Name : Abhishek Kumar
Roll number: 2005982
Subject : DBMS Lab Assignment 9.
(on Topic: PL/SQL)

Assignment 1:

```
SQL> connect
Enter user-name: Abhishek
Enter password:200982
Connected.
ASS-1
QU.1---
SQL> DECLARE
  2  AREA NUMBER(5,2);
  3  PI CONSTANT NUMBER(4,2):=3.14;
  4  BEGIN
  5  DECLARE
  6  RADIUS NUMBER(5);
  7  BEGIN
  8  RADIUS:=&RADIUS;
  9  AREA:=PI*POWER(RADIUS,2);
 10  DBMS_OUTPUT.PUT_LINE('THE AREA OF RADIUS '||RADIUS||' IS
'||AREA);
 11  END;
 12  END;
 13  /
```

```
Enter value for radius: 5
old   8: RADIUS:=&RADIUS;
new   8: RADIUS:=5;
THE AREA OF RADIUS 5 IS 78.5

PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.

QU.2---
SQL> DECLARE
  2  VAR1 NUMBER(5);
  3  SQUARE NUMBER(7);
  4  CUBE NUMBER(7);
  5  DOUBL NUMBER(7);
  6  BEGIN
  7  VAR1 :=&VAR1;
  8  SQUARE:=POWER(VAR1,2);
  9  DBMS_OUTPUT.PUT_LINE('THE SQAURE OF A NUMBER IS
'||SQUARE);
```

```
 10  CUBE:=POWER(VAR1,3);
 11  DBMS_OUTPUT.PUT_LINE('THE CUBE OF A NUMBER IS '||CUBE);
 12  DOUBL:=VAR1*2;
 13  DBMS_OUTPUT.PUT_LINE('THE DOUBLE OF A NUMBER IS '||DOUBL);
 14  END;
 15  /
```

```
Enter value for var1: 5
old   7: VAR1 :=&VAR1;
new   7: VAR1 :=5;
THE SQAURE OF A NUMBER IS 25
THE CUBE OF A NUMBER IS 125
THE DOUBLE OF A NUMBER IS 10
```

PL/SQL procedure successfully completed.

QU.3---

```
SQL> DECLARE
  2  A NUMBER(5) ;
  3  B NUMBER(5);
  4  TEMP NUMBER(5) ;
  5  BEGIN
  6  A :=&A;
  7  B :=&B;
  8  DBMS_OUTPUT.PUT_LINE('THE VALUES BEFORE SWAPPING OF A AND
B IS '||A||' '||B) ;
  9  TEMP :=A;
 10  A :=B;
 11  B :=TEMP;
 12  DBMS_OUTPUT.PUT_LINE('THE VALUES AFTER THE SWAP IS
'||A||' '||B) ;
 13  END;
 14  /
```

```
Enter value for a: 6
old   6: A :=&A;
new   6: A :=6;
Enter value for b: 3
old   7: B :=&B;
new   7: B :=3;
THE VALUES BEFORE SWAPPING OF A AND B IS 6 3
THE VALUES AFTER THE SWAP IS 3 6

PL/SQL procedure successfully completed.
```

QU.4--

```
SQL> DECLARE
  2  HRS NUMBER(2) ;
  3  RATE NUMBER(2) ;
  4  GROSS NUMBER(5,2) ;
  5  NETPAY NUMBER(5,2) ;
  6  BEGIN
  7  HRS :=&HRS;
```

```
  8  RATE :=&RATE;
  9  GROSS:=HRS*RATE;
 10  NETPAY:=GROSS-(GROSS*28/100) ;
 11  DBMS_OUTPUT.PUT_LINE('THE GROSSPAY IS '||GROSS) ;
 12  DBMS_OUTPUT.PUT_LINE('THE NETPAY IS '||NETPAY) ;
 13  END;
 14  /
```

```
Enter value for hrs: 5
old   7: HRS :=&HRS;
new   7: HRS :=5;
Enter value for rate: 4
old   8: RATE :=&RATE;
new   8: RATE :=4;
THE GROSSPAY IS 20
THE NETPAY IS 14.4

PL/SQL procedure successfully completed.
```

QU.5----

```
SQL> DECLARE
  2  FIRST VARCHAR(30) ;
  3  LAST VARCHAR(30) ;
  4  BEGIN
  5  FIRST :=&FIRST;
  6  LAST :=&LAST;
  7  DBMS_OUTPUT.PUT_LINE('THE FULL NAME IS '|| FIRST||LAST||'
THE NAME SEPERATED BY COMMA IS '||FIRST ||', '||LAST) ;
  8  END;
  9  /
```

```
Enter value for first: 'Abhishek'
old   5: FIRST :=&FIRST;
new   5: FIRST :='Abhishek';
Enter value for last: 'Kumar'
old   6: LAST :=&LAST;
new   6: LAST :='Kumar';

PL/SQL procedure successfully completed.
```

QU.6---

```
SQL> DECLARE
  2  NUM NUMBER(6) ;
  3  BEGIN
  4  NUM :=&NUM;
  5  IF MOD(NUM, 2)=0
  6  THEN
  7  DBMS_OUTPUT.PUT_LINE('THE NUMBER IS '||NUM||' IS EVEN') ;
  8  ELSE
  9  DBMS_OUTPUT.PUT_LINE('THE NUMBER IS '||NUM||'ODD') ;
 10  END IF;
 11  END;
 12  /
```

```
SQL> /
Enter value for num: 6
old   4: NUM :=&NUM;
new   4: NUM :=6;
THE NUMBER IS 6 IS EVEN

PL/SQL procedure successfully completed.

SQL> /
Enter value for num: 9
old   4: NUM :=&NUM;
new   4: NUM :=9;
THE NUMBER IS 9ODD

PL/SQL procedure successfully completed.
```

QU.7---

```
SQL> DECLARE
  2  A NUMBER(7) ;
  3  B NUMBER(7) ;
  4  C NUMBER(7) ;
  5  BEGIN
  6  A :=&A;
  7  B :=&B;
  8  C :=&C;
  9  IF A>B
 10  THEN
 11  IF A>C
 12  THEN
 13  DBMS_OUTPUT.PUT_LINE('A IS GREATEST') ;
 14  ELSE
 15  DBMS_OUTPUT.PUT_LINE('C IS GREATEST') ;
 16  END IF;
 17  ELSE IF B>C
 18  THEN
 19  DBMS_OUTPUT.PUT_LINE(' B IS GREATEST') ;
 20  ELSE
 21  DBMS_OUTPUT.PUT_LINE(' C IS GREATEST') ;
 22  END IF;
 23  END IF;
 24  END;
 25  /
```

```
Enter value for a: 4
old   6: A :=&A;
new   6: A :=4;
Enter value for b: 9
old   7: B :=&B;
new   7: B :=9;
Enter value for c: 1
old   8: C :=&C;
new   8: C :=1;
B IS GREATEST

PL/SQL procedure successfully completed.
```

QU.8---

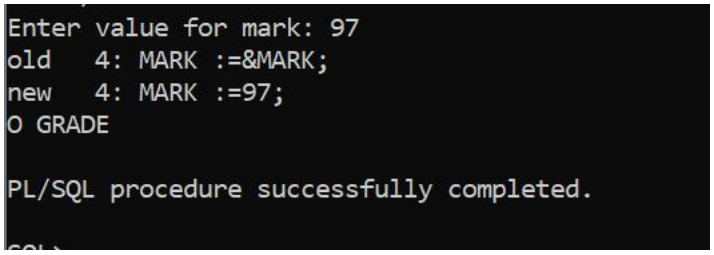
```
SQL> DECLARE
  2  MO NUMBER(7) ;
  3  BEGIN
  4  MO :=&MO;
  5  CASE MO
  6  WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('JANUARY') ;
  7  WHEN 2 THEN DBMS_OUTPUT.PUT_LINE('FEBRUARY') ;
  8  WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('MARCH') ;
  9  WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('APRIL') ;
 10  WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('MAY') ;
 11  WHEN 6 THEN DBMS_OUTPUT.PUT_LINE('JUNE') ;
 12  WHEN 7 THEN DBMS_OUTPUT.PUT_LINE('JULY') ;
 13  WHEN 8 THEN DBMS_OUTPUT.PUT_LINE('AUGUST') ;
 14  WHEN 9 THEN DBMS_OUTPUT.PUT_LINE('SEPTEMBER') ;
 15  WHEN 10 THEN DBMS_OUTPUT.PUT_LINE('OCTOBER') ;
 16  WHEN 11 THEN DBMS_OUTPUT.PUT_LINE('NOVEMBER') ;
 17  WHEN 12 THEN DBMS_OUTPUT.PUT_LINE('DECEMBER') ;
 18  END CASE;
 19  END;
 20  /
```

```
Enter value for mo: 6
old   4: MO :=&MO;
new   4: MO :=6;
JUNE

PL/SQL procedure successfully completed.
```

QU.9----

```
SQL> DECLARE
  2  MARK NUMBER (5) ;
  3  BEGIN
  4  MARK :=&MARK;
  5  IF MARK>90 AND MARK<=100
  6  THEN
  7  DBMS_OUTPUT.PUT_LINE('O GRADE') ;
```

```
  8  ELSE IF MARK >80 AND MARK<=90
  9  THEN
 10  DBMS_OUTPUT.PUT_LINE('A GRADE') ;
 11  ELSE IF MARK>70 AND MARK<=80
 12  THEN
 13  DBMS_OUTPUT.PUT_LINE('B GRADE') ;
 14  ELSE IF MARK>60 AND MARK<=70
 15  THEN
 16  DBMS_OUTPUT.PUT_LINE('C GRADE') ;
 17  ELSE
 18  DBMS_OUTPUT.PUT_LINE('DGRADE');
 19  END IF;
 20  END IF;
 21  END IF;
 22  END IF;
 23  END;
 24  /
Enter value for mark: 97
old   4: MARK :=&MARK;
new   4: MARK :=97;
O GRADE

PL/SQL procedure successfully completed.
```



Assignment 2:

```
SQL> connect

Enter user-name: Abhishek

Enter password:2005982

Connected.

SQL> SET SERVEROUTPUT ON;

SQL> SET DEFINE ON;

ASSIGNMENT_2
```

QU1---

```
SQL> DECLARE
  2  YEAR NUMBER(7) ;
  3  MO NUMBER(7);
  4  BEGIN
  5  YEAR:=&YEAR;
  6  MO:=&MO;
  7  CASE MO
  8  WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('JANUARY-31') ;
  9  WHEN 2 THEN
 10  IF MOD(YEAR,4)=0 AND MOD(YEAR,100)!=0
 11  THEN
 12  DBMS_OUTPUT.PUT_LINE(' LEAP FEBRUARY-29');
 13  ELSE IF  MOD(YEAR,400)=0
 14  THEN
 15  DBMS_OUTPUT.PUT_LINE('LEAP FEBRUARY-29');
 16  ELSE
 17  DBMS_OUTPUT.PUT_LINE('NOT LEAP FEBRUARY-28');
 18  END IF;
 19  END IF;
 20  WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('MARCH-31') ;
 21  WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('APRIL-30') ;
 22  WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('MAY-31') ;
 23  WHEN 6 THEN DBMS_OUTPUT.PUT_LINE('JUNE-30') ;
 24  WHEN 7 THEN DBMS_OUTPUT.PUT_LINE('JULY-31') ;
 25  WHEN 8 THEN DBMS_OUTPUT.PUT_LINE('AUGUST-31') ;
 26  WHEN 9 THEN DBMS_OUTPUT.PUT_LINE('SEPTEMBER-30') ;
 27  WHEN 10 THEN DBMS_OUTPUT.PUT_LINE('OCTOBER-31') ;
 28  WHEN 11 THEN DBMS_OUTPUT.PUT_LINE('NOVEMBER-30') ;
```

```
 29  WHEN 12 THEN DBMS_OUTPUT.PUT_LINE('DECEMBER-31') ;
 30  END CASE;
 31  END;
 32  /
```

```
Enter value for year: 2001
old   5: YEAR:=&YEAR;
new   5: YEAR:=2001;
Enter value for mo: 12
old   6: MO:=&MO;
new   6: MO:=12;
DECEMBER-31

PL/SQL procedure successfully completed.

SQL> /
Enter value for year: 2001
old   5: YEAR:=&YEAR;
new   5: YEAR:=2001;
Enter value for mo: 2
old   6: MO:=&MO;
new   6: MO:=2;
NOT LEAP FEBRUARY-28

PL/SQL procedure successfully completed.
```

QU.2---

```
SQL> DECLARE
  2  NUM NUMBER(7) ;
  3  B NUMBER(7);
  4  REV NUMBER(7) ;
  5  DIV CONSTANT NUMBER(2):=10;
  6  BEGIN
  7  NUM  :=&NUM;
  8  REV:=0;
  9  WHILE NUM!=0
 10  LOOP
```

```
 11  B:=MOD(NUM,10);
 12  REV:=REV*10+B;
 13  NUM:=NUM/10 ;
 14  END LOOP;
 15  DBMS_OUTPUT.PUT_LINE('REVERSE OF A NUMBER IS ' ||REV) ;
 16  END;
 17  /
```

```
SQL> /
Enter value for num: 112
old   7: NUM  :=&NUM;
new   7: NUM  :=112;
REVERSE OF A NUMBER IS 211

PL/SQL procedure successfully completed.
```

QU.3--

```
SQL> DECLARE
  2  S1 NUMBER(7) ;
  3  I  NUMBER(7);
  4  J NUMBER(7);
  5  BEGIN
  6  S1 :=&S1;
  7  FOR I IN 1..S1
  8  LOOP
  9  FOR J IN 1..I
 10  LOOP
 11  DBMS_OUTPUT.PUT('*') ;
 12  END LOOP;
 13  DBMS_OUTPUT.NEW_LINE;
 14  END LOOP;
 15  END;
 16  /
```

```
Enter value for s1: 5
old   6: S1 :=&S1;
new   6: S1 :=5;
*
**
***
****
*****

PL/SQL procedure successfully completed.
```

Q.4---

```
SQL> DECLARE
  2  RADI NUMBER(5);
  3  AREA NUMBER(5,2);
  4  PI CONSTANT NUMBER(5,2):=3.14;
  5  BEGIN
  6  FOR RADI IN 3..7
  7  LOOP
  8  AREA:=PI*POWER(RADI,2);
  9  INSERT INTO AREAS(RADIUS,AREA) VALUES(RADI,AREA);
 10  END LOOP;
 11  END;
 12  /
```

```
SQL> SELECT *FROM AREAS;

    RADIUS       AREA
---------- ----------
         3      28.26
         4      50.24
         5       78.5
         6     113.04
         7     153.86
```

Q.5--

```
SQL> CREATE TABLE ITEM(ITEMNUM NUMBER(7)) ;
```

```
Table created.


SQL> DECLARE
  2  ITEMNO NUMBER(7) ;
  3  BEGIN
  4  ITEMNO :=&ITEMNO;
  5  FOR ITEMNO IN 1..5
  6  LOOP
  7  INSERT INTO ITEM(ITEMNUM) VALUES(ITEMNO) ;
  8  END LOOP;
  9  END;
 10  /
Enter value for itemno: 1
old   4: ITEMNO :=&ITEMNO;
new   4: ITEMNO :=1;


PL/SQL procedure successfully completed.
```

QU.6---

```
SQL> DECLARE
  2  DELITEM NUMBER(7) ;
  3  BEGIN
  4  DELITEM :=&DELITEM;
  5  DELETE  FROM ITEM WHERE ITEMNUM=DELITEM;
  6  END;
```

```
  7  /
Enter value for delitem: 4
old   4: DELITEM :=&DELITEM;
new   4: DELITEM :=4;

PL/SQL procedure successfully completed.

SQL> SELECT *fROM ITEM;

   ITEMNUM
----------
         1
         2
         3
         5
```

QU.7---

```
SQL> DECLARE
  2  SID STUDENT_417.STUDENTID%TYPE;
  3  NAME STUDENT_417.FIRST%TYPE;
  4  PH STUDENT_417.PHONE%TYPE;
  5  CITE STUDENT_417.CITY%TYPE;
  6  BEGIN
  7  SID :=&SID;
  8  SELECT FIRST,CITY,PHONE,STUDENTID INTO NAME,CITE,PH,SID FROM
STUDENT_417 WHERE STUDENTID=SID;
  9  DBMS_OUTPUT.PUT('THE NAME IS '||NAME);
 10  DBMS_OUTPUT.NEW_LINE;
 11  DBMS_OUTPUT.PUT('THE PHONE NUMBER IS '||PH);
 12  DBMS_OUTPUT.NEW_LINE;
 13  DBMS_OUTPUT.PUT('THE CITY IS '||CITE);
 14  DBMS_OUTPUT.NEW_LINE;
 15  END;
 16  /
```

```
Enter value for sid: '00104'
old   7: SID :=&SID;
new   7: SID :='00104';
THE NAME IS BRIAN
THE PHONE NUMBER IS 2125555555
THE CITY IS HOPE

PL/SQL procedure successfully completed.
```

QU.8--

```
SQL> DECLARE
  2  RLY NUMBER(4) ;
  3  RLM NUMBER(2) ;
  4  RLD NUMBER(2) ;
  5  RTY NUMBER(4) ;
  6  RTM NUMBER(2) ;
  7  RTD NUMBER(2) ;
  8  BEGIN
  9  RLY :=&RLY;
 10  RLM :=&RLM;
 11  RLD :=&RLD;
 12  IF RLM=1 OR RLM=3 OR RLM=5 OR RLM=7 OR RLM=8 OR RLM=10
 13  THEN
 14  IF RLD>=1 AND RLD<=28
 15  THEN
 16  RTD:=RLD+3;
 17  RTM :=RLM;
 18  RTY :=RLY;
 19  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;
 20  ELSE
 21  RTD:=RLD-31+3;
 22  RTM:=RTM+1;
```

```
 23  RTY:=RLY;

 24  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 25  END IF;

 26   ELSE IF RLM=4 OR RLM=6 OR RLM=9 OR RLM=11

 27  THEN

 28  IF RLD>=1 AND RLD<=27

 29  THEN

 30  RTD:=RLD+3;

 31  RTM :=RLM;

 32  RTY :=RLY;

 33  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 34  ELSE

 35  RTD:=RLD-30+3;

 36  RTM:=RTM+1;

 37  RTY:=RLY;

 38  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 39  END IF;

 40  ELSE IF RLM=2

 41  THEN

 42  IF MOD(RLY,4)=0 AND MOD(RLY,100)!=0

 43  THEN

 44  --DBMS_OUTPUT.PUT_LINE(' LEAP FEBRUARY-29');

 45  IF RLD>=27 AND RLD<=29

 46  THEN

 47  RTD :=RLD-30+3;

 48  RTM :=RLM+1;

 49  RTY :=RLY;
```

```
 50  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 51  ELSE

 52  RTD :=RLD+3;

 53  RTM:=RLM;

 54  RTY:=RLY;

 55  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 56  END IF;

 57  ELSE IF  MOD(RLY,400)=0

 58  THEN

 59  --DBMS_OUTPUT.PUT_LINE('LEAP FEBRUARY-29');

 60  IF RLD>=27 AND RLD<=29

 61  THEN

 62  RTD :=RLD-30+3;

 63  RTM :=RLM+1;

 64  RTY :=RLY;

 65  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 66  ELSE IF RLD<27

 67  THEN

 68  RTD :=RLD+3;

 69  RTM:=RLM;

 70  RTY:=RLY;

 71  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 72  END IF;

 73  ELSE

 74  DBMS_OUTPUT.PUT_LINE('NOT LEAP FEBRUARY-28');

 75  IF RLD>=27 AND RLD<=28
```

```
 76  THEN

 77  RTD:=RLD-29+3;

 78  RTM :=RLM+1;

 79  RTY :=RLY;

 80  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 81  ELSE

 82  RTD :=RLD+3;

 83  RTM:=RLM;

 84  RTY:=RLY;

 85  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 86  END IF;

 87  END IF;

 88  IF RLM=12

 89  THEN

 90  IF RLD>=29 AND RLD<=31

 91  THEN

 92  RTD :=RLD-31+3;

 93  RTM:=1;

 94  RTY:=RLY+1;

 95  INSERT INTO
VIDEOSTORE(RENTALDATE,RENTALMONTH,RENTALYEAR,RETURNDATE, RETURNMONTH,
RETURNYEAR) VALUES(RLD,RLM,RLY,RTD, RTM, RTY) ;

 96  ELSE

 97  RTD:=RLD+3;

 98  RTM:=RLM;

 99  RTY:=RLY;

100  END IF;

101  END IF;

102  END IF;
```

```
103  END IF;
104  END;
```

QU.9--

```
SQL> DECLARE
  2  RID LOCATION_417.ROOMID%TYPE ;
  3  CAP NUMBER(2);
  4  BEGIN
  5  RID :=&RID;
  6  SELECT CAPACITY INTO CAP FROM LOCATION_417 WHERE ROOMID=RID;
  7  IF CAP<50
  8  THEN
  9  UPDATE LOCATION_417 SET CAPACITY=50
 10  WHERE ROOMID=RID;
 11  ELSE
 12  DBMS_OUTPUT.PUT_LINE('THE ROOMID YOU ENTERED IS CAPACITY
GREATER OR EQUAL TO 50');
 13  END IF;
 14  END;
 15  /
Enter value for rid: 14
old   5: RID :=&RID;
new   5: RID :=14;
```

```
SQL> SELECT *fROM LOCATION_417;

    ROOMID BUILDIN ROO   CAPACITY R
---------- ------- --- ---------- -
        11 GANDHI  101          2 O
        12 GANDHI  103          2 O
        13 KENNEDY 202         35 L
        14 KENNEDY 204         50 L
        15 NEHRU   301         50 L
        16 NEHRU   309         45 C
        17 GANDHI  105          2 O
        18 KENNEDY 206         40 L
        20 GANDHI  107          2 O
        21 GANDHI  109          2 O
        22 XYH     145
```
