ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ: ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ
ΕΞΑΜΗΝΟ: ΣΤ’

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ
ΜΙΧΑΗΛ ΣΤΕΦΑΝΙΔΑΚΗΣ

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΡΙΑΣ
ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΛΑΣΟΠΟΥΛΟΥ ΒΑΛΕΡΙΑ
ΑΜ: Π2017062
ΕΞΑΜΗΝΟ ΦΟΙΤΗΣΗΣ: ΣΤ’

ΑΝΑΦΟΡΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΕΡΓΑΣΙΑΣ #2

Για την υλοποίηση του κώδικα αρχικά έκανα **import** το module **re** της Python το οποίο όπως καταλαβαίνουμε και από το όνομα μας παρέχει τη δυνατότητα να χρησιμοποιήσουμε regular expressions.

Στη συνέχεια διάβασα ένα-ένα τα ζητούμενα της εργασίας και τα υλοποίησα με τη σειρά που αναφέρονταν. Πέρα από τη δημιουργία των κανονικών εκφράσεων για την ορθή λειτουργία του κώδικα είναι απαραίτητη η συγγραφή και κάποιων μηχανών ταιριάσματος ώστε να μπορέσουμε να διαβάσουμε και να επεξεργαστούμε το **.txt** αρχείο που μας δίνεται.

Αναλυτικότερα τα βήματα που ακολούθησα για την υλοποίηση του κάθε ζητούμενου είναι:

Ζητούμενο 1

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **(r'<title>(.+?)</title')**

Με την έκφραση αυτή γίνεται η εξαγωγή και η εκτύπωση του τίτλου.

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Το **<tittle>** καθώς και το **</tittle>** απλά μας λένε να ψάξουμε για την συγκεκριμένη ακολουθία μέσα στο κείμενο
3) Η **.** ψάχνει για οποιονδήποτε χαρακτήρα εκτός από new line. Το **+** μας λέει ότι μπορούμε να έχουμε έναν ή περισσότερους χαρακτήρες ενώ το **?** πως το να υπάρχει κάποιος χαρακτήρας δεν είναι απαραίτητο. Οι τρεις αυτοί quantifiers μπήκαν σε () ώστε να θεωρηθούν **group** (στην συγκεκριμένη περίπτωση **group(1)**) το οποίο θα μπορέσουμε εύκολα να εκτυπώσουμε στη συνέχεια.
4) Η εξαγωγή και η εκτύπωση του τίτλου γίνεται με την εντολή:

```
m = rexp1.search(contents)
print(m.group(1))
```

Φυσικά έχει δημιουργηθεί ήδη μεταβλητή **contents** η οποία μας βοηθάει να διαβάσουμε το κείμενο

Ζητούμενο 2

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **(r'<!--(.*?)-->',re.DOTALL)**
Με την έκφραση αυτή γίνεται η απαλοιφή των σχολίων.

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Το **<!--** καθώς και το **-->** απλά μας λένε να ψάξουμε για την συγκεκριμένη ακολουθία μέσα στο κείμενο
3) Η **.** ψάχνει για οποιονδήποτε χαρακτήρα εκτός από new line. Το ***** μας λέει ότι μπορούμε να έχουμε μηδέν ή περισσότερους χαρακτήρες ενώ το **?** πως το να υπάρχει κάποιος χαρακτήρας δεν είναι απαραίτητο. Οι τρεις αυτοί quantifiers μπήκαν σε () ώστε να θεωρηθούν **group** (στην συγκεκριμένη περίπτωση **group(1)**).
4) Το **flag DOTALL** ουσιαστικά προσθέτει και τον χαρακτήρα **/n** στη **.** ώστε η κανονική έκφραση να καλύψει/ελέγξει παραπάνω από μία σειρές.
5) Η απαλοιφή των σχολίων γίνεται με την εντολή:
   contents = rexp2.sub(' ', contents)
   Ουσιαστικά εδώ αντικαθιστούμε τα **< !-- -- >** και οτιδήποτε βρίσκεται μέσα σε αυτά με το κενό.

Ζητούμενο 3

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **(r'<(script|style).*?>.*?</(script|style)>',re.DOTALL)**
Με την έκφραση αυτή γίνεται η απαλοιφή των σχολίων.

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string

2) Το **<(script|style)** καθώς και το **</(script|style)>'** δηλώνουν πως μέσα στο κείμενο ψάχνουμε για οποιαδήποτε ακολουθία περιλαμβάνει το **<script** ή το **<style,** μηδέν ή περισσότερους χαρακτήρες εκτός από new line αν αυτοί υπάρχουν, και το **</script>** ή το **</style>**
3) Η **.** ψάχνει για οποιονδήποτε χαρακτήρα εκτός από new line. Το ***** μας λέει ότι μπορούμε να έχουμε μηδέν ή περισσότερους χαρακτήρες ενώ το **?** πως το να υπάρχει κάποιος χαρακτήρας δεν είναι απαραίτητο. Οι τρεις αυτοί quantifiers μπήκαν σε () ώστε να θεωρηθούν **group** (στην συγκεκριμένη περίπτωση **group(1)**).
4) Το **flag DOTALL** ουσιαστικά προσθέτει και τον χαρακτήρα **/n** στη **.** ώστε η κανονική έκφραση να καλύψει/ελέγξει παραπάνω από μία σειρές.
5) Η απαλοιφή των σχολίων γίνεται με την εντολή:

contents = rexp3.sub(' ', contents)

Ουσιαστικά εδώ αντικαθιστούμε οτιδήποτε καλύπτει η rexp3 με το κενό.

Ζητούμενο 4

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **(r'<a.+?href="(.*?)".*?>(.*?)',re.DOTALL)**
Με την έκφραση αυτή γίνεται εξαγωγή και εκτύπωση του συνδέσμου (ιδιότητα `href`) από `<a>` tags και του κειμένου τους.

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Το **<a** καθώς και το **** δηλώνουν πως μέσα στο κείμενο ψάχνουμε για οποιαδήποτε ακολουθία περιλαμβάνει το **<a** και το **** μέσα στο κείμενο
3)
4) Η **.** ψάχνει για οποιονδήποτε χαρακτήρα εκτός από new line. Το ***** μας λέει ότι μπορούμε να έχουμε μηδέν ή περισσότερους χαρακτήρες ενώ το **?** πως το να υπάρχει κάποιος χαρακτήρας δεν είναι απαραίτητο. Οι τρεις αυτοί quantifiers μπήκαν σε () ώστε να θεωρηθούν **group** (στην συγκεκριμένη περίπτωση **group(1)** και

group(2)), ώστε να μπορούν στη συνέχεια εύκολα να ξεχωρίσουν από το κείμενο και να εκτυπωθούν. Το **+** δηλώνει ότι μπορούμε να έχουμε έναν ή περισσότερους χαρακτήρες.

5) Το **href="(.*?)"** μας δηλώνει πως ψάχνουμε στο κείμενο μετά από το **<a** κτλ τη λέξη **href** η οποία στην συνέχεια ακολουθείτε από τα εξής: **="(.*?)".** Η σημασία των χαρακτήρων στην παρένθεση εξηγείται παραπάνω.
6) Το **flag DOTALL** ουσιαστικά προσθέτει και τον χαρακτήρα **/n** στη **.** ώστε η κανονική έκφραση να καλύψει/ελέγξει παραπάνω από μία σειρές.
7) Η εξαγωγή και εκτύπωση του συνδέσμου (ιδιότητα `href`) από `<a>` tags και του κειμένου τους γίνεται με την εντολή:

   for m in rexp4.finditer(contents):
   print('{} {}'.format(m.group(1),m.group(2)))

   Ουσιαστικά εδώ επιλέγει και εκτυπώνει τα link που βρίσκονται στην κανονική έκφραση **rexp4** (τα link στην κανονική έκφραση βρίσκονται στις ομάδες που δημιουργήθηκαν για αυτό και τις εκτυπώνουμε).

Ζητούμενο 5

Οι ολοκληρωμένες κανονικές εκφράσεις για το συγκεκριμένο ερώτημα είναι οι εξής:

rexp5a = re.compile(r'<.+?>|</.+?>',re.DOTALL)
rexp5b = re.compile(r'<.+?/>',re.DOTALL)

Με τις εκφράσεις αυτές γίνεται απαλοιφή όλων των tags από το κείμενο.

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Η **.** ψάχνει για οποιονδήποτε χαρακτήρα εκτός από new line. Το **+** μας λέει ότι μπορούμε να έχουμε έναν ή περισσότερους χαρακτήρες ενώ το **?** πως το να υπάρχει κάποιος χαρακτήρας δεν είναι απαραίτητο.

3) Η κανονική έκφραση 5α μας λέει πως ψάχνει για tag της μορφής **<something> ...** (όπως εξηγείται παραπάνω) **</something>** ή **</something >**.
4) Ενώ η 5β μας λέει πως ψάχνει για tag της μορφής **< something … />**
5) Το **flag DOTALL** ουσιαστικά προσθέτει και τον χαρακτήρα **/n** στη **.** ώστε η κανονική έκφραση να καλύψει/ελέγξει παραπάνω από μία σειρές.
6) Η απαλοιφή όλων των tags από το κείμενο γίνεται με τις εντολές:

    contents = rexp5a.sub(' ', contents)
    contents = rexp5b.sub(' ', contents)

    Ουσιαστικά εδώ αντικαθιστούμε τα tags με το κενό.

Ζητούμενο 6

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **rexp6 = re.compile(r'&(amp|gt|lt|nbsp);')**
Με την έκφραση αυτή γίνεται η μετατροπή των ειδικών HTML entities που υπάρχουν στο κείμενο σύμφωνα με τον πίνακα που αναφέρεται στην εκφώνηση:

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Η έκφραση αυτή δηλώνει ότι ψάχνει στο κείμενο τις ακολουθίες **&** ή **>** ή **<** ή ** **.

Η μετατροπή των ειδικών HTML entities που υπάρχουν στο κείμενο σύμφωνα με τον πίνακα που αναφέρεται στην εκφώνηση γίνεται με την εντολή:

```
def repl(m):
if(m.group(0)=='&amp;'):
return '&'
elif(m.group(0)=='&gt;'):
```

```
return '>'
elif(m.group(0)=='&lt;'):
return '<'
else:
return ' '
```

Ουσιαστικά εδώ ελέγχουμε όλο το κείμενο και όταν βρίσκουμε κάποια από τις ακολουθίες που αναφέρθηκαν παραπάνω την αντικαθιστούμε με την επιθυμητή που μας ζητάται.

Ζητούμενο 7

Η ολοκληρωμένη κανονική έκφραση για το συγκεκριμένο ερώτημα είναι η εξής: **rexp7 = re.compile(r'\s+')**
Με την έκφραση αυτή γίνεται η μετατροπή ακολουθιών συνεχόμενων χαρακτήρων whitespace σε ένα ακριβώς κενό:

1) Το **r** εδώ δηλώνει πως θέλουμε η έκφραση να αντιμετωπιστεί ως raw string
2) Η έκφραση αυτή δηλώνει ότι ψάχνει στο κείμενο ενός ή παραπάνω χαρακτήρων whitespace.

Η μετατροπή ακολουθιών συνεχόμενων χαρακτήρων whitespace σε ένα ακριβώς κενό γίνεται με την εντολή:

contents = rexp7.sub(' ', contents)

Ουσιαστικά εδώ αντικαθιστούμε τα πολλαπλά whitespace με ένα μόνο κενό.

Ζητούμενο 8

Το κείμενο τυπώνεται με την εντολή: **print(contents)**