

Ασκήσεις - Εισαγωγή στα Αρχεία

Άσκηση 7.1

Ένα αρχείο κειμένου (text file ή ASCII file) σε κάθε γραμμή του έχει: *όνομα* και *ηλικία*

Hatziyiannopoulos	17
Papadopoulos	19
Alexakis	35
Papadopoulos	25
Alexiadis	32
Papayiannis	24
Palatzian	37

Το πλήθος των γραμμών θεωρείται άγνωστο.

Το αρχείο αυτό αν δεν υπάρχει, μπορεί να δημιουργηθεί με κάποιον text editor (π.χ. το σημειωματάριο, ή τον text editor του περιβάλλοντος ανάπτυξης εφαρμογών, όπως το Code Blocks).

A. Να γραφεί πρόγραμμα που να διαβάζει τα δεδομένα και να τα εμφανίζει στην οθόνη.

Οδηγία: (βλ. Παράδειγμα π 7.3).

Δηλ. το πρόγραμμα πρέπει:

α) να δηλώσει και ανοίξει ένα ρεύμα `fin` προς ένα υπάρχον αρχείο για ανάγνωση.

```
ifstream fin ("ονομα αρχείου");
```

β) να επιχειρήσει να διαβάσει τα στοιχεία μιας νέας εγγραφής από το αρχείο

```
fin >> onoma >> ilikia;
```

γ) να ελέγξει αν βρήκε το τέλος αρχείου

```
έλεγχος της τιμής της fin.eof()
```

δ) αν δεν βρήκε το τέλος αρχείου να εμφανίσει τα στοιχεία αυτής της εγγραφής στην οθόνη

```
cout << onoma << " " << ilikia << endl;
```

να επαναλάβει τα β-δ.

ε) να κλείσει το ρεύμα προς το αρχείο.

```
fin.close();
```

7.1.B (Οι φοιτητές να προχωρήσουν σ' αυτό αφού υλοποιήσουν επιτυχώς το στάδιο A.)

Στο μέρος αυτό της άσκησης πρέπει να προστίθενται εντολές κάθε ένα από τα παρακάτω ερωτήματα. Οι προσθήκες κώδικα γίνονται κυρίως μέσα στον βρόχο ανάγνωσης/επεξεργασίας ενώ πριν από αυτόν τοποθετούνται κυρίως οι δηλώσεις νέων μεταβλητών και εισαγωγές από το πληκτρολόγιο νέων τιμών. Προχωρούμε σταδιακά, σε ένα νέο ερώτημα αφού πρώτα ελέγξουμε ότι λειτουργεί σωστά ο κώδικας που γράψαμε για το προηγούμενο. Κάθε ερώτημα μπορεί να απαντηθεί ξεχωριστά.

7.1.B1. (υπολογισμός μέσης ηλικίας)

Να προστεθούν μέσα στον βρόχο ανάγνωσης οι εντολές για τον υπολογισμό της μέσης ηλικίας όλων των ατόμων. (βλ. Παράδειγμα π7.4)

7.1.B2. (γράψιμο σε νέο αρχείο, έλεγχος ισότητας συμβολοσειρών)

Να προστεθούν πριν, μέσα και μετά τον βρόχο ανάγνωσης εντολές ώστε το πρόγραμμα να γράφει σε νέο αρχείο εκείνες τις εγγραφές, δηλ. όνομα και ηλικία, που το όνομά τους είναι το ίδιο με κάποιο όνομα που θα έχει εισάγει ο χρήστης.

Οδηγίες:

Όσον αφορά στο χειρισμό του αρχείου εξόδου βλ. παράδειγμα π7.4.

Όσον αφορά στο θέμα σύγκρισης συμβολοσειρών:

Αν δηλώσουμε τις συμβολοσειρές σαν αντικείμενα της κλάσης `string` της C++ (π.χ. `string s1, s2;`) μπορεί να γίνει χρήση του τελεστή `==` για τον έλεγχο ισότητας π.χ `if (s1 == s2)`

Αν όμως δηλώσουμε τις συμβολοσειρές σαν πίνακες χαρακτήρων, θα πρέπει να γίνει χρήση της συνάρτησης `strcmp(s1, s2)` του `<cstring>` η οποία συγκρίνει τις συμβολοσειρές `s1` και `s2` και επιστρέφει 0 αν είναι ίδιες, έναν αρνητικό αριθμό αν `s1 < s2` ή έναν θετικό αριθμό αν `s1 > s2`. (βλ. Παράδειγμα π6.6α.)

7.1.B3. (συνένωση συμβολοσειρών, έλεγχος αν μια συμβολοσειρά περιέχεται σε άλλη)

Να προστεθεί ο απαραίτητος κώδικας πριν, μέσα και μετά το βρόχο ανάγνωσης/επεξεργασίας εντολές ώστε το πρόγραμμα να δημιουργεί μια νέα ενιαία συμβολοσειρά η οποία να αποτελείται από την συγκόλληση όλων των ονομάτων του αρχείου τα οποία περιέχουν μια συμβολοσειρά που θα εισάγει ο χρήστης. Τα ονόματα μπορούν να διαχωρίζονται με έναν κενό χαρακτήρα π.χ. αν ο χρήστης εισάγει ή καθορίσει την συμβολοσειρά αναζήτησης "ian", τότε η νέα συμβολοσειρά θα πρέπει να είναι "Hatziyiannopoulos Papayiannis Palatzian".

Οδηγίες:

Πριν τον βρόχο πρέπει να δηλωθούν δυο συμβολοσειρές. Η μια θα αποθηκεύσει το αποτέλεσμα της συγκόλλησης των ονομάτων ενώ η δεύτερη αντιπροσωπεύει τη συμβολοσειρά αναζήτησης. Βολεύει ιδίως στο στάδιο αυτό οι συμβολοσειρές να δηλωθούν ως αντικείμενο τύπου `string` της C++ παρά τότε ως πίνακες χαρακτήρων.

Μέσα στον βρόχο ανάγνωσης/επεξεργασίας θα γίνεται έλεγχος αν το όνομα που αναγνώστηκε από το αρχείο εμπεριέχει τη συμβολοσειρά αναζήτησης οπότε, στην περίπτωση που την εμπεριέχει, το όνομα θα προστίθεται (κολλάει) στην συμβολοσειρά αποτελέσματος.

Αν οι συμβολοσειρές έχουν δηλωθεί ως αντικείμενο τύπου `string` της C++ τότε μπορεί να γίνει χρήση της μεθόδου-συνάρτησης `find()` του `<string>` για την έλεγχο ύπαρξης και του τελεστή `+` για την συγκόλληση.. Π.χ. `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="ABCDXCDE"`, τότε η `s.find("CD")` θα δώσει ως αποτέλεσμα το 2 ενώ η `s.find("CDE")` θα δώσει ως αποτέλεσμα το `-1`. Αν θέλω στην `s` να προσθέσω την `s1` τότε αυτό γίνεται εύκολα με την πρόταση `s = s + s1 ;`

Αν θέλαμε οι συμβολοσειρές να δηλωθούν ως πίνακες χαρακτήρων της C (που δεν είναι και τόσο βολικό εδώ), τότε για τον έλεγχο ύπαρξης θα μπορούσε να γίνει χρήση της συνάρτησης `strstr()` με έναν τρόπο

παρόμοιο με το παράδειγμα π6.5β ή με το π6.6γ, ενώ για τη συγκόλληση συμβολοσειρών θα έπρεπε να χρησιμοποιηθεί η συνάρτηση `strcat (str1, s1)` η οποία κολλάει το `s1` στο τέλος της `str1`,

Άσκηση 7.2 (αν υπάρχει χρόνος, αλλιώς για το σπίτι)

Να γραφεί πρόγραμμα που να δημιουργήσει ένα αρχείο κειμένου (text file) που σε κάθε γραμμή του να περιέχει ένα όνομα και έναν αριθμό, π.χ. ηλικία, τα οποία θα πληκτρολογεί ο χρήστης. Το αρχείο να έχει `N` γραμμές (εγγραφές).

Hatziyiannopoulos	17
Papadopoulos	19
Alexakis	35
Papadopoulos	25
Alexiadis	32
Papayiannis	24
Palatzian	37

Οδηγία: το πρόγραμμα πρέπει

α) να δηλώσει και ανοίξει ένα ρεύμα `f` προς ένα νέο αρχείο που θα δημιουργηθεί για γράψιμο.

```
ofstream f ("ονομα αρχείου");
```

↑ output file stream

↑ Όνομα που διαλέγει ο προγραμματιστής

ή

```
fstream f («ονομα αρχείου», ios::out)
```

↑ σε output mode

β) να διαβάσει τα στοιχεία μιας νέας εγγραφής από το πληκτρολόγιο

```
cin >> onoma >> ilikia;
```

γ) να γράψει τα στοιχεία αυτής της εγγραφής στο αρχείο

```
f << onoma << " " << ilikia << endl;
```

να επαναλάβει τα β και γ, συνολικά `N` φορές

δ) να κλείσει το ρεύμα προς το αρχείο.

```
f.close();
```

Μετά την εκτέλεση ελέγξτε αν όντως τα δεδομένα γράφτηκαν στο αρχείο.

7.2.B (Προσθήκη δεδομένων στο τέλος ενός αρχείου)

Να γραφεί πρόγραμμα στο οποίο ο χρήστης θα εισάγει από το πληκτρολόγιο τα στοιχεία `N` νέων εγγραφών (όνομα και ηλικία) τα οποία θα προστεθούν στο τέλος των δεδομένων που τυχόν ήδη υπάρχουν σε ένα αρχείο.

Οδηγία: Στο πρόγραμμα που ήδη έχει γραφεί παραπάνω πρέπει να αλλάξει μόνο το σημείο `a` όπου δηλώνεται και ανοίγει το ρεύμα προς το αρχείο. Τώρα πλέον πρέπει να ανοιχτεί ένα ρεύμα `f` προς το αρχείο σε κατάσταση (mode) εξόδου (output) και προσθήκης (append). Αυτό θα επιτρέψει ώστε οι εντολές εξόδου προς το αρχείο (δηλ. `f <<`) να γράφουν τα νέα δεδομένα στο τέλος των δεδομένων που ήδη τυχόν υπάρχουν. Έτσι πλέον η δήλωση και το άνοιγμα πρέπει να τροποποιηθεί ως εξής:



α)

```
ofstream f («ονομα αρχείου», ios::app)
```

↑ output file stream

↑ Άνοιγμα σε append mode

ή

 file stream 

```
fstream f («ονομα αρχείου», ios::out|ios::app)
```

Μετά την εκτέλεση ελέγξτε αν όντως τα δεδομένα προστέθηκαν στο αρχείο.