Home assignment #4 (352 &452),
Due date 5/20 5:30PM
(points = 100)

Q1. (10 points)
Define an associative array type, name it tp1. The data type of elements should be char(1), index by binary_integer. Then define a variable of this TYPE tp1, populate this associative array with capital letters from A to F (total 6 letters). Print out the contents of this associative array. (This question is not related to any database table).

Hints:
There are several ways to finish this question.

a) A simple way, you can populate the associative array by literally typing:
```
v_aa (1) := 'A' ;
v_aa (2) := 'B' ;
v_aa (3) := 'C' ;
. . .
v_aa (6) := 'F' ;
```

b) For more general cases, if it is from A to Z, then it is not efficient. You may recall that usually in other languages, x := `('A')+1)` would assign value `'B'to x.` But that does not work here.

You may use function `CHR(ASCII('A')+1)` to get the next letter (‘`B`’).
Say, the program initializes x := `'A'`, then later it increases the variable (x), such as
```
  x := CHR(ASCII(x)+1).
```

The SQL statements below are for your reference:

```
SELECT TO_CHAR(ASCII('A'))      FROM DUAL;
SELECT CHR(ASCII('A')) as n     FROM DUAL;
SELECT CHR(ASCII('A')+1) as n   FROM DUAL;
```

Q2. (20 points) (AA, index by string)
Use table of Employees for this question.
In an anonymous PL/SQL block, define an associative array type called Lname_salary that uses the last_name as its index, the values of elements will be salary.
You will use the first 12 employees info in department (ID) 50 to populate the associative array (variable).
Your program will printout the contents of the variable of this AA two times: one time in the order of populating, second time in the order of the index (last name).

Note, the following select statement has guaranteed that the last name is unique in department 50.
```
select last_name,  count (*) from employees where department_ID  = 50
group by last_name having count (*) > 1
RESULT:
no rows selected
```

Q3. (25 points) (Varray)
Use table of Employees for this question.
In an anonymous PL/SQL block, define a Varray type of size 15, called Lname, that varray will use last_names as its elements. Define v1 as its variable.

(1) Use the first 12 employees last names in department (ID) 50 to populate this varray variable v1, populating it in order by employee_id (using " order by employee_id " in your cursor definition), print out the contents of this Varray in the population loop.
(2) Display limit of v1.
(3) Display the value of last index.
(4) Add two elements to the end of v1, the new last names are Washington, Lincoln.
(5) Display the limit of v1 and the value of last index again (after step 5).
(6) Delete all the elements in v1.
(7) Display the limit of v1 and the value of last index again (after step 6).

Q4. (25 points) (NT)
Use table of Employees for this question.
In an anonymous PL/SQL block, define a nested table, called Lname, that Nested table will use last_names as its elements. Define v1 as its variable.

(1) Use the first 12 employees last names in department (ID) 50 to populate this nested table, populating it in order by employee_id (using " order by employee_id " in your cursor definition), print out the contents of this Nested table in the population loop.
(2) Display limit of v1. (it will output null)
(3) Display the value of last index.
(4) Add two elements to the end of v1, the new last names are Washington, Lincoln.
(5) Using for loop, print out all the elements of v1 (as result after step 4),
(6) Display the current size (count) of v1 and the value of last index again,
(7) Delete the elements from 5 to 7 (including 5 and 7),
(8) Display the current size (count) of v1 and the value of last index (as result after step 7),
(9) Print out the existing elements of the nested table; be careful, the NT is sparse now.

Q5. (10 points) (AA)
Use table of Employees for this question.
In an anonymous PL/SQL block, define an associative array, called Lname, that Associative array will use last_names as its elements, index by PLS_INTEGER. Define v1 as its variable.
You will use the first 12 employees last names in department (ID) 50 to populate this variable v1, populating it in order by employee_id (using this order by in your cursor definition).
Your program will printout the contents of this v1.

Q6. (10 points) Exercise on Varray of Varray, NT of NT
Although a collection has only one dimension, you can model a multidimensional collection with a collection whose elements are collections.
The program below defines a varray of integer VA1, then defines a Varray of that Varray VA1.
Then this program populates, operates on their variables. You can run it, it should work.

In this question, you are required to do:

(1) based on the program provided, you define a NT (nested table) of elements of integer, then defined a NT of that NT. Your program will be very similar to the code below. Assuming we use the same variable names as below for the NT, NT of NT.
(2) Initialize the NT variable x1 with values of (4, 5, 6).
Initialize variable nY1 with three elements (x1, x1, x1).
(3) Print out the value of 2nd element's 3rd element of nY1. (nY1 (2) (3))
(4) In executable section, add 4th element for nY1 with value (11, 22, 33, 44, 55, 66)
Print out the value of nY1 element 4's element 3.

```
DECLARE
   TYPE VA1 IS VARRAY(10) OF INTEGER;     -- varray of integer
     X1 VA1 := VA1(1,2,3);
   TYPE nVA1 IS VARRAY(10) OF VA1;      -- varray of varray of integer
      nY1  nVA1 := nVA1 (X1, VA1(11,12,13), VA1(21,22,23), X1);
   i    INTEGER;

BEGIN
    i := nY1(2)(3);               -- value "13", two dimensional indexes
    DBMS_OUTPUT.PUT_LINE(' i = ' || i);

    nY1.EXTEND;                  -- add one more element. index = 5
    nY1(5) := VA1(51,52,53, 54,55);  -- Assign values to element #5
    nY1(4) := VA1(41,42,43,44);    -- was X1, now change
    nY1(4)(4) := 88;                -- replace 88 with 44

    nY1(4).EXTEND;           -- add element to 4th varray element
    nY1(4)(5) := 45;           -- store integer 45 there
       DBMS_OUTPUT.PUT_LINE(' nY1(5)(5) = ' || nY1 (5)(5));
END;
/
```

Hints:
Q3. & Q4.
If there is trouble, very possible that your program does not use constructor and extend.

Q5. If you have time and interests, you may exercise some methods to this AA.

Q6. If you have time and interests, you may exercise operations on this multidimensional collection.