

Προγραμματιστική Εργασία #1

```

graph LR
    start(( )) --> q0((q0))
    q0 -- "0, 1" --> q1((q1))
    q0 -- "2" --> q2((q2))
    q1 -- "0-9" --> q3((q3))
    q2 -- "0-3" --> q3
    q3 -- ":" --> q4((q4))
    q4 -- "0-5" --> q5((q5))
    q5 -- "0-9" --> q6(((q6)))
  
```

Περιγραφή Πίνακα Μεταβάσεων:

Στον πίνακα μεταβάσεων, θα κινηθούμε από πάνω προς τα κάτω. Οριζόντια είναι οι τιμές που δέχονται οι διάφορες καταστάσεις (states), ενώ κάθετα οι καταστάσεις που δέχονται τις τιμές. Οι τιμές για την εργασία είναι από 0 έως 9 και τα σημεία στίξης (. :). Η αρχική κατάσταση (state) είναι η q0. Η q0 δέχεται τις τιμές 0,1 ή 2. Εάν δεχθεί μία εκ των 0 ή 1 προχωράει στην κατάσταση q1 αλλιώς αν δεχθεί την τιμή 2 προχωράει στην κατάσταση q2. Εάν βρίσκεται στην q1 και δεχθεί τιμή 0-9 τότε συνεχίζει στην q3. Αν είναι στην q2 και δεχθεί τιμή 0-3 τότε συνεχίζει στην κατάσταση q3. Από την q3 εάν δεχθεί ένα από τα σημεία στίξης : ή . συνεχίζει στην κατάσταση q4. Από την q4 εάν δεχθεί τιμή 0-5 συνεχίζει στην q5. Και τέλος, εάν δεχθεί τιμή από 0-9 προχωρά και σταματά στην κατάσταση υποδοχής, που είναι και η τελευταία κατάσταση, την q6. Σε οποιαδήποτε περίπτωση εάν δεχθεί μη αποδεκτή τιμή τότε από την κατάσταση στην οποία βρίσκεται δεν προχωράει, αλλά δίνει το ERROR_TOKEN και υποδεικνύει που βρίσκεται το λάθος.

Συνοπτική Περιγραφή Κώδικα:

getchar(): Διαβάζει τον χαρακτήρα που της ανατίθεται και έπειτα τον δίνει. Στην συνάρτηση ανατίθενται χαρακτήρες προς ανάγνωση μέσω της συνάρτησης *scan()*, όπου μέσω μίας while-loop δίνεται κάθε φορά στην *getchar()* ένας χαρακτήρας σε διαφορετική θέση για ανάγνωση, τον οποίο στη συνέχεια θα τον επιστρέψει.

scan(): Η συνάρτηση *scan()* διαβάζει το input που δίνεται από τον χρήστη. Στη συνέχεια μέσω της while-loop παίρνει κάθε χαρακτήρα ξεχωριστά και βλέπει αν ο συγκεκριμένος χαρακτήρας βρίσκεται στο λεξικό μεταβάσεων και γίνεται αποδεκτός από την κατάσταση στην οποία ο μεταγλωττιστής βρίσκεται εκείνη τη στιγμή. Αν ναι, τότε προχωράει στην επόμενη κατάσταση, έως ότου φτάσει στην κατάσταση υποδοχής. Αν όχι, τότε επιστρέφει ένα "ERROR_TOKEN".

td(): Είναι το λεξικό μεταβάσεων, το οποίο περιέχει τις καταστάσεις μετάβασης, που δέχεται ο μεταγλωττιστής και προχωράει για την αναγνώριση του input που έδωσε ο χρήστης.

ad(): Περιέχει την κατάσταση υποδοχής, στην οποία φτάνει ο μεταγλωττιστής, όταν αναγνώσει επιτυχημένα όλους τους χαρακτήρες του δοσμένου input.

Το πρόγραμμα, εκτελείται μέσω της while-loop που βρίσκεται μεταξύ των γραμμών 64-72. Μέσω αυτής, χρησιμοποιείται η συνάρτηση *scan()*, όπου διαβάζει τους χαρακτήρες του δοσμένου input έως το πέρας του. Αν υπάρχει κάποιο πρόβλημα με κάποιον από τους χαρακτήρες και επιστραφεί το ERROR_TOKEN, τότε μέσω της while-loop θα εκτυπωθεί σε ποια ακριβώς θέση υπήρξε λανθασμένος χαρακτήρας. Αν όχι, τότε εκτυπώνει το token της κατάστασης υποδοχής και το δοσμένο input που επιτυχημένα διάβασε ο μεταγλωττιστής.

Ενδεικτικά Στιγμιότυπα δοκιμών:

- Δοκιμές με αποδεκτές τιμές:

```
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>12.58
TIME_TOKEN : 12.58
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>07.58
TIME_TOKEN : 07.58
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>14:35
TIME_TOKEN : 14:35
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>04:20
TIME_TOKEN : 04:20
aristotle@reggaeshark ~/Desktop/compilers/ex1 $
```

- Δοκιμές με μη αποδεκτές τιμές

```
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>12:65
ERROR TOKEN      Unrecognized input at pos 4 of 12:65
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>12,45
ERROR TOKEN      Unrecognized input at pos 3 of 12,45
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>34:30
ERROR TOKEN      Unrecognized input at pos 1 of 34:30
```

```
aristotle@reggaeshark ~/Desktop/compilers/ex1 $ python3 ex1.py
What is the time?>1a.45
ERROR TOKEN      Unrecognized input at pos 2 of 1a.45
```