**Ministerul Educației și Cercetării**

|  |
| --- |
| **Universitatea Tehnica a Moldovei** |

**Facultatea Calculatoare, Informatica si Microelectronica**

**Structuri de date și Algoritmi**

**Raport**

|  |
| --- |
| Lucrarea de Laborator nr. 3  Tema: Algoritmi de prelucrare a fișierelor text |

|  |  |
| --- | --- |
| A efectuat stundent(ul/a) grupei: | SI-212 |
| Numele si prenumele elev(ului/ei): | Vozian Vladimir |
|  |  |
| Profesorul: | Mititelu Vitalie |

|  |
| --- |
| Chișinău 2022 |

**Scopul lucrării:**

Programarea algoritmilor de prelucrare a fișierelor text prin utilizarea funcțiilor, pointerilor, alocării dinamice a memoriei în limbajul C.

**Sarcina:**

Pentru fișierele text de efectuat sarcina specificată în varianta de mai jos:

**Varianta:**

16.Să se scrie un fișier de comenzi care are ca parametri două cuvinte și o listă de fișiere. Să se construiască un fișier prin concatenarea fișierelor din listă după ce se aplică următoarele modificări: pentru fișierele de rang impar din listă se vor înlocui toate aparițiile primului cuvânt cu al doilea, iar pentru cele de rang par, se va înlocui ultima apariție din fiecare linie a celui de-al doilea cuvânt cu primul.

**Rezumat**

Un fişier este o structură dinamică, situată în memoria secundară (pe disk-uri). Limbajul C permite operarea cu fişiere:

* de **tip text** - un astfel de fişier conţine o succesiune de **linii**, separate prin new line ('\n')
* de **tip binar** - un astfel de fişier conţine o succesiune de octeti, **fără nici o structură**.

Prelucrarea unui fişier presupune asocierea acestuia cu un canal de I/E (numit flux sau stream). Există trei canale predefinite, care se deschid automat la lansarea unui program:

* **stdin** - fişier de intrare, text, este intrarea standard - tastatura
* **stdout** - fişier de iesire, text, este ieşirea standard - ecranul monitorului.
* **stderr** – fişier de iesire, text, este ieşirea standard unde sunt scris mesajele de eroare - ecran.

Pentru a prelucra un fişier, trebuie parcurse următoarele etape:

* **se defineşte** o variabilă de tip **FILE**\* pentru accesarea fişierului; **FILE** este un tip structură definit în <stdio.h>, care conţine informaţii referitoare la fişier şi la tamponul de transfer de date între memoria centrală şi fişier (adresa, lungimea tamponului, modul de utilizare a fişierului, indicator de sfârsit, de poziţie în fişier). Puteți citi mai multe [aici](http://stackoverflow.com/questions/5672746/what-exactly-is-the-file-keyword-in-c).
* **se deschide fişieru**l pentru un anumit **mod de acces**, folosind funcţia de bibliotecă **fopen**, care realizează şi asocierea între variabila fişier şi numele extern al fişierului
* **se prelucrează fişierul** în citire/scriere cu **funcţiile specifice**
* **se închide fişierul** folosind funcţia de bibliotecă **fclose**

#### **Funcții**

##### fopen

FILE \*[fopen](http://www.opengroup.org/onlinepubs/009695399/functions/fopen.html)(const char \*filename, const char \*mod);

deschide fişierul cu numele **filename** pentru acces de tip **mod**.

Returnează **pointer la fişier** sau **NULL** dacă fişierul nu poate fi deschis; valoarea returnată este memorată în variabila fişier, care a fost declarată pentru accesarea lui.

Modul de deschidere poate fi:

* “**r**” - **readonly** , este permisă doar citirea dintr-un fişier existent
* “**w**” - **write**, crează un nou fişier, sau dacă există deja, distruge vechiul continut
* “**a**” - **append**, deschide pentru scriere un fişier existent ( scrierea se va face în continuarea

informaţiei deja existente în fişier, deci pointerul de acces se plasează la sfârşitul fişierului )

* “**+**” - permite scrierea şi citirea - **actualizare** (ex: “r+”, “w+”, “a+”). Între read şi write trebuie repoziţionat cursorul de acces printr-un apel la **fseek**.
* “**b**” - specifică fişier de tip **binar**
* “**t**” - specifică fişier de tip **text** (implicit), la care se face automat conversia CR-LF(“\n\f”) în sau din CR ('\n').

##### fclose

int [fclose](http://www.opengroup.org/onlinepubs/009695399/functions/fclose.html)(FILE \*pFile);

**închide fişierul** asociat cu variabila **pFile** şi eliberează zona tampon; returnează 0 la succes, EOF (end of file) la eroare

##### fseek

int [fseek](http://www.opengroup.org/onlinepubs/009695399/functions/fseek.html)(FILE \*pFile, long offset, int whence);

**repoziţionează pointerul** asociat fişierului **pFile**; offset - numărul de octeţi între poziţia dată de whence şi noua poziţie.

whence - are una din cele trei valori posibile:

* SEEK\_SET = 0 - Căutarea se face de la începutul fişierului
* SEEK\_CUR = 1 - Căutare din poziţia curentă
* SEEK\_END = 2 - Căutare de la sfârşitul fişierului

##### ftell

long [ftell](http://www.opengroup.org/onlinepubs/009695399/functions/ftell.html)(FILE \*pFile);

**întoarce poziţia curentă** în cadrul fișierului asociat cu pFile.

##### fgetpos

int [fgetpos](http://www.opengroup.org/onlinepubs/009695399/functions/fgetpos.html)(FILE \*pFile, fpos\_t \*ptr);

această funcţie **memorează poziţia curentă** în variabila ptr în cadrul fişierului asociat cu pFile (ptr va putea fi folosit ulterior cu funcţia fsetpos).

##### fsetpos

int [fsetpos](http://www.opengroup.org/onlinepubs/009695399/functions/fsetpos.html)(FILE \*pFile, const fpos\_t \*ptr);

această funcţie **setează poziţia curentă** în fişierul asociat cu pFile la valoarea ptr, obţinută anterior prin funcţia fgetpos.

##### feof

int [feof](http://www.opengroup.org/onlinepubs/009695399/functions/feof.html)(FILE \*fis);

returnează 0 dacă nu s-a detectat **sfârşit de fişier** la ultima operaţie de citire, respectiv o valoare nenulă (adevărată) pentru sfârşit de fişier.

##### freopen

FILE\* [freopen](http://www.opengroup.org/onlinepubs/009695399/functions/freopen.html)(const char \*filename, const char \*mode, FILE \*fp);

se închide fişierul fp, se deschide fişierul cu numele filename în modul mode şi acesta se asociază la fp; se întoarce fp sau NULL în caz de eroare.

##### fflush

int [fflush](http://www.opengroup.org/onlinepubs/009695399/functions/fflush.html)(FILE \*fp);

Această funcţie se utilizează pentru fişierele deschise pentru scriere şi are ca efect scrierea în fişier a datelor din bufferul asociat acestuia, care înca nu au fost puse în fişier.

**Codul deplin al programului:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdbool.h>

char\* replaceWord(const char\* s, const char\* oldW, const char\* newW){

    char\* result;

    int i, cnt = 0;

    int newWlen = strlen(newW);

    int oldWlen = strlen(oldW);

    for (i = 0; s[i] != '\0'; i++){

        if (strstr(&s[i], oldW) == &s[i]){

            cnt++;

            i += oldWlen - 1;

        }

    }

    result = (char\*)malloc(i + cnt \* (newWlen - oldWlen) + 1);

    i = 0;

    while (\*s){

        if (strstr(s, oldW) == s){

            strcpy(&result[i], newW);

            i += newWlen;

            s += oldWlen;

        }

        else

            result[i++] = \*s++;

    }

    result[i] = '\0';

    return result;

}

char\* replaceLastLineWord(const char\* s, const char\* oldW, const char\* newW){

    char\* result;

    int i = 0;

    int newWlen = strlen(newW);

    int oldWlen = strlen(oldW);

    int c = 0;

    while(strstr(&s[i], oldW)){

        i++;

    }

    result = (char\*)malloc(strlen(s) - (strlen(oldW) - strlen(newW)));

    i--;

    while(\*s){

        if(c == i){

            strcpy(&result[i], newW);

            i += newWlen;

            s += oldWlen;

        }else

            result[c++] = \*s++;

    }

    result[i] = '\0';

    return result;

}

int main(){

    char word1[50], word2[50];

    int len1,len2;

    char rasp;

    printf("Dati primul cuvant parametric=> ");

    fflush(stdin);

    scanf("%s", &word1);

    printf("\nDati al doilea cuvant parametric=> ");

    fflush(stdin);

    scanf("%s", &word2);

    len1 = strlen(word1);

    len2 = strlen(word2);

    char \*fisier = "finalF.txt";

    FILE \*f;

    f = fopen(fisier, "w");

        char \*fisier1 = "primulF.txt";

        FILE \*g;

        char string[250];

        char\* result = NULL;

            g = fopen(fisier1, "r");

            fgets(string, 250 ,g);

            printf("%s", string);

            fclose(g);

        result = replaceWord(string, word1, word2);

        fwrite(result, 1, strlen(result), f);

        char \*fisier2 = "doiF.txt";

        FILE \*h;

        result = NULL;

            h = fopen(fisier2, "r");

            fgets(string, 250 ,h);

            printf("%s", string);

            fclose(h);

        fwrite("\n", 1, strlen("\n"), f);

        result = replaceLastLineWord(string, word2, word1);

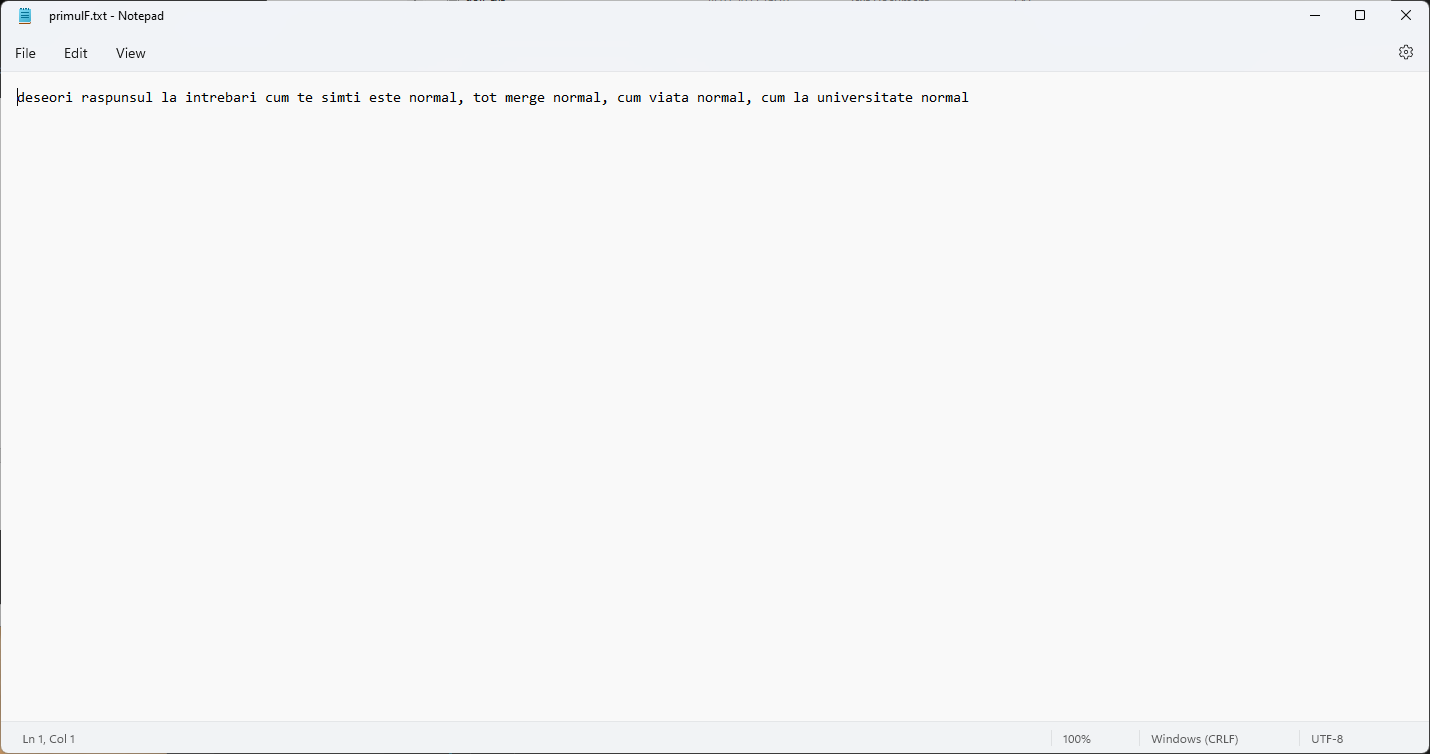
        fwrite(result, 1, strlen(result), f);

fclose(f);

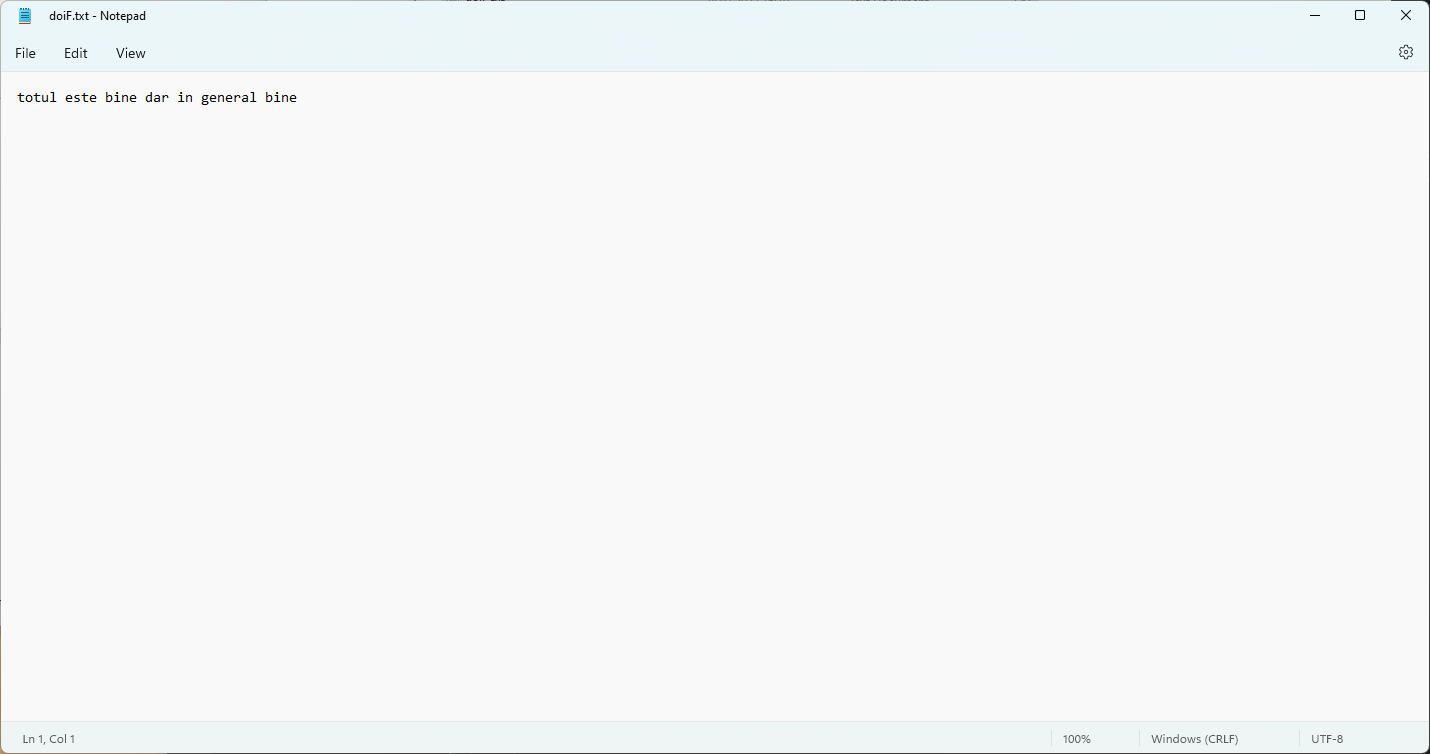
}

**Exemple de execuție a programului:**

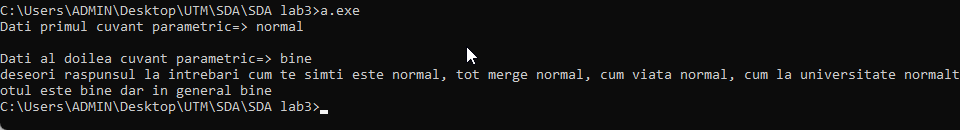
Textul inițial din primul fișier



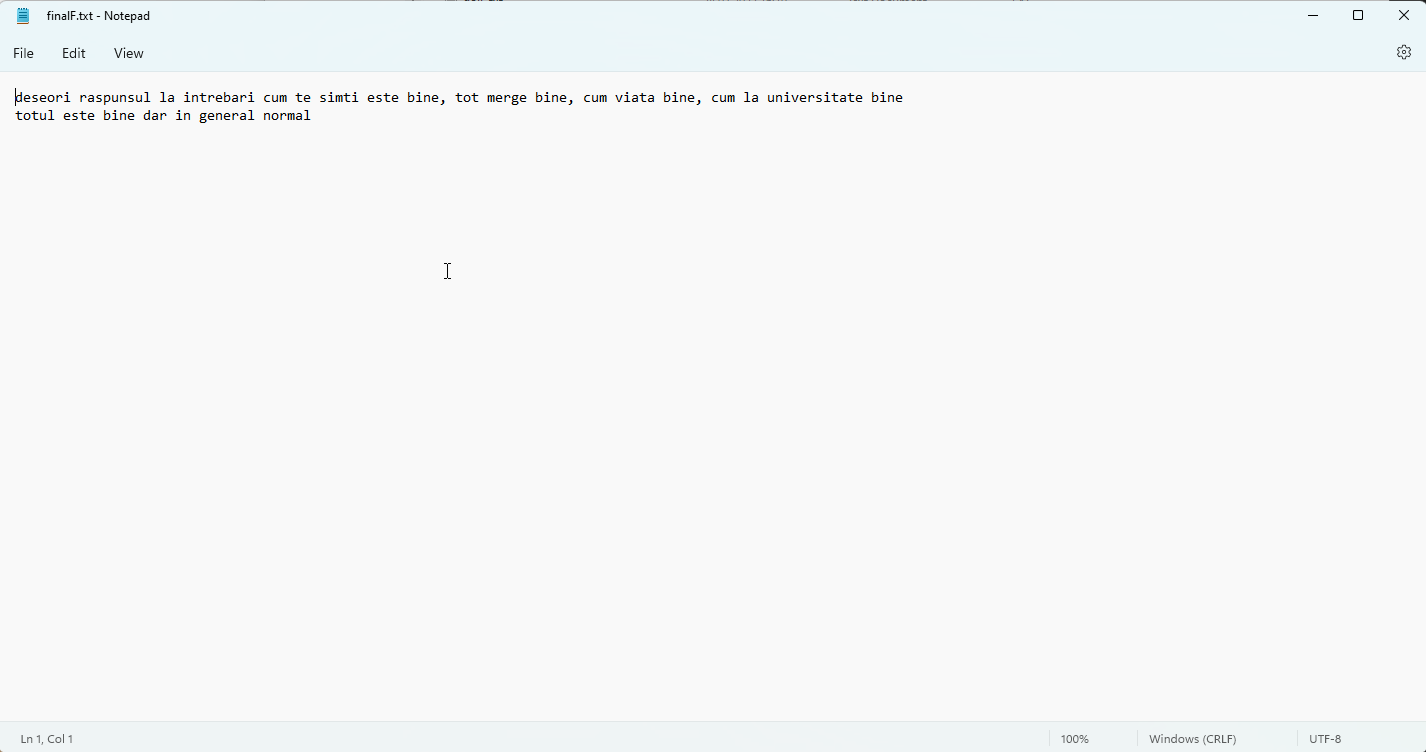
Textul inițial din al doilea fișier



Ca primul cuvânt parametric a fost dat cuvântul normal, iar al doilea a fost bine.



Acesta este și rezultatul, concatenarea celor 2 fișiere de mai sus modificate conform cerinței, pentru fișierele de rang impar din listă se vor înlocui toate aparițiile primului cuvânt cu al doilea, iar pentru cele de rang par, se va înlocui ultima apariție din fiecare linie a celui de-al doilea cuvânt cu primul. Deci in primul fișier cuvântul normal a fost peste tot înlocuit cu cuvântul bine, iar in al doilea fișier ultima apariție pe linie a cuvântului bine era înlocuită cu normal.



**Concluzii pe baza efectuării lucrării de laborator:**

* Deci sa lucrat cu fișierele text, ele fiind performate și concatenate in unul final conform cerinței.
* Lucrul asupra lucrării este util în întelegerea mai profundă a lucrului cu fișierele și a stocării și operării mai efectivă cu informația.

**Surse**

<https://ocw.cs.pub.ro/courses/programare/laboratoare/lab12>