

**Τμήμα Εφαρμοσμένης Πληροφορικής
ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

Εξάμηνο Α'

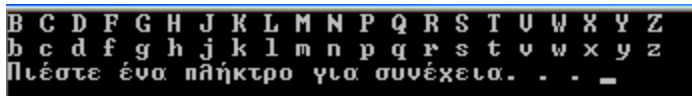
Φύλλο Ασκήσεων 7 – ΧΑΡΑΚΤΗΡΕΣ & ΑΛΦΑΡΙΘΜΗΤΙΚΑ

**Διδάσκοντες: Μάγια Σατρατζέμη, Αλέξανδρος Χατζηγεωργίου, Στέλιος Ξυνόγαλος,
Θεόδωρος Κασκάλης, Ηλίας Σακελλαρίου, Απόστολος Αμπατζόγλου**

Παρατηρήσεις:

1. Τα δεδομένα εισόδου διαβάζονται με τη σειρά που δηλώνονται στις εκφωνήσεις. Για κάθε δεδομένο εισόδου να χρησιμοποιείτε προτρεπτικό μήνυμα που θα ενημερώνει τον χρήστη για την τιμή που αναμένεται.
2. Αντίστοιχα για τα δεδομένα εξόδου και όπου δεν υπάρχουν περαιτέρω διευκρινήσεις για τη μορφή τους, αυτά θα εμφανίζονται με ξεχωριστές εντολές `printf("...\n")` το καθένα και με τη σειρά που δηλώνονται στις εκφωνήσεις.
3. Τα αριθμητικά δεδομένα αναπαρίστανται πάντα από μεταβλητές ακέραιου τύπου (`int` ή `long`). Σε αντίθετη περίπτωση (μεταβλητές τύπου `double`) θα γίνονται οι απαραίτητες διευκρινήσεις.

1. Να γραφεί ένα πρόγραμμα το οποίο θα δέχεται ως είσοδο ένα αλφαριθμητικό το πολύ 25 χαρακτήρων (`char[26]`) και θα υπολογίζει κι εμφανίζει το πλήθος των 'Α' (κεφαλαίων αγγλικών χαρακτήρων) που περιέχονται στο αλφαριθμητικό.
2. Να γράψετε πρόγραμμα, το οποίο:
 - a. θα δέχεται ως είσοδο έναν ακέραιο από το 1 ως το 26 και θα εμφανίζει σε μια γραμμή με ένα κενό μεταξύ τους τον αριθμό, το αντίστοιχο κεφαλαίο, και το αντίστοιχο πεζό (δηλαδή "μικρό") γράμμα του αγγλικού αλφαβήτου.
 - b. θα εκτελεί την αντίστροφη διαδικασία, δηλαδή θα δέχεται από το πληκτρολόγιο ένα κεφαλαίο γράμμα του αγγλικού αλφαβήτου και θα εμφανίζει τη "σειρά" του στην αλφάβητο (π.χ. $A \Rightarrow 1$ κλπ).
3. Να γραφεί ένα πρόγραμμα που θα αφαιρεί τους κενούς χαρακτήρες (χαρακτήρας 32 στον κώδικα ASCII) που ενδεχόμενα μπορεί να υπάρχουν στην αρχή και το τέλος ενός αλφαριθμητικού και θα εμφανίζει το νέο αλφαριθμητικό που προκύπτει και το μήκος του.
4. Να γραφεί πρόγραμμα που θα μετατρέπει έναν θετικό ακέραιο σε αλφαριθμητικό προκαθορισμένου μήκους `L` τοποθετώντας μηδενικά στις τυχόν κενές θέσεις που μένουν μπροστά από τον αριθμό (π.χ. αν δοθεί ο ακέραιος 4437 και `L=7`, τότε το αποτέλεσμα θα είναι το αλφαριθμητικό '0004437').
5. Υλοποιήστε τη συνάρτηση `IsConsonant(ch)`, η οποία να επιστρέφει `TRUE` αν ο `ch` είναι σύμφωνο της αγγλικής γλώσσας, δηλαδή οποιοδήποτε γράμμα εκτός από τα πέντε φωνήεντα: 'a', 'e', 'i', 'o', 'u'. Όπως και η `IsVowel`, η συνάρτησή σας θα πρέπει να αναγνωρίζει τα σύμφωνα και στις δύο μορφές τους (πεζά και κεφαλαία). Γράψτε ένα πρόγραμμα δοκιμής που να εμφανίζει όλα τα κεφαλαία σύμφωνα με ένα κενό μεταξύ τους, και στη συνέχεια σε νέα γραμμή όλα τα πεζά σύμφωνα με ένα κενό μεταξύ τους.



```
B C D F G H J K L M N P Q R S T U W X Y Z
b c d f g h j k l m n p q r s t u w x y z
Πιέστε ένα πλήκτρο για συνέχεια. . . _
```

6. Γράψτε μια συνάρτηση `RandomWord` που να επιστρέφει μια τυχαίως κατασκευασμένη "λέξη" η οποία θα αποτελείται από τυχαίως επιλεγμένα γράμματα. Το πλήθος των γραμμάτων της λέξης θα πρέπει να επιλέγεται με τυχαίο τρόπο, με την επιλογή ενός αριθμού που βρίσκεται μεταξύ των τιμών των σταθερών `#define MinLetters` και `MaxLetters`. Γράψτε ένα κυρίως πρόγραμμα που να δοκιμάζει τη συνάρτησή σας εμφανίζοντας πέντε τυχαίες λέξεις. Χρησιμοποιείτε τις τιμές 2 και 8 για τις σταθερές αντίστοιχα.

7. Στο παιχνίδι Scrabble σε κάθε γράμμα της αγγλικής αλφαβήτου αντιστοιχούν συγκεκριμένοι πόντοι (βαθμοί) ως εξής:

Πόντοι	Γράμματα
1	A, E, I, L, N, O, R, S, T, U
2	D, G
3	B, C, M, P
4	F, H, V, W, Y
5	K
8	J, X
10	Q, Z

Για παράδειγμα η λέξη “FARM” στο Scrabble αξίζει 9 πόντους: 4 για το F, από 1 για τα A και R και 3 για το M. Γράψτε μια συνάρτηση `ScrabbleScore` που να δέχεται ως όρισμα μια λέξη (με 20 το πολύ χαρακτήρες - `char[21]`) και να επιστρέφει το σκορ που θα κέρδιζε η λέξη αν παιζόταν στον πίνακα του Scrabble, χωρίς να μετρά τους υπόλοιπους πόντους που δίνονται στο παιχνίδι. Για τον υπολογισμό του σκορ, θα πρέπει να αγνοείτε όλους τους χαρακτήρες που δεν είναι κεφαλαία γράμματα. Πιο συγκεκριμένα, τα πεζά γράμματα θα πρέπει να θεωρείτε ότι αναπαριστούν κενά τετραγωνίδια τα οποία μπορούν να αντιπροσωπεύουν οποιοδήποτε γράμμα αλλά δίνουν μηδέν πόντους. Γράψτε ένα κυρίως πρόγραμμα για να ελέγξετε τη συνάρτηση `ScrabbleScore`. Δοκιμάστε τις λέξεις “XI”, “HORN”, “SCRABBLE” και “QUIZZICAL” και βεβαιωθείτε ότι επιστρέφουν την τιμή 9, 7, 14 και 28 αντίστοιχα.

8. Υλοποιήστε μια συνάρτηση `Capitalize(str)` που να επιστρέφει ένα αλφαριθμητικό στο οποίο ο αρχικός χαρακτήρας είναι κεφαλαίος (αν είναι γράμμα) και όλα τα υπόλοιπα γράμματα να μετατρέπονται σε πεζά. Οι χαρακτήρες που δεν είναι γράμματα δεν θα πρέπει να επηρεάζονται. Για παράδειγμα, οι κλήσεις `Capitalize(“BOOLEAN”)` και `Capitalize(“Boolean”)` θα πρέπει να επιστρέφουν και οι δύο το αλφαριθμητικό “Boolean”.

9. Γράψτε μια διαδικασία `DateString(int day, int month, int year, char date[10])` που θα δέχεται ημέρα (day), μήνα (month) και έτος (year) και θα επιστρέφει ένα αλφαριθμητικό (`date[10]`) το οποίο θα αποτελείται από την ημέρα του μήνα, μια παύλα, τα τρία πρώτα γράμματα του ονόματος του μήνα (στα αγγλικά), άλλη μια παύλα, και τα δυο τελευταία ψηφία του έτους.

Για παράδειγμα, η κλήση της διαδικασίας

```
DateString(22, 11, 1963, date)
```

θα πρέπει να επιστρέφει το αλφαριθμητικό “22-Nov-63” μέσω της παραμέτρου `date`.

10. Γράψτε μια συνάρτηση `RegularPluralForm(word)` που να επιστρέφει τον πληθυντικό της αγγλικής λέξης του ορίσματος `word` σύμφωνα με τους παρακάτω καθιερωμένους κανόνες της αγγλικής γραμματικής:

- Αν η λέξη τελειώνει σε s, x, ch ή sh, προστίθενται στη λέξη τα γράμματα es.
- Αν η λέξη τελειώνει σε y και πριν από το y υπάρχει σύμφωνο, το y μετατρέπεται σε ies.
- Σε όλες τις άλλες περιπτώσεις, προστίθεται απλώς ένα s.

Γράψτε ένα πρόγραμμα δοκιμής και σχεδιάστε ένα σύνολο περιπτώσεων ελέγχου για να επαληθεύσετε ότι το πρόγραμμα σας λειτουργεί σωστά. Συγκεκριμένα: στο κυρίως πρόγραμμα ο χρήστης θα δίνει ως είσοδο μια λέξη και θα εμφανίζεται ο πληθυντικός της λέξης ενώ στη συνέχεια θα εξετάζονται οι περιπτώσεις a, b, c με 7 λέξεις, οι 6 για τον έλεγχο των περιπτώσεων a-b και 1 λέξη για την περίπτωση c. Οι λέξεις θα καθορίζονται εντός του προγράμματος και θα είναι αποθηκευμένες σε πίνακα. Θεωρείστε ότι κάθε λέξη δεν είναι μεγαλύτερη από 21 χαρακτήρες.

11. Οι διευθύνσεις ηλεκτρονικής αλληλογραφίας αποτελούνται από ένα ή περισσότερους χαρακτήρες ακολουθούμενους από τον χαρακτήρα “@” και μια διεύθυνση ενός εξυπηρετητή αλληλογραφίας (mail server). Έτσι μια διεύθυνση είναι:

alanturing@it.heaven.pantheon
Όνομα Διεύθυνση mail server

Παραδείγματα διευθύνσεων είναι οι `alanturing@it.heaven.pantheon`, `student@uom.gr`, και `wirth@yahoo.com`. Να γραφεί πρόγραμμα το οποίο:

- α) Να διαβάζει από τον χρήστη μια e-mail διεύθυνση της παραπάνω μορφής (ως αλφαριθμητικό).
- β) Να τυπώνει στην οθόνη το μέρος της διεύθυνσης που αντιστοιχεί στο όνομα καθώς και το μήκος του ονόματος.
- γ) Να τυπώνει στην οθόνη το μέρος της διεύθυνσης που αντιστοιχεί στη διεύθυνση του εξυπηρετητή αλληλογραφίας (mail server).
- δ) Σε περίπτωση που υπάρχουν κενά (χαρακτήρας 32 στον κώδικα ASCII) στην αρχή και στο τέλος της αρχικής διεύθυνσης ηλεκτρονικού ταχυδρομείου να αφαιρεί τα κενά αυτά, πριν κάνει την επεξεργασία που περιγράφεται παραπάνω.

12. Το σύστημα κρατήσεων μιας αεροπορικής εταιρείας δεν επιτρέπει την καταχώριση του ίδιου ονόματος δύο φορές σε μία πτήση. Να γραφεί πρόγραμμα το οποίο θα διαβάζει τα ονόματα των επιβατών ενός αεροσκάφους το πολύ 100 θέσεων και θα τα καταχωρεί σε κατάλληλο πίνακα. Η καταχώριση θα σταματάει όταν αντί για όνομα δοθούν τέσσερις παύλες ("----"). Το πρόγραμμα δεν πρέπει να επιτρέπει την καταχώριση του ίδιου ονόματος στον πίνακα και τέλος θα εμφανίζει τα ονόματα όλων των επιβατών της πτήσης. Για παράδειγμα:

```
Passenger in seat 1 :nick
Passenger in seat 2 :helen
Passenger in seat 3 :john
Passenger in seat 4 :----

Passenger List
-----
In seat 1:nick
In seat 2:helen
In seat 3:john
Press any key to continue...
```

```
Passenger in seat 1 :john
Passenger in seat 2 :john
Passenger john already exists.
Passenger in seat 2 :mary
Passenger in seat 3 :----

Passenger List
-----
In seat 1:john
In seat 2:mary
Press any key to continue...
```

13. Να γραφεί συνάρτηση η οποία θα δέχεται ως παραμέτρους δύο αλφαριθμητικά και θα εμφανίζει τους κοινούς τους χαρακτήρες. Στην πρώτη περίπτωση θα εμφανίζονται όλοι οι χαρακτήρες της πρώτης λέξης που περιέχονται στην δεύτερη, ακόμη και αν υπάρχουν επαναλήψεις. Στην δεύτερη περίπτωση θα εμφανίζονται οι κοινοί χαρακτήρες χωρίς επαναλήψεις. Για παράδειγμα:

```
Please insert two word to find the common characters.
Word 1: committee
Word 2: communication
Common Characters of committee in communication:committ

----
Set of (unique) Common Characters of committee and communication:comit

----
Press any key to continue . . .
```

14. Να γραφεί συνάρτηση η οποία θα δέχεται ως παράμετρο ένα αλφαριθμητικό και θα αναγραμματίζει με τυχαίο τρόπο τους χαρακτήρες της. Μια απλή προσέγγιση είναι για κάποιο αριθμό επαναλήψεων να επιλέγετε δύο τυχαίους χαρακτήρες της λέξης (τυχαίες θέσεις) και να ανταλλάσετε τις θέσεις τους. Οι τυχαίοι αριθμοί θα προκύπτουν με αρχικοποίηση της γεννήτριας τυχαίων αριθμών. Στο κυρίως πρόγραμμα θα εμφανίζετε το αρχικό αλφαριθμητικό και το αναγραμματισμένο.

```
Word to shuffle :apple
Initial Word: apple, New Word: elpap
Press any key to continue . . .
```

15. Να γραφεί ένα πρόγραμμα το οποίο θα δέχεται ως είσοδο ένα αλφαριθμητικό το πολύ 25 χαρακτήρων (char str[26]) και θα υπολογίζει και εμφανίζει το πλήθος του κάθε κεφαλαίου λατινικού χαρακτήρα (A-Z) που περιέχεται στο αλφαριθμητικό. Αν ένας χαρακτήρας δεν περιέχεται στο αλφαριθμητικό τότε δεν θα εμφανίζεται το πλήθος του. Για την εύρεση του πλήθους εμφάνισης ενός οποιουδήποτε χαρακτήρα σε ένα αλφαριθμητικό θα πρέπει να ορίσετε μια συνάρτηση που θα δέχεται το αλφαριθμητικό και τον χαρακτήρα και επιστρέφει το πλήθος εμφανίσεων του χαρακτήρα αυτού στο αλφαριθμητικό.

```
Dose ena alfarithmitiko to poly 25 charaktiron: THiS iS a TEST SENTenCE
C --> 1
E --> 3
H --> 1
N --> 1
S --> 4
T --> 3
Press any key to continue . . .
```

16. Να γραφεί ένα πρόγραμμα το οποίο θα δέχεται ένα αλφαριθμητικό μέχρι 100 χαρακτήρες και να εμφανίζει τα ψηφία που περιέχονται μέσα στο αλφαριθμητικό και το άθροισμά τους, όπως φαίνεται στο παρακάτω παράδειγμα

```
Dose ena alfarithmitiko: Test25Phrase35Containing22Numbers11
2 + 5 + 3 + 5 + 2 + 2 + 1 + 1 = 21
Press any key to continue . . .
```

17. Να γραφεί ένα πρόγραμμα το οποίο θα δέχεται 2 μεταβλητές Onoma και Epitheto θα τις συνενώνει, προσθέτοντας ένα κενό ανάμεσα, και θα τις εκτυπώνει με κεφαλαία γράμματα.

```
Dwse onoma: nikolaos
Dwse eponymo: papadimitriou

Onoma = nikolaos
Eponymo = papadimitriou
Onomateponymo = NIKOLAOS PAPADIMITRIOU
Press any key to continue . . .
```