```
1)
a)
create or replace function add1(a integer)
returns integer as
        $$
        begin
                return a+1;
        end;
        $$
language plpgsql;
select add1(9);
b)
create or replace function sum(a integer,b integer)
returns integer as
        $$
        begin
                return a+b;
        end;
        $$
language plpgsql;
select sum(4,5);
c)
create or replace function checkNum(a integer)
returns bool as
        $$
        begin
                if (a%2=0) then
                        return true;
                else return false;
                end if;
        end;
        $$
language plpgsql;
select checkNum(3);
d)
create or replace function checkPassword(a varchar, password varchar default 'N')
returns bool as
        $$
        begin
                if (a = password) then
                        return true;
                else return false;
                end if;
        end;
        $$
language plpgsql;
select checkPassword('N','N');
e)
create or replace function print(inout a varchar, out valid varchar)
as
$$
        declare password varchar = 'Nurs';
begin
        if checkPassword(a,password) = true then
                valid = 'valid';
        else
                valid = 'not valid';
        end if;
        raise notice '% password is %', password, valid;
end
$$
        language plpgsql;
select print('urs');
```

```
2)
a)
create table table1
(
        id serial primary key,
        name varchar,
        date   date,
        age integer
);
create table check
(
        id serial primary key,
        time_action timestamp
);
create or replace function checkTimestamp()
        returns trigger
        language plpgsql
as
$$
begin
        insert into check values (now());
        raise notice 'timestamp is %', now();
        return new;
end;
$$;
create trigger timestampTrigger
        after insert or update or delete
        on table1
        for each row
execute procedure checkTimestamp();
insert into table1
values('name', '31-12-2003');
update table1
set name = 'Nurs'
where id = 1;
b)
create or replace function calculateAge()
        returns trigger
as
$$
begin
        new.age = date_part('year', age(current_date, new.date));
        return new;
end;
$$
language plpgsql;
create trigger ageTrigger
        before insert
        on table1
        for each row
execute procedure calculateAge();
insert into table1
VALUES ('name5', '2001-10-12');
c)
create table table2
(
        id    serial primary key,
        name  varchar,
        cost float
);
create or replace function tax()
        returns trigger
as
```

```
$$
begin
        new.cost := cast((new.cost * 1.12) as float);
        return new;
end;
$$
language plpgsql;
create trigger taxTrigger
        before insert
        on table2
        for row
execute procedure tax();
insert into table2
VALUES ('Nurs', 1000);
d)
create or replace function stop()
        returns trigger
        language plpgsql
as
$$
begin
        return null;
end;
$$;
create trigger stopTrigger
        before delete
        on table2
        for each row
execute procedure stop();
delete
from table2
where name = 'Nurs';
d)
create table table3
(
        id       serial primary key,
        name varchar,
        password varchar
);
create trigger superTrigger
        before insert
        on table3
        for each row
execute procedure superFunction();
create or replace function superFunction()
        returns trigger
as
$$
begin
        perform checkPassword(new.password);
        perform print(new.password);
        return new;
end
$$
language plpgsql;
insert into table3
VALUES ('Nurs', '1234');
3)
create table table4
(
        id   serial primary key,
        name varchar,
        salary   integer,
        workExp  integer,
        discount integer,
        age integer
);
a)create or replace procedure incSal()
as
$$
begin
        update table4
        set salary = salary + (salary * (workExp/ 2 * 0.1));
        update table4
        set discount = 10 + (workExp / 5 * 0.1);
end;
$$
language plpgsql;
insert into table4
VALUES ('Nurs', 15000, 3);
b)
create or replace procedure change()
as
$$
begin
        update table4
        set salary = salary * 1.15
        where age >= 40;
        update table4
        set salary = salary * 1.15, discount = 20
        where workExp > 8;
end;
$$
language plpgsql;
insert into table4
VALUES ('name',100000, 9,18);
insert into table4
VALUES ('name',100000, 9,43);
```