

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ-2

Lesson-4



Programming

ΑΛΦΑΡΙΘΜΗΤΙΚΑ

1. Πίνακας αλφαριθμητικών

→ Ο πίνακας αλφαριθμητικών δηλώνεται στην c με τον εξής τρόπο:

`char s[l][size]`. Όπου:

- `l` → Αριθμός γραμμών πίνακα (Άρα πόσα αλφαριθμητικά θα έχω)!!
- `size` → μέγιστος αριθμός γραμμάτων που μπορεί να έχει ένα αλφαριθμητικό.

2. Διάβασμα πίνακα αλφαριθμητικών από τον χρήστη.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char lines[4][20]; // Θα έχω 4 αλφαριθμητικά με μέγιστο μέγεθος 20 γράμματα το καθένα
    for(int i=0; i<4; i++) // Μονοδιάστατος πίνακας αλφαριθμητικών με 4 στοιχεία
    {
        printf("Give String_%d:", i);
        gets(lines[i]);
    }
}
```

3.Μετατροπή αλφαριθμητικού σε αριθμό.

atoi()→ Συνάρτηση η οποία δέχεται σαν όρισμα ένα αλφαριθμητικό και εμφανίζει τον αντίστοιχο ακέραιο αριθμό.

atof()→ Συνάρτηση η οποία δέχεται σαν όρισμα ένα αλφαριθμητικό και εμφανίζει τον αντίστοιχο δεκαδικό αριθμό.

!!Σε περίπτωση που δεχτεί σαν όρισμα αλφαριθμητικό που ξεκινάει με γράμματα επιστρέφει 0/0.000.

!!Αν ξεκινάω με ψηφία επιστρέφει τον αριθμό μεχρι και το τελευταίο ψηφίο.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main()
{
    char lines[100]="1999";
    printf("%d\n",atoi(lines));//1999
    char line1[100]="1999a";
    printf("%d\n",atoi(line1));//1999
    char line2[20]="asd34f";
    printf("%d\n",atoi(line2));//0
    char line3[20]="21k1";//21
    printf("%d\n",atoi(line3));
    char line4[20]="21.6";
    printf("%lf\n",atof(line4));//21.6
    char line5[20]="asdfg34.7";//0.000000
    printf("%lf\n",atof(line5));
```

ΑΡΧΕΙΑ

- Δήλωση αρχείου → `FILE *fp;`
- Άνοιγμα αρχείου → `fp=fopen("Όνομα_αρχείου","Τύπος προσπέλασης αρχείου");`
(Ανοίγει υπάρχον αρχείο για διάβασμα και γράψιμο, ενώ αν το αρχείο δεν υπάρχει το δημιουργεί, αν το έχει ανοίξει για γράψιμο).

Τύποι Προσπέλασης αρχείου:

- “r” → για να διαβάσω δεδομένα από ένα αρχείο.
- “w” → για να γράψω σε ένα αρχείο
- “a” → προσθήκη στο αρχείο
- “r+” → Άνοιγμα και για διάβασμα και για γράψιμο σε αρχείο (Αν το αρχείο υπάρχει (Προτεραιότητα το read)).
- Διάβασμα δεδομένων από αρχείο → Άνοιγμα αρχείου για διάβασμα

1. `fgets(line, μέγεθος αλφαριθμητικού, δείκτης σε αρχείο)` → Όσο είναι διάφορο του NULL διαβάζει γραμμή. [\[github link\]](#)
2. `fscanf(δείκτης σε αρχείο, "%d(format αρχείου μου)", &x)` → διαβάζει ένα ένα τα περιεχόμενα από ένα αρχείο με συγκεκριμένο format. Όσο η συνάρτηση επιστρέφει μεγαλύτερο του 0 διαβάζουμε από το αρχείο περιεχόμενα. [\[github link\]](#)

Παράδειγμα

1. `fscanf(fp, "%d-%d", &a, &b)` → Αρχείο πρέπει να έχει συγκεκριμένο format[int-int].
 2. `fscanf(fp, "%*d-%d", &a, &b)` → Παρόμοιο με το παραπάνω με μόνη διαφορά ότι με το αστεράκι πριν το προσδιοριστικό παραλείπει τον συγκεκριμένο ακέραιο.
- Εκτύπωση σε αρχείο → Άνοιγμα αρχείου για γράψιμο
 1. `fprintf(fp, "format", variables)` [\[github link\]](#)
 2. `fputc("όνομα αρχείου", character);` [\[github link\]](#) → Τοποθέτηση χαρακτήρα στο αρχείο μας.
 - Κλείσιμο αρχείου
`fclose(pointer to file).`

ΑΝΑΓΝΩΣΗ ΑΠΟ ΑΡΧΕΙΟ

```
#include <stdio.h>
int main()
{
    int a,b;
    FILE *fp;//δήλωση δείκτη που δείχνει σε μία δομη αρχείο
    //εφόσον είναι δείκτης αρχική τιμή είναι το NULL
    fp=fopen("exam.txt","r");//άνοιγμα του αρχείου exam για διάβασμα
    //Αν δεν υπάρχει ο δείκτης θα συνεχίζει να δείχνει στην τιμή NULL
    //αλλιώς θα δείξει στην θέση της μνήμης που είναι αποθηκευμένο το αρχείο
    while(fscanf(fp,"%d-%d",&a,&b)==1)//>0
    {
        //operation
    }//η fscanf έχει παρόμοια λειτουργία με την scanf
    //μόνο που τα δεδομένα δεν τα εισάγει ο χρήστης και πρέπει
    //να δηλώσει και το συγκεκριμένο format που θα έχει το αρχείο
    //ώστε να διαβάσει τα δεδομένα
    //Εναλλακτικός τρόπος
    char line[100];
    while(fgets(line,100,fp)!=NULL)
    {
        if(line[strlen(line)-1]=='\n') line[strlen(line)-1]='\0';
        //Επειδή σε ένα αρχείο ο τελευταίος χαρακτήρας που δέχεται μία γραμμή είναι το enter-->'\\n\\
        //για αλλαγή γραμμής και εμείς θέλουμε να διαβάζουμε την γραμμή χωρίς το '\\n' στην θέση που αποθηκεύεται
        //που είναι και η τελευταία του αλφαριθμητικού βάζουμε τον τερματικό χαρακτήρα!!!!1
        //operation
    }//παρόμοια λειτουργία με την gets
    //Εκχωρεί σε ένα αλφαριθμητικό επαναληπτικά την κάθε γραμμή του αρχείου και όταν σταματάει να διαβάζει γραμμή
    //το αλφαριθμητικό παίρνει την τιμή NULL
    fclose(fp);//κλείσιμο αρχείου
```

ΕΓΓΡΑΦΗ ΣΕ ΑΡΧΕΙΟ

```
#include <stdio.h>
int main()
{
    //Εκτύπωση στοιχείων ενός πίνακα μέσα σε ένα αρχείο
    FILE *fp;
    char name[100];
    printf("Give filename:");
    gets(name);
    int a[]={12,23,10,4,5,6,7,8,11,56};
    fp=fopen(name,"w");//Άνοιγμα αρχείου για εγγραφή
    //ακόμα και αν δεν υπάρχει το αρχείο θα δημιουργηθεί στον ίδιο φάκελο που βρίσκεται το source file
    for(int i=0;i<sizeof(a)/sizeof(a[0]);i++)
    {
        fprintf(fp,"NUmber_%d:%d\n",i+1,a[i]); //Εκτύπωση μέσα στο αρχείο με το συγκεκριμένο format(
        //"NUmber_%d:%d\n")[11 characters/line]
        char ch='A'+a[i];
        fputc(ch,fp); //τοποθέτηση χαρακτήρα στο αρχείο με βάσιμα τον πίνακα ASCII
        //και τον δικό μου πίνακα
    }
    fclose(fp);
}
```


ΑΣΚΗΣΕΙΣ

1. Να κατασκευαστεί πρόγραμμα το οποίο θα δέχεται σαν είσοδο από ένα αρχείο δεκαδικές τιμές (η οποίες θα αντιστοιχούν σε βαθμολογίες), θα κατασκευάζει δυναμικά ένα πίνακα και θα τις αποθηκεύει και θα εμφανίζει τον μέσο βαθμό καθώς και το ποσοστό επιτυχίας στο μάθημα.
2. Να κατασκευαστεί συνάρτηση η οποία θα δέχεται σαν όρισμα έναν πίνακα ακεραίων και θα τον ταξινομεί. Να κατασκευαστεί επίσης συνάρτηση που θα μετατρέπει έναν δεκαδικό αριθμό στον αντιστοιχο 2-δικο του. Τέλος να κατασκευαστεί συνάρτηση η οποία θα δέχεται σαν όρισμα ένα πίνακα και θα αποθηκεύει σε αρχείο τα αποτελέσματα των συναρτήσεων για αυτόν τον πίνακα. Στην κύρια συνάρτηση να κατασκευαστεί ένας πίνακας 6 θέσεων και να δίνονται τιμές από τον χρήστη και να καλείται η συνάρτηση αποθήκευσης σε αρχείο για τον πίνακα.

3. Να κατασκευαστεί πρόγραμμα διαχείρισης τραπεζικών λογαριασμών το οποίο:

- Στην κύρια συνάρτηση θα κατασκευάζει 2 πίνακες έναν με τα id των πελάτων(String) και έναν με τα υπόλοιπα των λογαριασμών του(double). Τα δεδομένα σε αυτούς θα εισάγονται από ένα αρχείο της μορφής(id-υπολοιπο).
- Να κατασκευαστεί συνάρτηση που θα κάνει κατάθεση στον λογαριασμό του πελάτη, η οποία θα ζητάει από τον χρήστη να εισάγει το id του και το ποσό κατάθεσης. Αν τον id δεν αντιστοιχεί σε έγκυρο id να εμφανίζει κατάλληλο μήνυμα, αλλιώς να κάνει κατάθεσι τον χρημάτων.
- Να κατασκευαστεί συνάρτηση που θα κάνει ανάληψη των χρημάτων από έναν λογαριασμό, η οποία θα ζητάει από τον χρήστη να εισάγει το id του και το ποσό κατάθεσης. Αν τον id δεν αντιστοιχεί σε έγκυρο id να εμφανίζει κατάλληλο μήνυμα. Αν το id είναι έγκυρο να έλεγχει αν το ποσό ανάληψης είναι μεγαλύτερο από το υπόλοιπο του λογαριασμού και να εμφανίζει το κατάλληλο μήνυμα αλλιώς να πραγματοποιεί την ανάληψη.
- Να κατασκευαστεί συνάρτηση η οποία θα αποθηκεύει τα δεδομένα σε ένα αρχείο.