,time and find the compound interest
ci=p*(1+r/100)n

```
DECLARE
      P NUMBER(6,2):=&P;
      R NUMBER(6,2):=&R;
      T NUMBER(6,2):=&T;
      CI NUMBER(6,2);
BEGIN
      CI:=P*POWER(1+(R/100),T);
      DBMS_OUTPUT.PUT_LINE('COMPOUND INTEREST IS '||CI);
END;
/
```

34.WAP to calculate the sum of 1!+2!+......+n!

```
DECLARE
      N NUMBER:=&N;
      S NUMBER:=0;
      F NUMBER:=1;
BEGIN
      FOR I IN 1..N
      LOOP
            FOR J IN 1..I
            LOOP
                  F:=F*J;
            END LOOP;
      S:=S+F;
      F:=1;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF FACT IS '||S);
END;
/
```

35.WAP to calculate the sum of 1+1/2+1/3+......+1/n

```
DECLARE
      N NUMBER:=&N;
      A NUMBER;
      S NUMBER(6,2):=0;
BEGIN
      FOR I IN 1..N
      LOOP
            A:=1/I;
            S:=S+A;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF NO ARE '||S);
END;
/
```

36.WAP to calculate the sum of 1/1!+1/2!+.....+1/n!

```
DECLARE
      N NUMBER:=&N;
      S NUMBER(6,2):=0;
      F NUMBER:=1;
BEGIN
      FOR I IN 1..N
      LOOP
            FOR J IN 1..I
            LOOP
                  F:=F*J;
            END LOOP;
            S:=S+(1/F);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM IS '||S);
END;
/
```

37.WAP to calculate the sum of 1/1!+2/2!+......+n/n!

```
DECLARE
      N NUMBER(4):=&N;
      S NUMBER(6,2):=0;
      F NUMBER(4):=1;
BEGIN
      FOR I IN 1..N
      LOOP
            FOR J IN 1..I
            LOOP
                  F:=F*J;
            END LOOP;
            S:=S+(I/F);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF FACT IS '||S);
END;
/
```

38.Write a program to display the months between two dates of a year

```
DECLARE
      D DATE:='&D';
      D1 DATE:='&D1';
BEGIN
      WHILE D < D1
      LOOP
            DBMS_OUTPUT.PUT_LINE(TO_CHAR(D,'MONTH'));
            D:=ADD_MONTHS(D,1);
      END LOOP;
END;
/
```

39.Write a program to accept the date and print the weekdays from the given date

```
DECLARE
      D DATE:='&D';
      WD DATE;
BEGIN
```

```
      WD:=D+6;
      WHILE D <= WD
      LOOP
            DBMS_OUTPUT.PUT_LINE(TO_CHAR(D,'DAY'));
            D:=D+1;
      END LOOP;
END;
/
```

40.WAP to accept the date and print the weekdays from the given date along with date format

```
DECLARE
      D DATE:='&D';
      WD DATE;
BEGIN
      WD:=D+6;
      WHILE D <= WD
      LOOP
            DBMS_OUTPUT.PUT_LINE(TO_CHAR(D,'DAY')||D);
            D:=D+1;
      END LOOP;
END;
/
```

41.Writa a program to accept a year and check whether it is leap year or not

```
DECLARE
      Y NUMBER:=&Y;
      R NUMBER;
BEGIN
      IF MOD(Y,4)=0 AND MOD(Y,100)!=0 OR MOD(Y,400)=0
      THEN
            DBMS_OUTPUT.PUT_LINE(Y ||' IS A LEAP YEAR');
      ELSE
            DBMS_OUTPUT.PUT_LINE(Y ||' IS NOT A LEAP YEAR');
      END IF;
END;
/
```

42.Write a program to accept a year and display all sundays along with the date

```
DECLARE
      Y NUMBER(4):=&YYYY;
      A DATE;
      B DATE;
      I NUMBER(2):=1;
BEGIN
      A:=TO_DATE('01-JAN-'||Y,'DD-MON-YYYY');
      B:=LAST_DAY(ADD_MONTHS(A,11));
      WHILE A <= B
      LOOP
            IF TO_CHAR(A,'D')=1 THEN
            DBMS_OUTPUT.PUT_LINE(LPAD(I,2,'0')||'-
'||UPPER(TO_CHAR(A,'DAY'))||A);
```

```
            I:=I+1;
            END IF;
            A:=A+1;
      END LOOP;
END;
/
```

43.WAP to accept a string and count how many vowels present in the string

```
DECLARE
      V VARCHAR2(300):='&V';
      CNT NUMBER(5):=0;
      C CHAR;
BEGIN
      FOR I IN 1..LENGTH(V)
      LOOP
            C:=SUBSTR(V,I,1);
            IF C IN ('A','E','I','O','U') THEN
            CNT:=CNT+1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('NO OF VOWELS PRESENT = '||CNT);
END;
/
```

44.Write a program to accept a year and check whether it is leap year or not . If it is
leap year then display how many sundays present in that year

```
DECLARE
      D DATE:='&YEAR';
      Y VARCHAR2(20);
      CNT NUMBER(5):=0;
      V VARCHAR2(20);
BEGIN
      Y:=TO_CHAR(D,'YYYY');
      D:=TO_DATE('01-JAN-'||Y);
      IF MOD(Y,4)=0 AND MOD(Y,100)!=0 OR MOD(Y,400)=0 THEN
            FOR I IN 1..366
            LOOP
                  V:=TO_CHAR(D,'D');
                  IF V=1 THEN
                        CNT:=CNT+1;
                  END IF;
                  D:=D+1;
                  DBMS_OUTPUT.PUT_LINE('NO OF VOWELS PRESENT = '||CNT);
            END LOOP;

END;
/
```

45.Write a program to accept a char and check it is vowel or consonant

```
DECLARE
      C CHAR:='&C';
BEGIN
```

```
      IF C='A' OR C='E' OR C='I' OR C='O' OR C='U' THEN
      DBMS_OUTPUT.PUT_LINE('VOWEL');
      ELSE
      DBMS_OUTPUT.PUT_LINE('CONSONANT');
      END IF;
END;
/
```

46.WAP to accept A,B,C & D check whether it is Ramanujan number or not

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      C NUMBER:=&C;
      D NUMBER:=&D;
BEGIN
      IF
        POWER(A,3)+POWER(B,3)=POWER(C,3)+POWER(D,3) THEN
      DBMS_OUTPUT.PUT_LINE(A||CHR(179)||'+'||B||CHR(179)||'='||C||CHR(1
79)||'+'||D||CHR(179));
      ELSE
      DBMS_OUTPUT.PUT_LINE(A||CHR(179)||'+'||B||CHR(179)||'!='||C||CHR(
179)||'+'||D||CHR(179));
      END IF;
END;
/
```

47.WAP to accept the CMR & LMR & find out the total bill amount
i)0-100 units Rs.50 per unit ii)101-200n units Rs.o.25 per unit
iii)>200 units Rs.1.25 per unit

```
DECLARE
      LMR NUMBER(5):=&LMR;
      CMR NUMBER(5):=&CMR;
      TOT NUMBER(5):=0;
      BILL NUMBER(7,2):=0;
BEGIN
      TOT:=CMR-LMR;
      IF TOT <= 100 THEN
            BILL:=TOT*.50;
      ELSIF TOT > 100 AND TOT <= 200 THEN
            BILL:=(100*.50)+((TOT-100)*.75);
      ELSE
            BILL:=(100*.50)+(100*.75)+(TOT-200)*1.25;
      END IF;
      DBMS_OUTPUT.PUT_LINE('TOTAL UNIT CONSUMED   '||TOT);
      DBMS_OUTPUT.PUT_LINE('TOTAL BILL AMOUNT '||BILL);
END;
/
```

48.WAP or accept marks of 3 subject as i/p and calculate the total
marks and division of a student
i) If totmark>=60 then division is First
ii) If totmark <60 and totmark>=50 then division is second
iii) If totmark< 50 and >=35 then division is third
iv) If totmark< 35 then fail

```
DECLARE
      M1 NUMBER(2):=&M1;
      M2 NUMBER(2):=&M2;
      M3 NUMBER(2):=&M3;
      TOTMARK NUMBER(5,2);
      AVE NUMBER(5,2):=0;
BEGIN
      TOTMARK:=M1+M2+M3;
      AVE:=TOTMARK/3;
      IF AVE>=60 THEN
      DBMS_OUTPUT.PUT_LINE('THE DIVISION IS FIRST '||AVE);
      ELSIF AVE<60 AND AVE>=50 THEN
      DBMS_OUTPUT.PUT_LINE('THE DIVISION IS SECOND '||AVE);
      ELSIF AVE<50 AND AVE>=35 THEN
      DBMS_OUTPUT.PUT_LINE('THE DIVISION IS THIRD '||AVE);
      ELSE
      DBMS_OUTPUT.PUT_LINE('FAIL '||AVE);
      END IF;
END;
/
```

49.WAP to accept a number and print its multiplication table horinzontally

```
DECLARE
      J NUMBER:=&J;
      V VARCHAR2(1000);
      K NUMBER(3);
BEGIN
      FOR I IN 1..10
      LOOP
            K:=J*I;
            V:=V||J||'*'||I||'='||K||' ';
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

50.WAP to accept a string and print it in reverse order

```
DECLARE
      STR VARCHAR2(100):='&sTR';
      STR1 VARCHAR2(100);
      N NUMBER(5);
      L VARCHAR2(20);
BEGIN
      N:=LENGTH(STR);
      FOR I IN 1..N
      LOOP
            L:=SUBSTR(STR,I,1);
            STR1:=L||STR1;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(STR1);
END;
/
```

51.Write a program to accept a number and find out the sum of first and

last digits

```
DECLARE
      A NUMBER(4):=&A;
      B NUMBER(5):=0;
      C NUMBER(5):=0;
      S NUMBER(5);
BEGIN
      IF A>9 THEN
      C:=SUBSTR(A,1,1);
            B:=SUBSTR(A,LENGTH(A),1);
      S:=B+C;
      ELSE
            S:=A;
      END IF;
      DBMS_OUTPUT.PUT_LINE('SUM OF FIRST AND LAST DIGIT IS '||S);
END;
/
```

52.WAP to accept the basic salary and find out the ta,da,hra,lic and gs
i)ta 20% of basic, da 10% of basic, hra 30% of basic, lic 5% of basic

```
DECLARE
      BS NUMBER(6,2):=&BS;
      TA NUMBER(6,2);
      DA NUMBER(6,2);
      HRA NUMBER(6,2);
      GS NUMBER(6,2);
      LIC NUMBER(6,2);
      NS NUMBER(8,2);
BEGIN
      TA:=BS*(20/100);
      HRA:=BS*(30/100);
      DA:=BS*(10/100);
      LIC:=BS*(5/100);
      GS:=TA+HRA+DA;
      NS:=GS-LIC;
      DBMS_OUTPUT.PUT_LINE('EMPLOYEE BS IS '||BS);
      DBMS_OUTPUT.PUT_LINE('GROSS SALARY IS '||GS);
      DBMS_OUTPUT.PUT_LINE('NET SALARY IS '||NS);
END;
/
```

53.WAP to accept the length and breadth of a rectangle and find out the perimeter

```
DECLARE
      L NUMBER(4,2):=&L;
      B NUMBER(4,2):=&B;
      A NUMBER(4,2);
BEGIN
      A:=2*(L+B);
      DBMS_OUTPUT.PUT_LINE('THE PERIMETER OF RECTANGLE IS '||A);
END;
/
```

54.WAP to accept the cost price and selling price of an item and find

the loss or profit

```
DECLARE
      CP NUMBER(25,2):=&CP;
      SP NUMBER(25,2):=&SP;
      AMT NUMBER(7,2);
BEGIN
      IF CP < SP THEN
            AMT:=SP-CP;
            DBMS_OUTPUT.PUT_LINE('PROFIT IS '||AMT);
      ELSE
            AMT:=CP-SP;
            DBMS_OUTPUT.PUT_LINE('LOSS IS '||AMT);
      END IF;
END;
/
```

55.Writ a program to generate the following series

```
53 53 53 53 53
43 43 43 43
33 33 33
23 23
13
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN REVERSE 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||I||CHR(179);
            END LOOP;
            DBMS_OUTPUT.PUT_LINE(V);
            V:=NULL;
      END LOOP;
END;
/
```

56.WAP to accept a no in binary format and print it in decimal format

```
DECLARE
      N VARCHAR2(20):=&N;
      PRO NUMBER(10,4):=0;
      L VARCHAR2(10);
BEGIN
      FOR I IN 1..LENGTH(N)
      LOOP
            L:=SUBSTR(N,I,1);
            PRO:=PRO+L*POWER(2,LENGTH(N)-I);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE DECIMAL NUMBER IS '||PRO);
END;
/
```

57.WAP to accept two nos and input and find one no is raised to another one (without using any function)

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      R NUMBER:=1;
BEGIN
      FOR I IN 1..B
      LOOP
            R:=R*A;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('A RAISED POWER B IS '||R);
END;
/
```

58.WAP to accept a sentence and count the no of chars in that sentence

```
DECLARE
      STR VARCHAR2(100):='&STR';
      NO NUMBER(5):=0;
      I NUMBER;
BEGIN
      I:=INSTR(STR,'.');
      DBMS_OUTPUT.PUT_LINE('NO OF CHAR IS '||I);
END;
/
```

59.WAP to accept two strings and display the large one among those

```
DECLARE
      STR1 VARCHAR2(100):='&STR1';
      STR2 VARCHAR2(100):='&STR2';
BEGIN
      IF LENGTH(STR1) > LENGTH(STR2) THEN
            DBMS_OUTPUT.PUT_LINE(STR1 ||' IS GREATER');
      ELSIF LENGTH(STR1) < LENGTH(STR2) THEN
            DBMS_OUTPUT.PUT_LINE(STR2 ||' IS GREATER');
      ELSE
            DBMS_OUTPUT.PUT_LINE('BOTH STRINGS ARE EQUAL');
      END IF;
END;
/
```

60.WAP to display all the nos whose sum of digits is 9 from 1 to 9999

```
DECLARE
      N NUMBER;
      M NUMBER;
      S NUMBER:=0;
BEGIN
      FOR I IN 1..999
      LOOP
            N:=I;
            WHILE N>0
            LOOP
                  M:=MOD(N,10);
                  S:=S+M;
                  N:=TRUNC(N/10);
```

```
            END LOOP;
            IF S=9 THEN
            DBMS_OUTPUT.PUT_LINE(I||' ');
            END IF;
                  S:=0;
      END LOOP;
END;
/
```

61.WAP to accept a no and find the sum in a single digit

```
DECLARE
      N NUMBER(4):=&N;
      S NUMBER(10):=0;
BEGIN
      WHILE LENGTH(N)>1
      LOOP
            FOR I IN 1..LENGTH(N)
            LOOP
                  S:=S+SUBSTR(N,I,1);
            END LOOP;
            N:=S;
            S:=0;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE SUM IN SINGLE DIGIT IS '||N);
END;
/
```

62.Ente the no of days and find out the no of years and no of days and months

```
DECLARE
      D NUMBER:=&D;
      Y NUMBER;
      M NUMBER;
BEGIN
      Y:=TRUNC(D/365);
      M:=TRUNC(MOD(D,365)/30);
      D:=MOD(MOD(D,365),30);
      DBMS_OUTPUT.PUT_LINE(Y||' YEARS '||M||' MONTHS '||D||' DAYS');
END;
/
```

63.WAP to accept the date and print all the weekdays along with the given date

```
DECLARE
      D DATE:='&D';
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..7
      LOOP
            V:=TO_CHAR(D,'DAY')||D;
            DBMS_OUTPUT.PUT_LINE(V);
            D:=D+1;
      END LOOP;
END;
```

```
/
```

64.WAP while purchasing certain items,discount of each is as follows
i) If qty purchased > 1000 discount is 20%
ii) If the qty and price per item are i/p then calculate the expenditure

```
DECLARE
      QTY NUMBER(5):=&QTY;
      UP NUMBER(6,2):=&UP;
      DIS NUMBER(6,2):=0;
      TAMT NUMBER(10,2);
      BILL NUMBER(10,2);
BEGIN
      BILL:=QTY*UP;
      IF BILL > 1000 THEN
            DIS:=BILL*20/1000;
      END IF;
      TAMT:=BILL-DIS;
      DBMS_OUTPUT.PUT_LINE('THE TOTAL AMOUNT IS '||TAMT);
END;
/
```

65.Write a program to accept a string and count the no of individual chars

```
DECLARE
      V VARCHAR2(100):='&V';
      V1 VARCHAR2(100);
      LB NUMBER;
      LA NUMBER;
      DIFF NUMBER;
      C CHAR;
      N NUMBER(5):=0;
BEGIN
      V1:=V;
      WHILE LENGTH(V1)>0
      LOOP
            C:=SUBSTR(V1,1,1);
            LB:=LENGTH(V1);
            V1:=REPLACE(V1,C);
            LA:=NVL(LENGTH(V1),0);
            DIFF:=LB-LA;
            IF ASCII(C)=32 THEN
      DBMS_OUTPUT.PUT_LINE('SPACE'||' EXISTS '||DIFF||' TIMES');
            ELSE
      DBMS_OUTPUT.PUT_LINE(C||' EXISTS '||DIFF||' TIMES');
            END IF;
            N:=N+DIFF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('TOTAL LENGTH OF THE GIVEN STRING
'||V||'='||N);
END;
/
```

66.Write a program to display all combination of 1,2,&3

```
BEGIN
      FOR I IN 1..3
      LOOP
            FOR J IN 1..3
            LOOP
                  FOR K IN 1..3
                  LOOP
                        DBMS_OUTPUT.PUT_LINE(I||J||K);
                  END LOOP;
            END LOOP;
      END LOOP;
END;
/
```

67.Write a program to find out the series 12+22+32+42+....++n2

```
DECLARE
      N NUMBER:=&N;
      A NUMBER:=1;
      B NUMBER:=2;
      C NUMBER:=0;
      D NUMBER:=0;
      S NUMBER:=0;
BEGIN
      WHILE A<=N
      LOOP
            C:=C+A*A;
            A:=A+2;
      END LOOP;
      WHILE B<=N
      LOOP
            D:=D+B*B;
            B:=B+2;
      END LOOP;
      S:=C-D;
      DBMS_OUTPUT.PUT_LINE('RESULT IS '||S);
END;
/
```

68.Write a program to accep the time in HH & MIN format and find the total senconds

```
DECLARE
      H NUMBER:=&HOUR;
      M NUMBER:=&MINUTE;
      S NUMBER(10):=0;
BEGIN
      S:=(H*60*60)+(M*60);
      DBMS_OUTPUT.PUT_LINE(H||' HOURS '||M||' MINUTES '||'IS'||S||'
SECONDS');
END;
/
```

69.WAP to accept the distance between two cities in km and convert into mts ,cm & ft

```
DECLARE
```

```
      D NUMBER:=&D;
      M NUMBER:=0;
      CM NUMBER:=0;
      FT NUMBER:=0;
BEGIN
      M:=D*1000;
      CM:=M*100;
      FT:=ROUND(CM/12.3);
      DBMS_OUTPUT.PUT_LINE('DISTANCE IN METERS  IS '||M);
      DBMS_OUTPUT.PUT_LINE('DISTANCE IN CENTIMETERS IS '||CM);
      DBMS_OUTPUT.PUT_LINE('DISTANCE IN FOOT IS '||FT);
END;
/
```

70.Write a program to find the series x+x2/2!+x3/3!+.....+xn/n!

```
DECLARE
      N NUMBER:=&N;
      X NUMBER:=&X;
      S NUMBER:=0;
      F NUMBER:=1;
BEGIN
      FOR I IN 1..N
      LOOP
            FOR J IN 1..I
            LOOP
                  F:=F*J;
            END LOOP;
            S:=ROUND(s+(POWER(X,I)/F),3);
            F:=1;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF NUMBER IS '||S);
END;
/
```

71.Write a program to accept the population of hyderabad each year the population increases
2% after 4y what is the population of hyd

```
DECLARE
      P NUMBER:=&P;
      L NUMBER;
BEGIN
      FOR J IN 1..4
      LOOP
            L:=P*2/100;
            P:=P+L;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('POPULATION OF HYDERABAD AFTER 4 YEARS IS
'||TRUNC(P));
END;
/
```

72.WAP to accept the 3 dates and display the most recently month among 3 dates

```
DECLARE
```

```
      D1 DATE:='&D1';
      D2 DATE:='&D2';
      D3 DATE:='&D3';
      M1 NUMBER;
      M2 NUMBER;
      M3 NUMBER;
BEGIN
      M1:=TO_CHAR(D1,'MM');
      M2:=TO_CHAR(D2,'MM');
      M3:=TO_CHAR(D3,'MM');
      IF M1>M2 AND M1>M3 THEN
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(D1,'MON')||' IS RECENT MONTH');
      ELSIF M2>M1 AND M2>M3 THEN
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(D2,'MON')||' IS RECENT MONTH');
      ELSE
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(D3,'MON')||' IS RECENT MONTH');
      END IF;
END;
/
```

73.Accept a string and print in the following format

```
O
OR
ORA
ORAC
ORACL
ORACLE

DECLARE
      V VARCHAR2(20):='&V';
      C VARCHAR(20);
BEGIN
      FOR I IN 1..LENGTH(V)
      LOOP
            C:=SUBSTR(V,1,I);
      DBMS_OUTPUT.PUT_LINE(C);
      END LOOP;
END;
/
```

74.Write a program to accept the annual income of the emp and find the income tax

i) If the annsal > 60000 then tax is 10% of income
ii) If the annsal > 100000 then tax is Rs 800+16% of income
iii) If the annsal > 140000 then tax is Rs 2500+25% of income

```
DECLARE
      AI NUMBER(10,2):=&ANNUALINCOME;
      TAX NUMBER(10,3):=0;
BEGIN
      IF AI BETWEEN 36000 AND 50000 THEN
            TAX:=AI*10/100;
      ELSIF AI BETWEEN 50000 AND 100000 THEN
            TAX:=800+AI*16/100;
      ELSIF AI > 100000 THEN
            TAX:=2500+AI*25/100;
      END IF;
```

```
      DBMS_OUTPUT.PUT_LINE('ANNUAL INCOME '||AI);
      DBMS_OUTPUT.PUT_LINE('TAX '||TAX);
END;
/
```

75.WAP to accept a year as i/p & find how many even number present in that year

```
DECLARE
      Y NUMBER:=&YEAR;
      A VARCHAR2(20);
      CNT NUMBER(5):=0;
BEGIN
      FOR I IN 1..LENGTH(Y)
      LOOP
            A:=SUBSTR(Y,I,1);
            IF MOD(A,2)=0 THEN
                  CNT:=CNT+1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('NUMBER OF EVEN DIGIT IS '||CNT);
END;
/
```

76.WAP to accept a year as i/p & find how many odd number present in that year

```
DECLARE
      Y NUMBER:=&YEAR;
      A VARCHAR2(20);
      CNT NUMBER(5):=0;
BEGIN
      FOR I IN 1..LENGTH(Y)
      LOOP
            A:=SUBSTR(Y,I,1);
            IF MOD(A,2)!=0 THEN
                  CNT:=CNT+1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('NUMBER OF EVEN DIGIT IS '||CNT);
END;
/
```

77.WAP to accept a number and calculate the sum of numbers in even places

```
DECLARE
      N NUMBER:=&NUMBER;
      A VARCHAR2(10);
      S NUMBER:=0;
BEGIN
      FOR I IN 1..LENGTH(N)
      LOOP
            A:=SUBSTR(N,I,1);
            IF MOD(I,2)=0 THEN
                  S:=S+A;
            END IF;
```

```
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM OF EVEN PLACE IS '||S);
END;
/
```

78.WAP to accept the emp details and calculate the bonus based on the following conditions
i) If sal < 500 then bonus is 10% sal
ii) If sal > 3500 then bonus is 12% sal
iii) If sal > 1000 then bonus is 13.5% sal

```
DECLARE
      EMPNOV NUMBER:=&EMPNOV;
      SALV NUMBER;
      B NUMBER(7,2);
BEGIN
      SELECT SAL INTO SALV FROM EMP WHERE EMPNO=EMPNOV;
      IF SALV BETWEEN 500 AND 3500 THEN
            B:=SALV*10/100;
      ELSIF SALV BETWEEN 3500 AND 10000 THEN
            B:=SALV*12/100;
      ELSIF SALV>10000 THEN
            B:=SALV*13.5/100;
      END IF;
      DBMS_OUTPUT.PUT_LINE('EMPNO  '||EMPNOV);
      DBMS_OUTPUT.PUT_LINE('SALARY  '||SALV);
      DBMS_OUTPUT.PUT_LINE('BONUS  '||B);
END;
/
```

79.WAP to accept the empno and display ename,sal,hiredate and calculate ta,da,hra,lic,gross,exp and
print all emp details. ta is 30% of sal,da is 20% of sal,hra is 15% of sal,lic is 5% of sal

```
DECLARE
   EMPNOV NUMBER:=&EMPNOV;
   ENAMEV EMP.ENAME%TYPE;
   SALV EMP.SAL%TYPE;
   HIREDATEV EMP.HIREDATE%TYPE;
   EXP NUMBER(7,2);
   TA NUMBER(7,2);
   DA NUMBER(7,2);
   HRA NUMBER(7,2);
   LIC NUMBER(7,2);
   GROSS NUMBER(7,2);
   S NUMBER:=0;
BEGIN
   SELECT ENAME,SAL,HIREDATE INTO ENAMEV,SALV,HIREDATEV FROM EMP WHERE
EMPNO=EMPNOV;
   EXP:=ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATEV)/12,3);
   TA:=SALV*30/100;
   DA:=SALV*20/100;
   HRA:=SALV*15/100;
   LIC:=SALV*5/100;
   GROSS:=SALV+TA+DA+HRA-LIC;
   DBMS_OUTPUT.PUT_LINE('EMPNO   '||EMPNOV);
```

```
   DBMS_OUTPUT.PUT_LINE('ENAME   '||ENAMEV);
   DBMS_OUTPUT.PUT_LINE('SALARY  '||SALV);
   DBMS_OUTPUT.PUT_LINE('EXPERIENCE  '||EXP);
   DBMS_OUTPUT.PUT_LINE('TA   '||TA);
   DBMS_OUTPUT.PUT_LINE('DA   '||DA);
   DBMS_OUTPUT.PUT_LINE('HRA   '||HRA);
   DBMS_OUTPUT.PUT_LINE('LIC   '||LIC);
   DBMS_OUTPUT.PUT_LINE('GROSS   '||GROSS);
END;
/
```

80.WAP to accept the item no ,item name,qty,unit price and calculate the bill
If the bill > 500 then give discount 2% of bill amount and display the details

```
DECLARE
      INO NUMBER:=&INO;
      INAME VARCHAR2(50):='&INAME';
      QTY NUMBER(5):=&QTY;
      UP NUMBER(7,2):=&UP;
      DIS NUMBER(7,2):=0;
      BILL NUMBER(7,2);
      NET NUMBER(7,2);
BEGIN
      BILL:=QTY*UP;
      IF BILL>500 THEN
            DIS:=BILL*2/100;
      END IF;
      NET:=BILL-DIS;
      DBMS_OUTPUT.PUT_LINE('ITEM NO '||INO);
      DBMS_OUTPUT.PUT_LINE('ITEM NAME '||INAME);
      DBMS_OUTPUT.PUT_LINE('QUANTITY '||QTY);
      DBMS_OUTPUT.PUT_LINE('UNIT PRICE '||UP);
      DBMS_OUTPUT.PUT_LINE('BILL AMT '||BILL);
      DBMS_OUTPUT.PUT_LINE('DISCOUNT '||DIS);
      DBMS_OUTPUT.PUT_LINE('NET AMT '||NET);
END;
/
```

81.Write a program to generate sequence of numbers horizontally from 1 to 25

```
DECLARE
      V VARCHAR2(100);
BEGIN
      FOR I IN 1..25
      LOOP
            V:=V||' '||I;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

82.WAP to accept a empno and display empno,name,sal,exp,dname,grade and loc.

```
DECLARE
      EMPNOV NUMBER:=&EMPNO;
      ENAMEV EMP.ENAME%TYPE;
      HIREDATEV DATE;
      SALV EMP.SAL%TYPE;
      EXP NUMBER;
      DNAMEV DEPT.DNAME%TYPE;
      GRADEV SALGRADE.GRADE%TYPE;
BEGIN
      SELECT ENAME,SAL,HIREDATE,DNAME,GRADE INTO
ENAMEV,SALV,HIREDATEV,DNAMEV,GRADEV FROM EMP,DEPT,SALGRADE
      WHERE EMPNO=EMPNOV AND EMP.DEPTNO=DEPT.DEPTNO AND SAL BETWEEN
LOSAL AND HISAL;
      EXP:=ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATEV)/12,3);
      DBMS_OUTPUT.PUT_LINE('EMPNO  '||EMPNOV);
      DBMS_OUTPUT.PUT_LINE('ENAME  '||ENAMEV);
      DBMS_OUTPUT.PUT_LINE('SALARY  '||SALV);
      DBMS_OUTPUT.PUT_LINE('EXPERIENCE  '||EXP||' YEARS');
      DBMS_OUTPUT.PUT_LINE('DNAME  '||DNAMEV);
      DBMS_OUTPUT.PUT_LINE('GRADE  '||GRADEV);
END;
/
```

83.WAP to accept a empno and display empno,based on experience
calculate the bonus and store it into the bonus table
If exp > 5 years then bonus is 1 month salary
If exp between 5 and 9 years then bonus is 20% of annual salary
If exp more than 9 years then bonus is 1 month sal plus 25% of annual
salary

```
DECLARE
      EMPNOV NUMBER:=&EMPNO;
      ENAMEV EMP.ENAME%TYPE;
      HIREDATEV DATE;
      SALV EMP.SAL%TYPE;
      EXP NUMBER;
      DNAMEV DEPT.DNAME%TYPE;
      GRADEV SALGRADE.GRADE%TYPE;
BEGIN
      SELECT ENAME,SAL,HIREDATE,DNAME,GRADE INTO
ENAMEV,SALV,HIREDATEV,DNAMEV,GRADEV FROM EMP,DEPT,SALGRADE
      WHERE EMPNO=EMPNOV AND EMP.DEPTNO=DEPT.DEPTNO AND SAL BETWEEN
LOSAL AND HISAL;
      EXP:=ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATEV)/12,3);
      DBMS_OUTPUT.PUT_LINE('EMPNO  '||EMPNOV);
      DBMS_OUTPUT.PUT_LINE('ENAME  '||ENAMEV);
      DBMS_OUTPUT.PUT_LINE('SALARY  '||SALV);
      DBMS_OUTPUT.PUT_LINE('EXPERIENCE  '||EXP||' YEARS');
      DBMS_OUTPUT.PUT_LINE('DNAME  '||DNAMEV);
      DBMS_OUTPUT.PUT_LINE('GRADE  '||GRADEV);
END;
/
```

84.WAP to accept the empno, based upon the dname transfer the emps ie,
make the changes in the emp table. Transfer the emps from Accounting
dept to Research, Research dept to Operation, Opertion dept to Sales

and Sales to Accounting dept

```
DECLARE
   EMPNOV NUMBER:=&EMPNO;
   DNAMEV VARCHAR2(20);
   DNAMEVV VARCHAR2(20);
BEGIN
   SELECT DNAME INTO DNAMEV FROM EMP,DEPT WHERE EMPNO=EMPNOV AND
EMP.DEPTNO=DEPT.DEPTNO;
   IF DNAMEV='ACCOUNTING' THEN
           DNAMEVV:='RESEARCH';
   ELSIF DNAMEV='RESEARCH' THEN
           DNAMEVV:='SALES';
   ELSIF DNAMEV='SALES' THEN
           DNAMEVV:='OPERATIONS';
   ELSIF DNAMEV='OPERATIONS' THEN
           DNAMEVV:='ACCOUNTING';
   END IF;
   UPDATE EMP SET DEPTNO=(SELECT DEPTNO FROM DEPT WHERE DNAME=DNAMEVV)
WHERE EMPNO=EMPNOV;
END;
/
```

85.WAP to accept the empno and display all the details of emp. If emp doesnot exist display the appreciate message

```
DECLARE
      EMPNOV NUMBER:=&EMPNO;
      EMPV EMP%ROWTYPE;
BEGIN
      SELECT * INTO EMPV FROM EMP WHERE EMPNO=EMPNOV;
      DBMS_OUTPUT.PUT_LINE('EMPNO  '||EMPV.EMPNO);
      DBMS_OUTPUT.PUT_LINE('ENAME  '||EMPV.ENAME);
      DBMS_OUTPUT.PUT_LINE('JOB  '||EMPV.JOB);
      DBMS_OUTPUT.PUT_LINE('SALARY  '||EMPV.SAL);
      DBMS_OUTPUT.PUT_LINE('HIREDATE  '||EMPV.HIREDATE);
      DBMS_OUTPUT.PUT_LINE('DEPTNO  '||EMPV.DEPTNO);
      DBMS_OUTPUT.PUT_LINE('MGRNO  '||EMPV.MGR);
      DBMS_OUTPUT.PUT_LINE('COMMISSION  '||EMPV.COMM);
EXCEPTION
           WHEN NO_DATA_FOUND THEN
           DBMS_OUTPUT.PUT_LINE('EMP NUMBER DOES NOT EXIST');
END;
/
```

86.WAP to accept the empno and print all the details of emp,dept and salgrade

```
DECLARE
      E EMP%ROWTYPE;
      D DEPT%ROWTYPE;
      S SALGRADE%ROWTYPE;
BEGIN
      SELECT * INTO E FROM EMP WHERE EMPNO=&EMPNO;
      SELECT * INTO D FROM DEPT WHERE E.DEPTNO=DEPT.DEPTNO;
      SELECT * INTO S FROM SALGRADE WHERE E.SAL BETWEEN LOSAL AND
HISAL;
```

```
            DBMS_OUTPUT.PUT_LINE('EMPNO  '||E.EMPNO);
            DBMS_OUTPUT.PUT_LINE('DEPTNO  '||D.DEPTNO);
            DBMS_OUTPUT.PUT_LINE('DNAME  '||D.DNAME);
            DBMS_OUTPUT.PUT_LINE('LOCATION  '||D.LOC);
            DBMS_OUTPUT.PUT_LINE('GRADE  '||S.GRADE);
            DBMS_OUTPUT.PUT_LINE('HISALARY  '||S.HISAL);
            DBMS_OUTPUT.PUT_LINE('LOWSALARY  '||S.LOSAL);
END;
/
```

87.WAP to accept the mgrno and display the empno,ename,sal,dname and grade of all emps working under that mgr

```
DECLARE
      MGRV NUMBER:=&MGRV;
      CURSOR EMPCUR IS
      SELECT EMPNO,ENAME,SAL,DEPTNO,GRADE FROM EMP,SALGRADE WHERE
MGR=MGRV AND SAL BETWEEN LOSAL AND HISAL;
      X EMPCUR%ROWTYPE;
BEGIN
      OPEN EMPCUR;
      LOOP
            FETCH EMPCUR INTO X;
            EXIT WHEN EMPCUR%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMPNO  '||X.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME  '||X.ENAME);
            DBMS_OUTPUT.PUT_LINE('SALARY  '||X.SAL);
            DBMS_OUTPUT.PUT_LINE('DEPTNO  '||X.DEPTNO);
            DBMS_OUTPUT.PUT_LINE('GRADE  '||X.GRADE);
            DBMS_OUTPUT.PUT_LINE('******************');
      END LOOP;
      CLOSE EMPCUR;
END;
/
```

88.WAP to accept the empno and display the exp with minimum 3 decimal places

```
DECLARE
      EMPNOV NUMBER:=&EMPNOV;
      HIREDATEV DATE;
      EXPV NUMBER(10,5);
BEGIN
      SELECT HIREDATE INTO HIREDATEV FROM EMP WHERE EMPNO=EMPNOV;
      EXPV:=ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATEV)/12,3);
      DBMS_OUTPUT.PUT_LINE('EXPERIENCE OF EMP'||EMPNOV||' IS '||EXPV||'
YEARS ');
END;
/
```

89.Write a program to print the following series

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

90.Write a program to print the following series

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN REVERSE 1..I
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

91.Write a program to print the following series

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN REVERSE 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
```

```
/
```

92.Write a program to print the following series

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..5
            LOOP
                  V:=V||' '||I;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

93.Write a program to print the following series

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..5
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

94.Write a program to print the following series

```
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
```

```
      LOOP
            FOR J IN REVERSE 1..5
            LOOP
                  IF I<=J THEN
                  V:=V||' '||J;
                  END IF;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

95.Write a program to print the following series

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN REVERSE 1..5
      LOOP
            FOR J IN 1..5
            LOOP
                  IF I>=J THEN
                  V:=V||' '||I;
                  END IF;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

96.Write a program to print the following series

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||I;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
```

```
/
```

97.Write a program to print the following series

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

```
DECLARE
      A NUMBER:=1;
      V VARCHAR2(20):=1;
BEGIN
      DBMS_OUTPUT.PUT_LINE(V);
      FOR I IN 1..4
      LOOP
            IF SUBSTR(V,1,1)='1' THEN
                  V:='0'||V;
            ELSE
                  V:='1'||V;
            END IF;
      DBMS_OUTPUT.PUT_LINE(V);
      END LOOP;
END;
/
```

98.Write a program to print the following series

```
*
* *
* * *
* * * *
* * * * *
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||'*';
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

99.Write a program to print the following series

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
```

*

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||'*';
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
      FOR I IN REVERSE 1..5
      LOOP
            FOR J IN 2..I
            LOOP
                  V:=V||' '||'*';
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

100.Write a program to print the following series

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN I..5
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

101.Write a program to print the following series

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
DECLARE
      V VARCHAR2(20);
```

```
BEGIN
      FOR I IN REVERSE 1..5
      LOOP
            FOR J IN REVERSE 1..I
            LOOP
                  V:=V||' '||J;
            END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
      V:=NULL;
      END LOOP;
END;
/
```

102.WAP to accept 2 nos and find the sum and product of the nos and print the output

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      S NUMBER;
      M NUMBER;
BEGIN
      S:=A+B;
      M:=A*B;
      DBMS_OUTPUT.PUT_LINE('SUM OF '||'A'||' AND '||'B'||' IS '||S);
      DBMS_OUTPUT.PUT_LINE('PRODUCT OF '||'A'||' AND '||'B'||' IS
'||M);
END;
/
```

103.WAP to accept 2 nos and find the remainder when the first number is divided by sencond(dont use mod function)

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      C NUMBER;
      M NUMBER;
BEGIN
      C:=TRUNC(A/B);
      M:=A-C*B;
      DBMS_OUTPUT.PUT_LINE('REMAINDER IS '||M);
END;
/
```

104.WAP to display all the ASCII characters 0-9--48-57,A-Z--65-90,a-z--97-122

```
BEGIN
      FOR I IN 1..255
      LOOP
      DBMS_OUTPUT.PUT_LINE(I||'-'||CHR(I));
      END LOOP;
END;
/
```

105.Print the following format

```
ORACLE
ORACL
ORAC
ORA
OR
O
```

```
DECLARE
      STR VARCHAR2(10):='&STR';
      L VARCHAR2(10);
      N NUMBER(15);
BEGIN
      N:=LENGTH(STR);
      WHILE N>=1
      LOOP
      L:=SUBSTR(STR,1,N);
      N:=N-1;
      DBMS_OUTPUT.PUT_LINE(L);
      END LOOP;
END;
/
```

106.WAP to display "GOOD MORNING" or "GOOD AFTERNOON" or "GOOD NIGHT" depending upon the current time

```
DECLARE
      HH NUMBER;
BEGIN
      HH:=TO_CHAR(SYSDATE,'HH24');
      IF HH>6 AND HH<12 THEN
            DBMS_OUTPUT.PUT_LINE('GOOD MORNING');
      ELSIF HH>=12 AND HH<18 THEN
            DBMS_OUTPUT.PUT_LINE('GOOD AFTERNOON');
      ELSIF HH>=18 AND HH<25 THEN
            DBMS_OUTPUT.PUT_LINE('GOOD NIGHT');
      END IF;
END;
/
```

107.WAP to accept two strings and concat the two strings

```
DECLARE
      STR VARCHAR2(20):='&STR';
      STR1 VARCHAR2(20):='&STR1';
      V VARCHAR2(40);
BEGIN
      V:=STR||''||STR1;
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

108.WAP to accept a string and count the no of chars,words in that string

```
DECLARE
      STR VARCHAR2(20):='&STR';
      NOC NUMBER(4):=0;
```

```
      NOW NUMBER(4):=1;
      S CHAR;
BEGIN
      FOR I IN 1..LENGTH(STR)
      LOOP
            S:=SUBSTR(STR,I,1);
            NOC:=NOC+1;
            IF S=' ' THEN
                  NOW:=NOW+1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE NO. OF CHARS '||NOC);
      DBMS_OUTPUT.PUT_LINE('THE NO. OF WORDS '||NOW);
END;
/
```

109.WAP to accept the octal number and print it in decimal format

```
DECLARE
      N VARCHAR2(20):='&N';
      A NUMBER;
      P NUMBER:=0;
      C CHAR;
BEGIN
      A:=LENGTH(N);
      FOR I IN 1..A
      LOOP
            C:=SUBSTR(N,I,1);
            P:=P+C*POWER(8,A-I);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE INTEGER OF A GIVEN OCTAL IS '||P);
END;
/
```

110.WAP to accept the mgr and find how many emps are working under that mgr

```
DECLARE
   MGRV EMP.MGR%TYPE:=&MGRNO;
   N NUMBER:=0;
BEGIN
   SELECT COUNT(*) INTO N FROM EMP WHERE MGR=MGRV;
   DBMS_OUTPUT.PUT_LINE('NUMBER OF EMPLOYEE UNDER THAT MANAGER ARE
'||N);
END;
/
```

111.WAP to accept the empno and update the employee row on the following
If sal < 2600 then sal=sal+10% of sal make the changes in the emp table

```
DECLARE
   EMPNOV EMP.EMPNO%TYPE:=&EMPNO;
   SALV NUMBER(7,2):=0;
BEGIN
   SELECT SAL INTO SALV FROM EMP WHERE EMPNO=EMPNOV;
   IF SALV < 2600 THEN
           SALV:=SALV+SALV*(10/100);
```

```
   END IF;
   UPDATE EMP SET SAL=SALV WHERE EMPNO=EMPNOV;
   DBMS_OUTPUT.PUT_LINE('EMPNO IS '||EMPNOV);
   DBMS_OUTPUT.PUT_LINE('SAL IS '||SALV);
END;
/
```

112.Write the floyd's triangle

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
..............
79..............91
```

```
DECLARE
      N NUMBER:=1;
      V VARCHAR2(100);
BEGIN
      FOR I IN 1..92
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||N;
                  N:=N+1;
                  EXIT WHEN N=92;
            END LOOP;
            DBMS_OUTPUT.PUT_LINE(V);
            EXIT WHEN N=92;
            V:=NULL;
      END LOOP;
END;
/
```

113.WAP to accept the real value and print integer value only

```
DECLARE
      N NUMBER(7,3):=&N;
      A NUMBER(5);
BEGIN
      A:=TRUNC(N);
      DBMS_OUTPUT.PUT_LINE('REAL VALUE IS '||A);
END;
/
```

114.WAP to calculate the sum of n odd factorials

```
DECLARE
      N NUMBER:=&N;
      S NUMBER:=0;
      F NUMBER:=1;
BEGIN
      FOR I IN 1..N
      LOOP
            IF MOD(I,2)!=0 THEN
```

```
                  FOR J IN 1..I
                  LOOP
                        F:=F*J;
                  END LOOP;
            S:=S+F;
            F:=1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM '||S);
END;
/
```

115.WAP to calculate the sum of n even factorials

```
DECLARE
      N NUMBER:=&N;
      S NUMBER:=0;
      F NUMBER:=1;
BEGIN
      FOR I IN 1..N
      LOOP
            IF MOD(I,2)=0 THEN
                  FOR J IN 1..I
                  LOOP
                        F:=F*J;
                  END LOOP;
            S:=S+F;
            F:=1;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('SUM '||S);
END;
/
```

116.WAP to generate the nos which are prime and odd between 1 and 100

```
DECLARE
      N NUMBER;
      CNT NUMBER:=0;
BEGIN
      FOR I IN 1..100
      LOOP
            FOR J IN 1..I
            LOOP
                  IF MOD(I,J)=0 THEN
                        CNT:=CNT+1;
                  END IF;
            END LOOP;
            IF CNT <= 2 THEN
                  IF MOD(I,2)!=0 THEN
                        DBMS_OUTPUT.PUT_LINE(I);
                  END IF;
            END IF;
            CNT:=0;
      END LOOP;
END;
/
```

117.Write a program to generate following series
12
12 22
12 22 32
12 22 32 42
12 22 32 42 52

```
DECLARE
      V VARCHAR2(20);
BEGIN
      FOR I IN 1..5
      LOOP
            FOR J IN 1..I
            LOOP
                  V:=V||' '||J||CHR(178);
            END LOOP;
            DBMS_OUTPUT.PUT_LINE(V);
            V:=NULL;
      END LOOP;
END;
/
```

118.Find the roots of a quadratic equation

```
DECLARE
   A NUMBER(4):=&A;
   B NUMBER(4):=&B;
   C NUMBER(4):=&C;
   D NUMBER(8,2);
   R1 NUMBER(8,2);
   R2 NUMBER(8,2);
BEGIN
   D:=POWER(B,2)-4*A*C;
   IF D = 0 THEN
           DBMS_OUTPUT.PUT_LINE('ROOTS ARE EQUAL');
   ELSIF D > 0 THEN
           R1:=(-B+SQRT(D))/2*A;
           R2:=(-B-SQRT(D))/2*A;
   DBMS_OUTPUT.PUT_LINE('FIRST ROOT IS '||R1);
   DBMS_OUTPUT.PUT_LINE('SECOND ROOT IS '||R2);
   ELSE
   DBMS_OUTPUT.PUT_LINE('ROOTS ARE IMAGINARY');
   END IF;
END;
/
```

119.WAP to accept the 2 diff nos, assume that first one is smaller and second one is highest value then print the all even nos in between them horizontally

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      V VARCHAR2(100);
BEGIN
      FOR I IN A..B
```

```
      LOOP
            IF MOD(I,2)=0 THEN
                  V:=V||' '||I;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

120.WAP to accept two diff nos assume that first one is smaller and second one is highest value then print the all odd nos in between them horizontally

```
DECLARE
      A NUMBER:=&A;
      B NUMBER:=&B;
      V VARCHAR2(100);
BEGIN
      FOR I IN A..B
      LOOP
            IF MOD(I,2)!=0 THEN
                  V:=V||' '||I;
            END IF;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

121. Write a program to accept a year and display the emps belongs to that year?

```
DECLARE
      Y NUMBER(4):=&YEAR;
      CURSOR YEAR IS
      SELECT * FROM EMP WHERE TO_CHAR(HIREDATE,'YYYY')=Y;
      B YEAR%ROWTYPE;
BEGIN
      OPEN YEAR;
      LOOP
            FETCH YEAR INTO B;
            EXIT WHEN YEAR%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('EMP NAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('EMP SAL IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('*************************');
      END LOOP;
      CLOSE YEAR;
END;
/
```

122.Write a program to accept a mgr and display who are working under that mgr?

```
DECLARE
      MGRV NUMBER(4):=&MGR;
      CURSOR AMGR IS
      SELECT * FROM EMP WHERE MGR=MGRV;
      B AMGR%ROWTYPE;
BEGIN
      OPEN AMGR;
      LOOP
            FETCH AMGR INTO B;
            EXIT WHEN AMGR%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('EMP NAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('EMP SAL IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('*************************');
      END LOOP;
      CLOSE AMGR;
END;
/
```

123. Write a program to accept the grade and display emps belongs to that grade?

```
DECLARE
      GRADEV SALGRADE.GRADE%TYPE:=&GRADE;
      CURSOR A IS
      SELECT EMP.*,GRADE FROM EMP,SALGRADE WHERE SAL BETWEEN LOSAL AND
HISAL AND GRADE=GRADEV;
      B A%ROWTYPE;
```

```
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('MGR NO IS ' || B.MGR);
            DBMS_OUTPUT.PUT_LINE('COMM IS ' || B.COMM);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('GRADE IS ' || B.GRADE);
            DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('*************************');
      END LOOP;
      CLOSE A;
END;
/
```

124. Write a program to accept a deptno and display who are working in that dept?

```
DECLARE
      DEPTV EMP.DEPTNO%TYPE:=&DEPTNO;
      CURSOR A IS
      SELECT * FROM EMP WHERE DEPTNO=DEPTV;
      B A%ROWTYPE;
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('MGR NO IS ' || B.MGR);
            DBMS_OUTPUT.PUT_LINE('COMM IS ' || B.COMM);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);

DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('*************************');
      END LOOP;
      CLOSE A;
END;
/
```

125. Write a program to display all the information of emp table?

```
DECLARE
      CURSOR A IS
      SELECT * FROM EMP;
      B A%ROWTYPE;
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
```

```
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('MGR NO IS ' || B.MGR);
            DBMS_OUTPUT.PUT_LINE('COMM IS ' || B.COMM);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);
            DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('*************************');
      END LOOP;
      CLOSE A;
END;
/
```

126. Write a program to accept the location and display empno, name, sal , date of join and also display the total salary, avg salary and no of emps?

```
DECLARE
   LOCV DEPT.LOC%TYPE:='&LOC';
   TOT NUMBER(10,2):=0;
   ASAL NUMBER(10,2):=0;
   NOEMPS NUMBER(5):=0;
   CURSOR A IS
   SELECT EMP.*,LOC FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND
LOC=LOCV;
   B A%ROWTYPE;
BEGIN
   OPEN A;
   LOOP
           FETCH A INTO B;
           NOEMPS:=NOEMPS+1;
           TOT:=TOT+B.SAL;
           ASAL:=TOT/NOEMPS;
           EXIT WHEN A%NOTFOUND;
           DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
           DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
           DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
           DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
           DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);
           DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
           DBMS_OUTPUT.PUT_LINE('LOC IS ' || B.LOC);
           DBMS_OUTPUT.PUT_LINE('TOT IS ' || TOT);
           DBMS_OUTPUT.PUT_LINE('NOEMPS IS ' || NOEMPS);
           DBMS_OUTPUT.PUT_LINE('ASAL IS ' || ASAL);
           DBMS_OUTPUT.PUT_LINE('*************************');
   END LOOP;
   CLOSE A;
END;
/
```

127. Write a program to accept a range of salary (that is lower boundary and higher boundary) and print the details of emps along with loc,grade and exp?

```
DECLARE
```

```
   LOSALV SALGRADE.LOSAL%TYPE:=&LOSAL;
   HISALV SALGRADE.HISAL%TYPE:=&HISAL;
   EXP NUMBER(5,2);
           CURSOR A IS
   SELECT EMP.*,LOC,GRADE FROM EMP,DEPT,SALGRADE WHERE
EMP.DEPTNO=DEPT.DEPTNO
   AND SAL BETWEEN LOSALV AND HISALV
   AND SAL BETWEEN LOSAL AND HISAL;
   B A%ROWTYPE;
BEGIN
   OPEN A;
   LOOP
           FETCH A INTO B;
           EXIT WHEN A%NOTFOUND;
           EXP:=MONTHS_BETWEEN(SYSDATE,B.HIREDATE)/12;
                DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
                DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
                DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
                DBMS_OUTPUT.PUT_LINE('LOC IS ' || B.LOC);
                DBMS_OUTPUT.PUT_LINE('EXP IS ' || EXP);
                DBMS_OUTPUT.PUT_LINE('GRADE IS ' || B.GRADE);
                DBMS_OUTPUT.PUT_LINE('************************');
   END LOOP;
   CLOSE A;
END;
/
```

128. Write a program to print all the details of emps accepting the job?

```
DECLARE
   JOBV EMP.JOB%TYPE:='&JOB';
   CURSOR A IS
   SELECT * FROM EMP WHERE JOB=JOBV;
   B A%ROWTYPE;
BEGIN
   OPEN A;
   LOOP
           FETCH A INTO B;
           EXIT WHEN A%NOTFOUND;
           DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
           DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
           DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
           DBMS_OUTPUT.PUT_LINE('************************');
   END LOOP;
   CLOSE A;
END;
/
```

129. Write a program to display the details of emps year wise?

```
DECLARE
      CURSOR YEARS IS
      SELECT DISTINCT TO_CHAR(HIREDATE,'YYYY') YEARS1 FROM EMP ORDER BY
1;
      YEAR YEARS%ROWTYPE;
      CURSOR A IS
```

```
      SELECT * FROM EMP WHERE TO_CHAR(HIREDATE,'YYYY')=YEAR.YEARS1;
      B A%ROWTYPE;
BEGIN
      DBMS_OUTPUT.ENABLE(10000);
      OPEN YEARS;
      DBMS_OUTPUT.PUT_LINE('********************');
      LOOP
            FETCH YEARS INTO YEAR;
            EXIT WHEN YEARS%NOTFOUND;
              DBMS_OUTPUT.PUT_LINE('YEAR :' || YEAR.YEARS1);
                DBMS_OUTPUT.PUT_LINE('**********************');
              OPEN A;
            LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE('EMPNO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SALARY IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);
            DBMS_OUTPUT.PUT_LINE('************************');
      END LOOP;
      CLOSE A;
      END LOOP;
      CLOSE YEARS;
END;
/
```

130. Write a program to accept empno and print all the details along with loc and grade?

```
DECLARE
      EMPNOV EMP.EMPNO%TYPE:=&EMPNO;
      CURSOR A IS
      SELECT EMP.*,GRADE,LOC FROM EMP,DEPT,SALGRADE
      WHERE EMP.DEPTNO=DEPT.DEPTNO
      AND SAL BETWEEN LOSAL AND HISAL AND EMPNO=EMPNOV;
      B A%ROWTYPE;
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
              DBMS_OUTPUT.PUT_LINE('EMPNO IS ' || B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SALARY IS ' || B.SAL);
            DBMS_OUTPUT.PUT_LINE('JOB IS ' || B.JOB);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('LOC IS ' || B.LOC);
            DBMS_OUTPUT.PUT_LINE('GRADE IS ' || B.GRADE);
            DBMS_OUTPUT.PUT_LINE('************************');
      END LOOP;
      CLOSE A;
END;
/
```

131. Write a procedure to create your own print statement?

```
CREATE OR REPLACE PROCEDURE PRINT(V VARCHAR2)
IS
BEGIN
      DBMS_OUTPUT.PUT_LINE(V);
END;
/
```

132. Write a procedure to accept the deptno as parameter and display the details of that dept also display the total salary, no of employees, max sal and avg sal?

```
CREATE OR REPLACE PROCEDURE EMPPRO(DEPTNOV NUMBER)
IS
      CURSOR A IS
      SELECT * FROM EMP WHERE DEPTNO=DEPTNOV;
      B A%ROWTYPE;
      NOE NUMBER:=0;
      TOT NUMBER:=0;
      AVGS NUMBER(7,2):=0;
      MAXS NUMBER(7,2):=0;
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMPNO :'||B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME :'||B.ENAME);
            DBMS_OUTPUT.PUT_LINE('JOB :'||B.JOB);
            DBMS_OUTPUT.PUT_LINE('SAL :'||B.SAL);
            DBMS_OUTPUT.PUT_LINE('HIREDATE :'||B.HIREDATE);
            DBMS_OUTPUT.PUT_LINE('COMM :'||B.COMM);
            DBMS_OUTPUT.PUT_LINE('**********************');
            TOT:=TOT+B.SAL;
            NOE:=NOE+1;
            IF B.SAL>MAXS THEN
            MAXS:=B.SAL;
            END IF;
      END LOOP;
            AVGS:=TOT/NOE;
            DBMS_OUTPUT.PUT_LINE('NO OF EMPLOYEE :'||NOE);
            DBMS_OUTPUT.PUT_LINE('TOTAL SALARY :'||TOT);
            DBMS_OUTPUT.PUT_LINE('AVG SALARY :'||AVGS);
            DBMS_OUTPUT.PUT_LINE('MAX SALARY :'||MAXS);
      CLOSE A;
END;
/
```

133. Write a procedure to accept two different numbers and print all odd numbers between the two given numbers?

```
CREATE OR REPLACE PROCEDURE ODDNO(A NUMBER,B NUMBER)
IS
      N NUMBER(4);
BEGIN
      N:=A;
      WHILE N<B
      LOOP
```

```
            IF MOD(N,2)!=0 THEN
            DBMS_OUTPUT.PUT_LINE(N);
            END IF;
      N:=N+1;
      END LOOP;
END;
/
```

134. Write a procedure to accept two different numbers and print even numbers between the two given numbers?

```
CREATE OR REPLACE PROCEDURE EVENNO(A NUMBER,B NUMBER)
IS
      N NUMBER(4);
BEGIN
      N:=A;
      WHILE N<B
      LOOP
            IF MOD(N,2)=0 THEN
            DBMS_OUTPUT.PUT_LINE(N);
            END IF;
      N:=N+1;
      END LOOP;
END;
/
```

135. Write a procedure to accept deptno as input and print the details of emps along with grade?

```
CREATE OR REPLACE PROCEDURE EMP_DETAIL(DEPTNOV NUMBER)
IS
      CURSOR A IS
      SELECT EMP.*,GRADE FROM EMP,SALGRADE
      WHERE SAL BETWEEN LOSAL AND HISAL
      AND DEPTNO=DEPTNOV;
      B A%ROWTYPE;
BEGIN
      OPEN A;
      LOOP
            FETCH A INTO B;
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMPNO IS '||B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS '||B.ENAME);
            DBMS_OUTPUT.PUT_LINE('JOB IS '||B.JOB);
            DBMS_OUTPUT.PUT_LINE('SAL IS '||B.SAL);
            DBMS_OUTPUT.PUT_LINE('DEPTNO IS '||B.DEPTNO);
            DBMS_OUTPUT.PUT_LINE('GRADE IS '||B.GRADE);
      END LOOP;
      CLOSE A;
END;
/
```

136. Write a procedure to accept a number as parameter and print its multiplication table?

```
CREATE OR REPLACE PROCEDURE MULT(A NUMBER)
IS
```

```
      B NUMBER(2) DEFAULT 1;
      C NUMBER(3);
BEGIN
      WHILE B<=10
      LOOP
            C:=A*B;
            DBMS_OUTPUT.PUT_LINE(A||'*'||B||'='||C);
            B:=B+1;
      END LOOP;
END;
/
```

137. Write a procedure to accept two different numbers as input and print all even numbers and odd numbers in between them in two different horizontal lines?

```
CREATE OR REPLACE PROCEDURE EVENODD(A NUMBER,B NUMBER)
IS
      N NUMBER;
      EV VARCHAR2(1000);
      OD VARCHAR2(1000);
BEGIN
      N:=A;
      WHILE N<B
      LOOP
            IF MOD(N,2)!=0 THEN
            OD:=OD||' '||N;
            ELSE
            EV:=EV||' '||N;
            END IF;
      N:=N+1;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('THE ODD NOS ARE '||OD);
      DBMS_OUTPUT.PUT_LINE('THE EVEN NOS ARE '||EV);
END;
/
```

138. Write a procedure to accept a string and check whether it is palindrome or not?

```
CREATE OR REPLACE PROCEDURE STRPAL(STR VARCHAR2)
IS
      STR1 VARCHAR2(10);
      S VARCHAR2(10);
BEGIN
      FOR I IN REVERSE 1..LENGTH(STR)
      LOOP
            S:=SUBSTR(STR,I,1);
            STR1:=STR1||S;
      END LOOP;
      IF STR1=STR THEN
      DBMS_OUTPUT.PUT_LINE('IT IS PALINDROME '||STR1);
      ELSE
      DBMS_OUTPUT.PUT_LINE('IT IS NOT PALINDROME '||STR1);
      END IF;
END;
/
```

139. Write a procedure to accept a string and print it in reverse order?

```
CREATE OR REPLACE PROCEDURE STRREV(STR VARCHAR2)
IS
      STR1 VARCHAR2(10);
      S VARCHAR2(10);
BEGIN
      FOR I IN REVERSE 1..LENGTH(STR)
      LOOP
            S:=SUBSTR(STR,I,1);
            STR1:=STR1||S;
      END LOOP;
      DBMS_OUTPUT.PUT_LINE('ORIGINAL '||STR);
      DBMS_OUTPUT.PUT_LINE('REVERSE '||STR1);
END;
/
```

140. Write a procedure to accept the empno and print all the details of emp along with exp, grade and loc?

```
CREATE OR REPLACE PROCEDURE EMP_DET(EMPNOV NUMBER)
IS
      EXP NUMBER(6,2);
      E EMP%ROWTYPE;
      GRADEV SALGRADE.GRADE%TYPE;
      LOCV DEPT.LOC%TYPE;
BEGIN
      SELECT EMP.* INTO E FROM EMP WHERE EMPNO=EMPNOV;
      SELECT LOC INTO LOCV FROM DEPT WHERE DEPT.DEPTNO=E.DEPTNO;
      SELECT GRADE INTO GRADEV FROM SALGRADE WHERE E.SAL BETWEEN LOSAL
AND HISAL;
      EXP:=MONTHS_BETWEEN(SYSDATE,E.HIREDATE)/12;
      DBMS_OUTPUT.PUT_LINE('EMPNO IS '||E.EMPNO);
      DBMS_OUTPUT.PUT_LINE('ENAME IS '||E.ENAME);
      DBMS_OUTPUT.PUT_LINE('SAL IS '||E.SAL);
      DBMS_OUTPUT.PUT_LINE('JOB IS '||E.JOB);
      DBMS_OUTPUT.PUT_LINE('LOC IS '||LOCV);
      DBMS_OUTPUT.PUT_LINE('GRADE IS '||GRADEV);
      DBMS_OUTPUT.PUT_LINE('EXP IS '||EXP);
END;
/
```

141. Write a procedure to accept dname irrespective of case and print all the details of emps?

```
CREATE OR REPLACE PROCEDURE DETAILS(DNAMEV VARCHAR2)
IS
   CURSOR A IS
   SELECT EMP.*,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND
DNAME=DNAMEV;
   B A%ROWTYPE;
BEGIN
   OPEN A;
   LOOP
           FETCH A INTO B;
```

```
            EXIT WHEN A%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('EMPNO IS '||B.EMPNO);
            DBMS_OUTPUT.PUT_LINE('ENAME IS '||B.ENAME);
            DBMS_OUTPUT.PUT_LINE('SAL IS '||B.SAL);
            DBMS_OUTPUT.PUT_LINE('JOB IS '||B.JOB);
            DBMS_OUTPUT.PUT_LINE('DNAME IS '||B.DNAME);
            DBMS_OUTPUT.PUT_LINE('HIREDATE IS '||B.HIREDATE);
   END LOOP;
END;
/
```

142. Write a procedure to accept a string and print it in reverse case?

```
CREATE OR REPLACE PROCEDURE S_R_CASE(STR VARCHAR2)
IS
      S VARCHAR2(10);
      V VARCHAR2(10);
      N NUMBER(3);
BEGIN
      FOR I IN 1..LENGTH(STR)
      LOOP
            S:=SUBSTR(STR,I,1);
            N:=ASCII(S);
            IF N BETWEEN 65 AND 90 THEN
                  V:=V||CHR(N+32);
            ELSE
                  V:=V||CHR(N-32);
            END IF;
      END LOOP;
DBMS_OUTPUT.PUT_LINE('STRING IN REVERSE CASE IS '||V);
END;
/
```

143. Write a function to accept the empno and return exp with minimum 3 decimal?

```
CREATE OR REPLACE FUNCTION E_DETAILS(EMPNOV NUMBER) RETURN NUMBER
IS
   HIREDATEV EMP.HIREDATE%TYPE;
   EXP NUMBER(6,3);
BEGIN
   SELECT HIREDATE INTO HIREDATEV FROM EMP WHERE EMPNO=EMPNOV;
   EXP:=MONTHS_BETWEEN(SYSDATE,HIREDATEV)/12;
   RETURN EXP;
END;
/
```

144. Write a function to accept a number and print the factorial of that number?

```
CREATE OR REPLACE FUNCTION FAC(NUM NUMBER) RETURN NUMBER
IS
   FACT NUMBER(4):=1;
BEGIN
   FOR I IN REVERSE 1..NUM
   LOOP
            FACT:=FACT*I;
```

```
   END LOOP;
   RETURN FACT;
END;
/
```

145. Write a function to accept a grade and return the number of emps belongs to that grade?

```
CREATE OR REPLACE FUNCTION EMPGRADE(GRADEV NUMBER) RETURN VARCHAR2
 IS
    N NUMBER(4);
 BEGIN
    SELECT COUNT(*) INTO N FROM EMP,SALGRADE
    WHERE SAL BETWEEN LOSAL AND HISAL AND GRADE=GRADEV;
    RETURN 'NO OF EMPS ARE'||N;
 END;
/
```

146. Write a program to accept the mgr number and return no of emp working at that mgr?

```
CREATE OR REPLACE FUNCTION N_EMPS(MGRV NUMBER) RETURN VARCHAR2
IS
   N NUMBER(4);
BEGIN
   SELECT COUNT(*) INTO N FROM EMP WHERE MGR=MGRV;
   RETURN 'THE NO OF EMPS ARE WORKING UNDER THIS MGR IS '||N;
END;
/
```

147. Write a function to accept a character string and print it in reverse case?

```
CREATE OR REPLACE FUNCTION REVERSE(STR VARCHAR2) RETURN VARCHAR2
IS
   STR1 VARCHAR2(20);
   S VARCHAR2(20);
   N NUMBER(4);
BEGIN
   FOR I IN 1..LENGTH(STR)
   LOOP
           S:=SUBSTR(STR,I,1);
           N:=ASCII(S);
           IF N BETWEEN 65 AND 90 THEN
                   STR1:=STR1||CHR(N+32);
           ELSE
                   STR1:=STR1||CHR(N-32);
           END IF;
   END LOOP;
   RETURN 'THE REVERSE CASE IS '||STR1;
END;
/
```

148. Write a function to accept a string and check whether it is palindrome or not?

```
 CREATE OR REPLACE FUNCTION STRPAL1(STR VARCHAR2) RETURN VARCHAR2
```

```
IS
   STR1 VARCHAR2(10);
   S VARCHAR2(10);
BEGIN
   FOR I IN REVERSE 1..LENGTH(STR)
   LOOP
           S:=SUBSTR(STR,I,1);
           STR1:=STR1||S;
   END LOOP;
   IF STR1=STR THEN
   RETURN 'IT IS PALINDROME '||STR1;
   ELSE
   RETURN 'IT IS NOT PALINDROME '||STR1;
   END IF;
END;
/
```

149. Write a function to accept the grade and return max, tot, avg salary and number of emps belongs to that grade as script without using any group functions?

```
CREATE OR REPLACE FUNCTION EMP_DETAILS_SCRIPT (GRADEV
SALGRADE.GRADE%TYPE) RETURN VARCHAR2
IS
      V VARCHAR2(30000);
      CURSOR EMP_CUR IS
      SELECT EMP.*,GRADE,DNAME FROM DEPT,EMP,SALGRADE
      WHERE GRADE=GRADEV AND EMP.DEPTNO=DEPT.DEPTNO AND
      SAL BETWEEN LOSAL AND HISAL;
      EMP_CUR_V EMP_CUR%ROWTYPE;
      MAXSAL EMP.SAL%TYPE:=0;
      MINSAL EMP.SAL%TYPE;
      AVGSAL NUMBER(6,2);
      SUMSAL NUMBER(10,2):=0;
      CNT NUMBER:=0;
      FLAG CHAR:=0;
      EX EXCEPTION;
BEGIN
      OPEN EMP_CUR;
      LOOP
            FETCH EMP_CUR INTO EMP_CUR_V;
            EXIT WHEN EMP_CUR%NOTFOUND;
            IF MAXSAL < EMP_CUR_V.SAL THEN
                  MAXSAL:=EMP_CUR_V.SAL;
            END IF;
            IF FLAG=0 THEN
                  MINSAL:=EMP_CUR_V.SAL;
                  FLAG:=1;
            ELSIF FLAG=1 AND MINSAL > EMP_CUR_V.SAL THEN
                  MINSAL:=EMP_CUR_V.SAL;
            END IF;
            SUMSAL:=SUMSAL+EMP_CUR_V.SAL;
            CNT:=CNT+1;
      ENDLOOP;
      IF CNT=0 THEN
      RAISE EX;
      END IF;
```

```
      AVGSAL:=SUMSAL/CNT;
      V:='THE MAXIMUM SALARY OF GRADE' ||GRADEV||' IS'||MAXSAL||'
MINIMUM SALARY IS'||MINSAL||
      'AVERAGE SALARY IS'||AVGSAL||' TOTAL EMPS WORKING FOR THIS GRADE
ARE'||CNT;
      CLOSE EMP_CUR;
      RETURN V;
      EXCEPTION
      WHEN EX THEN
      RETURN 'THERE IS NO EMPLOYEE WORKING FOR THIS GRADE, CHECK AND
RE-ENTER THE GRADE....';
END;
/
```

150. Create a package to store the following procedure for multiplication table,even-odd, function for factorial and function for palindrome?

```
CREATE OR REPLACE PACKAGE DATA
IS
      PROCEDURE MULT(A NUMBER);
      PROCEDURE EVEN_ODD(N NUMBER);
      FUNCTION FACT(N NUMBER) RETURN NUMBER;
      PRAGMA RESTRICT_REFERENCES(FACT,WNDS);
      FUNCTION PALEN(SRT VARCHAR2) RETURN VARCHAR2;
      PRAGMA RESTRICT_REFERENCES(PALEN,WNDS);
END;
/

CREATE OR REPLACE PACKAGE BODY DATA
IS
      PROCEDURE MULT(A NUMBER)
      IS
      M NUMBER;
      BEGIN
            FOR I IN 1..10
            LOOP
                  M:=A*I;
                  DBMS_OUTPUT.PUT_LINE(A||'*'||I||'='||M);
            END LOOP;
      END;
      PROCEDURE EVEN_ODD(N NUMBER)
      IS
      BEGIN
            IF MOD(N,2)=0 THEN
            DBMS_OUTPUT.PUT_LINE(N||' IS EVEN NUMBER');
            ELSE
            DBMS_OUTPUT.PUT_LINE(N||' IS NOT EVEN NUMBER');
            END IF;
      END;
      FUNCTION FACT(N NUMBER) RETURN NUMBER
      IS
            F NUMBER:=1;
      BEGIN
            FOR I IN 1..N
            LOOP
                  F:=F*I;
```

```
            END LOOP;
            RETURN F;
      END;
      FUNCTION PALEN(SRT VARCHAR2) RETURN VARCHAR2
      IS
            S CHAR;
            V VARCHAR2(50);
      BEGIN
            FOR I IN REVERSE 1..LENGTH(SRT)
            LOOP
                  S:=SUBSTR(SRT,I,1);
                  V:=V||S;
            END LOOP;
            IF V=SRT THEN
                  RETURN 'PALINDROME';
            ELSE
                  RETURN 'NOT PALINDROME';
            END IF;
      END;
END;
/
```

151.Write a database trigger halt the transaction on Sunday on EMP table

```
CREATE OR REPLACE TRIGGER SUN_TRI
AFTER INSERT OR UPDATE OR DELETE ON EMP
DECLARE
   DY VARCHAR2(200);
BEGIN
   DY:=TO_CHAR(SYSDATE,'DY');
   IF DY='SUN' THEN
          RAISE_APPLICATION_ERROR(-20005,'TODAY IS SUNDAY TRANSACTION
NOT ALLOWED TODAY');
   END IF;
END;
/
```

152.Write a database trigger halt the transaction of USER SCOTT on table EMP

```
CREATE OR REPLACE TRIGGER SCOTT_TRI
BEFORE INSERT OR UPDATE OR DELETE ON EMP
BEGIN
   IF USER = 'SCOTT' THEN
   RAISE_APPLICATION_ERROR(-20006,'TRANSACTION NOT ALLOWED FOR SCOTT');
   END IF;
END;
/
```

153.Write a database trigger halt the transaction between the the time 6pm to 10am on table emp

```
CREATE OR REPLACE TRIGGER OVER_TIME_TRI
BEFORE INSERT OR DELETE OR UPDATE ON EMP
DECLARE
   T NUMBER;
```

```
BEGIN
   T:=TO_CHAR(SYSDATE,'HH24');
   IF T NOT BETWEEN 10 AND 18 THEN
   RAISE_APPLICATION_ERROR(-20007,'TIME ALREADY OVER.....TRANSACTION
NOT ALLOWED NOW');
   END IF;
END;
```

154.Write a database trigger to halt the transaction for the employee SALESMAN and
PRESIDENT

```
CREATE OR REPLACE TRIGGER SALES_PRI
BEFORE INSERT OR UPDATE OR DELETE ON EMP
FOR EACH ROW
WHEN (OLD.JOB IN ('SALESMAN','PRESIDENT') OR
NEW.JOB IN ('SALESMAN','PRESIDENT'))
BEGIN
      RAISE_APPLICATION_ERROR(-20008,'TRANSACTION NOT ALLOWED FOR
SALESMAN AND PRESIDENT....');
END;
/
```

155.Write a database trigger stroe the username ,type of transaction ,date of transaction and time of transaction of table emp into the table EMP_LOG

```
CREATE OR REPLACE TRIGGER TRANS_TYPE
AFTER INSERT OR UPDATE OR DELETE ON EMP
DECLARE
      V VARCHAR2(50);
BEGIN
      IF INSERTING THEN
      V:='I';
      ELSIF UPDATING THEN
      V:='U';
      ELSE
      V:='D';
      END IF;
INSERT INTO EMP_LOG VALUES
(USER,V,SYSDATE,TO_CHAR(SYSDATE,'HH:MI:SS'));
END;
/
```

156.Write a database trigger store the deleted data of EMP table in EMPDEL table

```
CREATE OR REPLACE TRIGGER DEL_TRI
BEFORE DELETE ON EMP
FOR EACH ROW
BEGIN
      INSERT INTO EMPDEL
      VALUES
(:OLD.EMPNO,:OLD.ENAME,:OLD.JOB,:OLD.MGR,:OLD.HIREDATE,:OLD.SAL,:OLD.CO
MM,
      :OLD.DEPTNO,SYSDATE,TO_CHAR(SYSDATE,'HH:MI:SS));
```

```
END;
/
```

157.Write a database trigger display the message when the inserting hiredate is greater than system date

```
CREATE OR REPLACE TRIGGER HIREDATE_OVER
 AFTER INSERT ON EMP
 FOR EACH ROW
 BEGIN
    IF :NEW.HIREDATE > SYSDATE THEN
            RAISE_APPLICATION_ERROR(-20009,'INVALID HIREDATE.....');
    END IF;
 END;
/
```

158.Write a database trigger halt the transaction of EMP table if the deptno is does not exist in the dept table

```
CREATE OR REPLACE TRIGGER DEPT_NO
BEFORE INSERT OR UPDATE OR DELETE ON EMP
FOR EACH ROW
DECLARE
      DNO NUMBER:=0;
BEGIN
      SELECT COUNT(*) INTO DNO FROM DEPT WHERE DEPTNO=:NEW.DEPTNO;
      DBMS_OUTPUT.PUT_LINE(DNO);
      IF DNO=0 THEN
            RAISE_APPLICATION_ERROR(-20009,'DEPTNO NOT EXIST IN DEPT
TABLE....');
      END IF;
END;
/
```

159.Write a database trigger add Rs 500 if the inserting salary is less than Rs 1000

```
CREATE OR REPLACE TRIGGER SAL_ADD
BEFORE INSERT ON EMP
FOR EACH ROW
BEGIN
      IF :NEW.SAL <= 1000 THEN
            :NEW.SAL:=:NEW.SAL+500;
      END IF;
END;
/
```

160.Write a database trigger give the appropriate message if the record exceed more than 100 on EMP table

```
CREATE OR REPLACE TRIGGER EMP_OVER_REC
AFTER INSERT ON EMP
DECLARE
      R NUMBER;
BEGIN
      SELECT COUNT(*) INTO R FROM EMP;
      IF R>=100 THEN
```

```
            RAISE_APPLICATION_ERROR(-20009,'100 RECORD ALLOWED IN EMP
TABLE.....');
      END IF;
END;
/
```

161.Write a program to month and year and display the Calendar of that month.

```
DECLARE
      D NUMBER:=1;
      M VARCHAR2(10):='&MONTH';
      Y NUMBER:=&YEAR;
      C CHAR(20);
      V VARCHAR2(500);
      N NUMBER;
BEGIN
      N:=TO_CHAR(LAST_DAY(D||'-'||M||'-'||Y),'DD');
      C:= TO_CHAR(TO_DATE(D||'-'||M||'-'||Y),'DY');
      dbms_output.put_line('*********************************');
      dbms_output.put_line('*         '||M||'-'||Y||'        *');
      dbms_output.put_line('*SUN MON TUE WED THU FRI SAT *');
      dbms_output.put_line('*********************************');
      IF C='MON' THEN
                  V:='  ';
            ELSIF C='TUE' THEN
            V:='    ';
             ELSIF C='WED' THEN
            V:='      ';
             ELSIF C='THU' THEN
            V:='        ';
             ELSIF C='FRI' THEN
            V:='          ';
             ELSIF C='SAT' THEN
            V:='            ';
      END IF;
FOR I IN 1..N
LOOP
      V:=V||LPAD(I,4);
      IF LENGTH(V)=28 THEN
      dbms_output.put_line(LPAD(V,29,'*')||' *');
      V:=NULL;
      END IF;
END LOOP;
dbms_output.put_line('*'||RPAD(V,29)||'*');
END;
/
```