

## Ασκήσεις 6- Χαρακτήρες και Συμβολοσειρές

### Άσκηση 6.1 (βλ. παραδείγματα π6.1. και π6.2)

Να γραφεί πρόγραμμα που θα διαβάζει χαρακτήρες απ' το πληκτρολόγιο και να εμφανίζει τον κωδικό ASCII του κάθε χαρακτήρα. Η διαδικασία να επαναλαμβάνεται μέχρι να δοθεί ο χαρακτήρας '@'.

**Οδηγία:** θα χρειαστεί: είτε να εκχωρηθεί ο χαρακτήρας σε μια ακέραια μεταβλητή, είτε κατά την έξοδο να γίνει προσαρμογή (type casting) του χαρακτήρα σε `int`, βλ. σημειώσεις.

Στη συνέχεια, αφού υλοποιηθεί το παραπάνω, να προστίθεται -σταδιακά- κώδικας για τα ακόλουθα: Κάθε ερώτημα μπορεί να υλοποιηθεί ξεχωριστά από τα υπόλοιπα ή να γίνει με ξεχωριστό πρόγραμμα.

- Να ζητά απ' το χρήστη να εισάγει έναν αριθμό από 32-255 και να εμφανίζει τον χαρακτήρα που αντιστοιχεί σ' αυτόν τον αριθμό σύμφωνα με την κωδικοποίηση ASCII.
- Να διαβάζει έναν χαρακτήρα και να εμφανίζει τον αμέσως επόμενο απ' αυτόν. Για παράδειγμα αν πληκτρολογήθηκε το `d` να εμφανιστεί το `e`.
- Να διαβάζει ένα χαρακτήρα και να τον μετατρέπει σε κεφαλαίο.

**Οδηγία:** Αυτό μπορεί να γίνει :

- ελέγχοντας αν ο χαρακτήρας είναι μεταξύ 'a' και 'z' οπότε στην περίπτωση αυτή θα πρέπει να αφαιρεθεί η κατάλληλη ποσότητα για να γίνει κεφαλαίος, ή
- με χρήση της συνάρτησης `toupper(c)` του αρχείου συμπερίληψης `<cctype>` η οποία επιστρέφει ως κεφαλαίο τον χαρακτήρα που δέχεται ως παράμετρο, π.χ. με την `c = toupper(c)` τότε μετά την κλήση ο `c` θα είναι κεφαλαίος, αν αρχικά ήταν μικρός, αλλιώς παραμένει όπως ήταν.

### Άσκηση 6.2 (βλ. παραδείγματα π6.3 και π6.3)

Να γραφεί πρόγραμμα που θα διαβάζει μια συμβολοσειρά της C μεγέθους μέχρι 19 χαρακτήρων (χωρίς τον κωδικό τέλους) και να υπολογίζει το μήκος της συμβολοσειράς.

**Οδηγίες:** Ο κώδικας στην ουσία είναι ένας βρόχος `while (s[i] != '\0' ) { }`, όπου μέσα στο σώμα του βρόχου θα γίνεται αύξηση του μετρητή-δείκτη. Συγκρίνετε το αποτέλεσμα που θα βρείτε με το αποτέλεσμα της συνάρτησης `strlen()` του `<cstring>`.

Μια συμβολοσειρά `s` μπορεί να διαβαστεί από το πληκτρολόγιο με έναν από τους παρακάτω τρόπους:

```
cin >> s; (στην s εκχωρούνται οι χαρακτήρες μέχρι να βρεθεί κενό ή enter, συνεπώς η s δεν μπορεί να περιέχει κενά)
cin.getline(s, 20); (στην s θα εκχωρηθούν μέχρι 19 χαρακτήρες ή μέχρι να δοθεί Enter, δηλ. εδώ η s μπορεί να περιέχει και κενά)
```

Στη συνέχεια, προσθέστε κώδικα για τα ακόλουθα βήματα:

- b. Θα διαβάζει έναν χαρακτήρα που δίνει ο χρήστης και θα ελέγχει αν υπάρχει στη συμβολοσειρά. Στο τέλος, αν μεν υπάρχει, θα εμφανίζει τη θέση στην οποία βρίσκεται, αλλιώς θα εμφανίζει κατάλληλο μήνυμα.

**Οδηγία:** Ο κώδικας στην ουσία υλοποιείται με ένα βρόχο `while ( ) { }`, όπου μέσα στο σώμα του βρόχου θα γίνεται η ζητούμενη επεξεργασία.

- c. Όσοι χαρακτήρες της συμβολοσειράς είναι αριθμητικά ψηφία να αντικατασταθούν με τον χαρακτήρα '@'.

**Οδηγία:** Οι κωδικοί των αριθμητικών ψηφίων ('0' έως '9') βρίσκονται σε συνεχόμενες θέσεις στον πίνακα ASCII (από 48-57). Επίσης για τον έλεγχο μπορεί να γίνει χρήση της συνάρτησης `isdigit ( )` του αρχείου συμπερίληψης `<cctype>`.

### Άσκηση 6.3 (βλ. παράδειγμα π6.4)

Να γραφεί πρόγραμμα που θα διαβάζει δυο συμβολοσειρές της C μεγέθους μέχρι 19 χαρακτήρων (χωρίς τον κωδικό τέλους) και να τις συγκρίνει μεταξύ τους λεξικογραφικά. Για παράδειγμα το `AbCD` είναι μεγαλύτερης τάξης λεξικογραφικά από το `ABCDEF`. Αν είναι ίδιες θα εμφανίζει κατάλληλο μήνυμα, αλλιώς θα εμφανίζει τη μικρότερη (λεξικογραφικά) από τις δυο.

**Οδηγία:** Μπορεί να γίνει χρήση της συνάρτησης `strcmp(s1,s2)` του `<cstring>` η οποία συγκρίνει τις συμβολοσειρές `s1` και `s2` και επιστρέφει 0 αν είναι ίδιες, ή έναν αρνητικό αριθμό αν `s1<s2` ή έναν θετικό αριθμό αν `s1>s2`.

**6.3.B:** Να δημιουργεί μια νέα συμβολοσειρά που να προκύπτει από την συνένωση των δυο συμβολοσειρών που εισήχθηκαν, διαχωρίζοντας τις δυο συμβολοσειρές με κενό διάστημα. Π.χ. αν ο χρήστης εισάγει `bcd` και `klmno` τότε η νέα συμβολοσειρά θα είναι `bcd klmno`.

**Οδηγία:** Μπορεί να γίνει χρήση των συναρτήσεων `strcpy(s1,s2)` για την αντιγραφή (εκχώρηση) της `s2` στην `s1` και της `strcat(s1,s2)` για την συνένωση της `s2` στο τέλος της `s1` (βλ. παράδειγμα 6.4).

### Άσκηση 6.4 (όπως η άσκηση 6.3 αλλά να υλοποιηθεί με αντικείμενα τύπου `string` της C++)

Να γραφεί πρόγραμμα που θα διαβάζει δυο συμβολοσειρές της C++, δηλ. αντικείμενα της τάξης-κλάσης `string` και να τις συγκρίνει μεταξύ τους λεξικογραφικά. Αν είναι ίδιες θα εμφανίζει κατάλληλο μήνυμα, αλλιώς θα εμφανίζει τη μικρότερη (λεξικογραφικά) από τις δυο. Χρειάζεται να συμπεριλάβουμε το αρχείο `<string>`

**Οδηγία:** Οι δυο συμβολοσειρές `s1, s2` θα δηλωθούν ως μεταβλητές τύπου `string`. Στην ουσία είναι αντικείμενα της τάξης `string` και όχι πίνακες χαρακτήρων. Έτσι μπορεί να γίνει χρήση των τελεστών `==` `!=` `>` `>=` `<` `<=` για τη σύγκρισή τους.

**6.4.B:** Να δημιουργεί μια νέα συμβολοσειρά που να προκύπτει από την συνένωση των δυο συμβολοσειρών που εισήχθησαν, διαχωρίζοντάς τις με κενό διάστημα. Π.χ. αν ο χρήστης εισάγει bcd και KLMNO τότε η νέα συμβολοσειρά θα είναι bcd KLMNO.

**Οδηγία:** Αφού οι συμβολοσειρές είναι πλέον τύπου string, μπορούμε να χρησιμοποιούμε τους τελεστές + και += για την συνένωσή τους.

#### Άσκηση 6.5 (αν υπάρχει χρόνος, ειδάλλως για το σπίτι).

Να γραφεί πρόγραμμα που θα διαβάζει μια συμβολοσειρά και να ελέγχει αν μέσα σ' αυτήν περιέχεται η συμβολοσειρά "eta" οπότε και θα εμφανίζει την θέση στην οποία περιέχεται αλλιώς θα εμφανίζει ότι δεν περιέχεται. Π.χ. αν ο χρήστης εισάγει την "nikoletatos" θα εμφανίζει "υπάρχει στη θέση 5", αν εισάγει την "petalidis" θα εμφανίζει "υπάρχει στη θέση 1", ενώ αν εισάγει την "AbeTdf" θα εμφανίζει "δεν περιέχεται".

**Οδηγία:** Αν η συμβολοσειρά δηλωθεί ως αντικείμενο τύπου string της C++ τότε μπορεί να γίνει χρήση της μεθόδου-συνάρτησης find(). Η s1.find(s2) ελέγχει αν η s2 εμπεριέχεται μέσα στην s1 και στην περίπτωση που περιέχεται επιστρέφει τη θέση της s1 στην οποία περιέχεται η s2 ενώ αν η s2 δεν περιέχεται τότε η συνάρτηση θα επιστρέψει -1. Π.χ. αν string s="ABCDXUDE", τότε η s.find("CD") θα δώσει ως αποτέλεσμα το 2 ενώ η s1.find("CDE") θα δώσει ως αποτέλεσμα το -1.

Αν θέλουμε η συμβολοσειρά δηλωθεί ως πίνακας χαρακτήρων της C τότε μπορεί να γίνει χρήση της συνάρτησης strstr() με έναν τρόπο παρόμοιο με το παράδειγμα π6.5β ή με το π6.6γ

**6.5B:** Να δηλωθεί ένας πίνακας ονομάτων ο οποίος θα πάρει τιμές από το πληκτρολόγιο. Επίσης ο χρήστης να εισάγει μια νέα συμβολοσειρά από το πληκτρολόγιο. Να εμφανιστούν τα ονόματα εκείνα που περιέχουν την δεύτερη συμβολοσειρά. Π.χ. αν ο πίνακας περιέχει το "nikoletatos" το "petalidis" και το "nikolaou" και ο χρήστης εισάγει σαν συμβολοσειρά αναζήτησης την "eta" τότε το πρόγραμμα θα εμφανίζει ότι

περιέχεται στα:

nikoletatos petalidis

**Οδηγία:** Ένας πίνακας για N ονόματα μπορεί να δηλωθεί και ως πίνακας N αντικειμένων τύπου string της C++, π.χ. με τη δήλωση: string onomata[N]; όπου onomata[0] θα είναι το πρώτο όνομα, onomata[1] θα είναι το δεύτερο όνομα, κ.ο.κ.. Στη συνέχεια μπορεί να χρησιμοποιηθεί η find() όπως παραπάνω για να ελεγχθούν όλα τα ονόματα. Το πρόγραμμα για την περίπτωση των συμβολοσειρών-πινάκων της C είναι υλοποιημένο στο παράδειγμα π6.6γ. Στην C ένας πίνακας για N ονόματα μεγέθους 30 χαρακτήρων το πολύ, δηλώνεται σαν ένας δισδιάστατος πίνακας χαρακτήρων, π.χ. char onomata[N][30], και όπου onomata[0] είναι το πρώτο όνομα, onomata[1] είναι το δεύτερο όνομα, κ.λ.π. Το παράδειγμα π6.6γ λοιπόν θέλει ελάχιστη μετατροπή με βάση το παράδειγμα π6.9 για να δουλέψει με αντικείμενα τύπου string.