Create user:

```
Create user helen identified by userpass;

GRANT DBA to helen;
```

```
--NẾU BÁO LỖI thì thêm c##trước helen

Create user c##helen identified by userpass;

GRANT DBA to c##helen;
```

Using StudentSchema\createStudent.sql to create student database.

Do the following questions:

1. Write a pl/sql block to assign value to a variable and print out the value.
2. Write a pl/sql block to check number is Odd or Even.
3. Write a pl/sql block to check a number is greater or lower than 0.
4. Using database Student Management, write a PL/SQL block to check how many students are enrolled in section id 85. If 15 or more students are enrolled, section 85 is full. Otherwise, section 85 is not full. In both cases, a message should be displayed to the user, indicating whether section 85 is full.
5. Do question 4 again using a procedure with 1 parameter: section number. Write a PL/SQL block to call the procedure with parameters section 85.
6. Use a numeric FOR loop to calculate a factorial of 10 (10! = 1*2*3...*10). Write a PL/SQL block.
7. Write a procedure to calculate the factorial of a number. Write a PL/SQL block to call this procedure.
8. Write a function to calculate the factorial of a number. Write a PL/SQL block to call this function.

9. Write a procedure to calculate and print out the result of a division. If the denominator is 0 then adding exception. Write a PL/SQL block to call this procedure.

   ```
   EXCEPTION

   WHEN ZERO_DIVIDE THEN

   DBMS_OUTPUT.PUT_LINE ('A number cannot be divided by zero.');
   ```

10. Write a function to calculate the result of a division. Write a PL/SQL block to call this function.

11. Using database Student Management, write a procedure to display the student's name, street_address on the screen. If no record in the STUDENT table corresponds to the value of student_id provided by the user, the exception NO_DATA_FOUND is raised. Write a PL/SQL block to call this procedure with parameter is 25, 105.

12. Do the question 11 using function to return the student's record (declare a rowtype variable). Write a PL/SQL block to call this function and print out student's name, address, phone.

13. Write a procedure to check if the student is enrolled. If no record in the ENROLLMENT table corresponds to the value of v_student_id provided by the user, the exception NO_DATA_FOUND is raised. And if more than one record in the ENROLLMENT table then exception TOO_MANY_ROWS is raised.

    Write a pl/sql block to call procedure above with different values of student ID: 102, 103, 319

14. Write a procedure to find full name of the instructor corresponds to the value of instructor_id provided by the user.

Write a pl/sql block to call procedure above with different values of instructor_id: 107, 120

15. Write a function to return full name of the instructor corresponds to the value of instructor_id provided by the user. Write a PL/SQL block to call this function.

16. Write a procedure to find name of a student and how many courses this user enrolled. Print out the result on the screen. Write a PL/SQL block to call this function.

17. Write a function to calculate how many courses that student enrolled. Student_id provided by the user.

    Write a procedure to find name of student and how many courses this user enrolled.

    Write a pl/sql block to call procedure above with different values of student_id: 109, 530

18. Write a procedure to calculate how many students are registered for a given section.If a section has more than 10 students enrolled in it, an error message is displayed, indicating that this course has too many students enrolled (Using RAISE_APPLICATION_ERROR).

19. Write a pl/sql block to call procedure above with parameter

    a. sectionid 89.

    b. sectionid 155.