```
RESULTAT DES TESTS

========================= test session starts ==============================
platform darwin -- Python 3.6.2, pytest-5.2.2, py-1.8.0, pluggy-0.13.0
rootdir: /Users/emiliendurif/Dropbox/cpge/ipt_mpsi_ds/DS08/correction_auto/scrip
t/DS_08_hatoum
plugins: timeout-1.3.4
timeout: 60.0s
timeout method: signal
timeout func_only: False
collected 22 items

test_TP.py F...F...FF.F.F.F..FFFF                                        [100%]

=================================== FAILURES ===================================
_____ test_Q1_req _____

    def test_Q1_req ():
>       assert requete(sol_Q1_req) == requete(Q1_req)
E       AssertionError: assert [('DUPONT', '...', 'M.'), ...] == [('DUPONT', '..
.nsieur'), ...]
E         At index 0 diff: ('DUPONT', 'Alain', 'M.') != ('DUPONT', 'Alain', 'Mon
sieur')
E         Use -v to get the full diff

test_TP.py:98: AssertionError
_____ test_Q4_req _____

    def test_Q4_req ():
>       assert requete(sol_Q4_req) == requete(Q4_req)
E       AssertionError: assert [('DUHAMEL', ... 'Mme.'), ...] == [('DUHAMEL', ..
.Madame'), ...]
E         At index 0 diff: ('DUHAMEL', 'Evelyne', 'Melle.') != ('DUHAMEL', 'Evel
yne', 'Mademoiselle')
E         Use -v to get the full diff

test_TP.py:110: AssertionError
_____ test_Q7_req _____

    def test_Q7_req ():
>       assert requete(sol_Q7_req) == requete(Q7_req)
E       AssertionError: assert [('DUPONT', '...-92-21'), ...] == [('DUPONT', '..
.-52-50'), ...]
E         At index 2 diff: ('DUBOIS', '02-41-58-89-52') != ('DUPONT', '01-44-28-
52-50')
E         Right contains 48 more items, first extra item: ('AIACH', '04-91-52-51
-52')
E         Use -v to get the full diff

test_TP.py:122: AssertionError
_____ test_Q8_req _____

    def test_Q8_req ():
>       assert requete(sol_Q8_req) == requete(Q8_req)
E       AssertionError: assert [('BENATTAR',), ('MARTIN',)] == [('BENATTAR',...J
ean-Pierre')]
E         At index 0 diff: ('BENATTAR',) != ('BENATTAR', 'Bernard')
E         Use -v to get the full diff

test_TP.py:125: AssertionError
_____ test_Q10_res _____

    def test_Q10_res ():
>       assert sol_Q10_res.upper() ==  Q10_res.upper()
E       AssertionError: assert '(25.0, 59.0)' == '25 59'
E         - (25.0, 59.0)
E         + 25 59
```

```
test_TP.py:131: AssertionError
_____ test_Q11_res _____

    def test_Q11_res ():
>       assert sol_Q11_res.upper() ==  Q11_res.upper()
E       AssertionError: assert '(25.0, 59)' == '25 59'
E         - (25.0, 59)
E         + 25 59

test_TP.py:137: AssertionError
_____ test_Q12_req _____

    def test_Q12_req ():
>       assert requete(sol_Q12_req) == requete(Q12_req)
E       assert [(1,), (2,), ...], (26,), ...] == [(3,), (4,), ...), (73,), ...]
E         At index 0 diff: (1,) != (3,)
E         Left contains 183 more items, first extra item: (1120,)
E         Use -v to get the full diff

test_TP.py:143: AssertionError
_____ test_Q15_res _____

    def test_Q15_res ():
>       assert sol_Q15_res.upper() ==  Q15_res.upper()
E       assert "('SILLET', 'JACQUES', 649)" == 'NOM,PRENOM,MONTANT'
E         - ('SILLET', 'JACQUES', 649)
E         + NOM,PRENOM,MONTANT

test_TP.py:152: AssertionError
_____ test_Q15_req _____

    def test_Q15_req ():
>       assert requete(sol_Q15_req) == requete(Q15_req)
E       AssertionError: assert [('SILLET', 'Jacques', 649)] == []
E         Left contains one more item: ('SILLET', 'Jacques', 649)
E         Use -v to get the full diff

test_TP.py:155: AssertionError
_____ test_Q15_res2 _____

    def test_Q15_res2 ():
>       assert sol_Q15_res2.upper() ==  Q15_res.upper()
E       assert "('THOMASSE',...-CLAUDE', 20)" == 'NOM,PRENOM,MONTANT'
E         - ('THOMASSE', 'JEAN-CLAUDE', 20)
E         + NOM,PRENOM,MONTANT

test_TP.py:158: AssertionError
_____ test_Q15_req2 _____

    def test_Q15_req2 ():
>       assert requete(sol_Q15_req2) == requete(Q15_req)
E       AssertionError: assert [('THOMASSE',...-Claude', 20)] == []
E         Left contains one more item: ('THOMASSE', 'Jean-Claude', 20)
E         Use -v to get the full diff

test_TP.py:161: AssertionError
======================= 11 failed, 11 passed in 0.85s =======================
```

```
NOM = "HATOUM"
Prenom = "RALPH"
Classe = "MPSI2"
alpha="10"

## Question 1
Q1_req = "SELECT CLI_NOM, CLI_PRENOM, TIT_LIBELLE from T_CLIENT, T_titre where t
_client.TIT_CODE = t_titre.TIT_CODE "

## Question 2
Q2_req = "SELECT count (*) from t_client"
Q2_res = "96"

## Question 3
Q3_req = "SELECT CLI_NOM, CLI_PRENOM from T_CLIENT, T_titre where t_client.TIT_C
ODE = t_titre.TIT_CODE and t_titre.TIT_LIBELLE='Madame'"

## Question 4
Q4_req = "SELECT CLI_NOM, CLI_PRENOM, TIT_LIBELLE  from T_CLIENT, T_titre where
t_client.TIT_CODE = t_titre.TIT_CODE and ( t_titre.TIT_LIBELLE='Madame' or t_tit
re.TIT_LIBELLE='Mademoiselle')"


## Question 5
Q5_req = "SELECT count(*) from ( SELECT CLI_NOM, CLI_PRENOM, TIT_LIBELLE  from T
_CLIENT, T_titre where t_client.TIT_CODE = t_titre.TIT_CODE and ( t_titre.TIT_LI
BELLE='Madame' or t_titre.TIT_LIBELLE='Mademoiselle') )"
Q5_res = "17"

## Question 6
Q6_req = "SELECT CLI_NOM as 'Noms', CLI_PRENOM as 'Prénoms' from T_CLIENT, T_tit
re where t_client.TIT_CODE = t_titre.TIT_CODE and ( t_titre.TIT_LIBELLE='Madame'
 or t_titre.TIT_LIBELLE='Mademoiselle') order by CLI_NOM asc "


## Question 7
Q7_req = "SELECT CLI_NOM, TEL_NUMERO from T_CLIENT, T_TELEPHONE where T_TELEPHON
E.CLI_ID = T_CLIENT.CLI_ID"

## Question 8
Q8_req = "SELECT CLI_NOM, CLI_PRENOM  from (SELECT CLI_NOM, CLI_PRENOM, count(*)
 as nb from T_CLIENT group by CLI_NOM) where nb > 1 ;"

## Question 9
Q9_req = "SELECT CLI_NOM, nb  from (SELECT CLI_NOM, CLI_PRENOM, count(*) as nb f
rom T_CLIENT group by CLI_NOM) where nb > 1 ;"
Q9_res = "BENATTAR 2, MARTIN 3"

## Question 10
Q10_req = "SELECT sum(LIF_remise_pourcent)/count(lif_remise_pourcent), sum(lif_r
emise_montant)/count(LIF_REMISE_MONTANT) from T_LIGNE_FACTURE "
Q10_res = "25 59"

## Question 11
Q11_req = "SELECT max(LIF_REMISE_POURCENT), max(LIF_REMISE_MONTANT)  from T_LIGNE
_FACTURE "
Q11_res = "25 59"

## Question 12
Q12_req = "select DISTINCT FAC_ID as fac_id1 from T_LIGNE_FACTURE where (LIF_REM
ISE_MONTANT is not NULL or LIF_REMISE_MONTANT is not NULL)"


## Question 13
Q13_req = "select distinct CLI_ID from T_FACTURE join (select DISTINCT FAC_ID as
 fac_id1 from T_LIGNE_FACTURE where (LIF_REMISE_MONTANT is not NULL or LIF_REMIS
E_MONTANT is not NULL)) on T_FACTURE.FAC_ID = fac_id1"
```

```
## Question 14
Q14_req = ""


## Question 15
Q15_req = ""
Q15_res ="nom,prenom,montant"
```