

PROGRAMARE PROCEDURALĂ – LABORATOR NR. 10 –

1. Fișierul text *produse.csv* conține informații despre produsele dintr-un magazin, respectiv pe fiecare linie avem numele unui produs, cantitatea în care el se află în magazin și prețul său unitar. Informațiile de pe fiecare linie sunt despărțite între ele prin virgule. Să se scrie în fișierul text *stoc.txt* valoarea stocului din fiecare produs, iar pe ultima linie să se scrie stocul total.

CSV (Comma Separated Values) = fișier text în care informațiile sunt despărțite între ele printr-o virgulă

Exemplu:

produse.csv	stoc.txt
lapte,10,3.75	lapte: 37.50
telemea de oaie,3.20,16.80	telemea de oaie: 53.76
paine alba,20,5.25	paine alba: 105.00
	TOTAL: 196.26

Rezolvare:

```
#include<stdio.h>
#include<string.h>

int main ()
{
    char sir[1000],*p;
    int cant_p;
    float pret_p,suma_p;

    FILE *fin, *fout;

    fin = fopen("produse.csv", "r");
    fout = fopen("stoc.txt","w");
```

```

while(fgets(sir,1000,fin) != NULL)
{
    p=strtok(sir,",");
    fprintf(fout,"%s: ",p);

    p=strtok(NULL,",");
    sscanf(p,"%d",&cant_p);

    p=strtok(NULL, "\n");
    sscanf(p,"%f",&pret_p);
    fprintf(fout,"%0.2f\n",cant_p*pret_p);

    suma_p=suma_p+cant_p*pret_p;
}

fprintf(fout,"\nTOTAL: %0.2f",suma_p);

fclose(fin);
fclose(fout);

return 0;
}

```

2. Scrieți un program care calculează suma totală cheltuită la cumpărături, pentru un fișier text *raport.txt* de forma următoare: "Astăzi am cumpărat 5.50 kg de mere cu 2.20 RON kilogramul și 2 litri de lapte cu 4.30 RON litrul.". Suma totală se va scrie la sfârșitul fișierului, pe o linie nouă.

Rezolvare:

```

#include<stdio.h>

int main ()
{
    char cuvant[51];
    float cant_p, pret_p, total, nr_crt;
    int pozitie = 0;

    FILE *f = fopen("raport.txt", "r");

    total = 0;
    while(fscanf(f, "%s", cuvant) == 1)
        if(sscanf(cuvant, "%f", &nr_crt) == 1)
        {
            pozitie++;

```

```

        if(pozitie % 2 != 0)
            cant_p = nr_crt;
        else
        {
            pret_p = nr_crt;
            total = total + cant_p*pret_p;
        }
    }

    fclose(f);

    f = fopen("raport.txt", "a");

    fprintf(f, "\n\nTOTAL: %.2f RON", total);

    fclose(f);

    return 0;
}

```

3. Scrieți un program care afișează pe ecran toate cuvintele distincte dintr-un fișier text în ordinea descrescătoare a lungimilor. Dacă mai multe cuvinte au aceeași lungime, ele se vor afișa în ordine alfabetică. Presupunem că fișierul conține cel mult 500 de cuvinte și orice cuvânt este format din cel mult 20 de litere.

Rezolvare:

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main () {

    char fisier[101];
    char cuvinte[500][21];
    char linie[11000];
    int nrCuv = 0, i, j;
    FILE* finput;

    printf("Introduceti numele fisierului: \n");
    scanf("%s", fisier);

    finput = fopen(fisier, "r");

```

```

while (fgets(linie, 11000, finput))
{
    char* cuvant;
    cuvant = strtok(linie, " .,:;?!\\n");

    while (cuvant != NULL)
    {
        for (i = 0; i < strlen(cuvant); i++)
            cuvant[i] = tolower(cuvant[i]);

        for (i = 0; i < nrCuv; i++)
            if (strcmp(cuvant, cuvinte[i]) == 0)
                break;

        if (i == nrCuv) {
            strcpy(cuvinte[nrCuv], cuvant);
            nrCuv++;
        }

        cuvant = strtok(NULL, " .,:;?!\\n");
    }
}

fclose(finput);

for (int i = 0; i < nrCuv; i++)
    for (int j = i + 1; j < nrCuv; j++)
        if (strlen(cuvinte[i]) < strlen(cuvinte[j]))
        {
            char aux[21];
            strcpy(aux, cuvinte[i]);
            strcpy(cuvinte[i], cuvinte[j]);
            strcpy(cuvinte[j], aux);
        }
    else
        if (strlen(cuvinte[i]) == strlen(cuvinte[j]))
            if (strcmp(cuvinte[i], cuvinte[j]) > 0)
            {
                char aux[21];
                strcpy(aux, cuvinte[i]);
                strcpy(cuvinte[i], cuvinte[j]);
                strcpy(cuvinte[j], aux);
            }

printf("Cuvintele ordonate:\\n");
for (int i = 0; i < nrCuv; i++)
    printf("%s\\n", cuvinte[i]);

```

```

    return 0;
}

```

4. Scrieți un program care afișează toate perechile de cuvinte dintr-un fișier text care rimează între ele. Două cuvinte rimează dacă au ultimele două litere egale.
5. Fișierul *cuvinte.in* conține pe prima linie un cuvânt w format din $n \geq 1$ litere mici din alfabetul englez, iar pe următoarele linii un text în care cuvintele sunt despărțite prin spații și semnele de punctuație uzuale. Realizați un program care să scrie în fișierul text *cuvinte.out* toate pozițiile unde apare cuvântul w în fișierul *cuvinte.in* sau mesajul "Imposibil" dacă în fișierul de intrare nu apare deloc cuvântul w . O poziție cuvântului w în textul dat se va indica sub forma unei perechi (l, c) , unde l reprezintă numărul de ordine al liniei pe care apare cuvântul w , iar c indică al câtelea cuvânt este w pe linia l . Atenție, cuvântul w poate să apară de mai multe ori pe aceeași linie!

Exemplu:

cuvinte.in	cuvinte.out
<p>prune</p> <p>Ana are mere, pere</p> <p>si prune, sapte prune moi, dar</p> <p>sigur mai putine mere si</p> <p>pere decat prune...</p>	<p>(2,2)</p> <p>(2,4)</p> <p>(4,3)</p>