

Programiranje i programski jezici



Zadatak 1:

Napisati program koji u datoteku se upisuje prvih 10 prirodnih brojeva, a zatim se iz iste datoteke citaju brojevi dok se ne stigne do kraja i ispisuju se na standardni izlaz.

```
#include <stdio.h>
#include <stdlib.h>
main() {
    int i;
    FILE* f = fopen("podaci.txt", "w");
    if (f == NULL) {
        printf("Greska prilikom otvaranja datoteke podaci.txt za pisanje\n");
        exit(1);
    }

    for (i = 0; i<10; i++) fprintf(f, "%d\n", i);
    fclose(f);
}
```

Zadatak 1:

```
f = fopen("podaci.txt", "r");
if (f == NULL) {
    printf("Greska prilikom otvaranja datoteke podaci.txt za
citanje\n");
    exit(1); }
while(1) {
    int br;
    fscanf(f, "%d", &br);
    if (feof(f)) break;
    printf("Procitano : %d\n", br); }
fclose(f);
}
```

Funkciju **feof** ne bi trebalo pozivati pre pokusaja citanja. Sledeci kod moze dovesti do greske:

```
while (!feof(f)) scanf("%d",&br);
```

Zadatak 2:

Napisati program koji u datoteku dadaje tekst.

```
#include <stdio.h>
main() {
    FILE* datoteka;
    if ( (datoteka=fopen("dat.txt","a"))==NULL) {
        fprintf(stderr,"Greska : nisam uspeo da otvorim dat.txt\n");
        return 1;
    }
    fprintf(datoteka,"Zdravo svima\n");
    fclose(datoteka);
}
```

Zadatak 3:

NCP koji kopira datoteku cije se ime zadaje kao prvi argument komandne linije u datoteku cije se ime zadaje kao drugi argument komandne linije. Uz svaku liniju se zapisuje i njen broj.

```
#include <stdio.h>
#define MAX_LINE 256
int getline(char s[], int lim) {
    char* c = fgets(s, lim, stdin);
    return c==NULL ? 0 : strlen(s);
}
main(int argc, char* argv[]) {
    char line[MAX_LINE];
    FILE *in, *out;
    int line_num;
    if (argc != 3) {
        fprintf(stderr, "Upotreba : %s ulazna_datoteka
izlazna_datoteka\n", argv[0]);
        return 1;
    }
```

Zadatak 3:

```
if ((in = fopen(argv[1], "r")) == NULL) {
    fprintf(stderr, "Neuspesno otvaranje datoteke %s\n",
argv[1]);
    return 1;
}
if ((out = fopen(argv[2], "w")) == NULL) {
    fprintf(stderr, "Neuspesno otvaranje datoteke %s\n", argv[2]);
    return 1;
}
line_num = 1;
while (fgets(line, MAX_LINE, in) != NULL) {
    fprintf(out, "%-3d : \t", line_num++);
    fputs(line, out);
}
fclose(in);
fclose(out);
}
```

Zadatak 4:

Definisati strukturu `prozor_vrata` koja sadrži sledeće podatke navedene podatke. Vrata se prave od dasaka fiksne širine 30cm i dovoljne dužine, nadovezivanjem dasaka. Ostatak daske koji se dobije pri pravljenju vrata ne može da se upotrebi za druga vrata. Program sadrži funkciju `Vrata` kojom se određuje broj dasaka potreban za pravljenje svih vrata. Program sadrži i funkciju `Prozor` koja određuje najmanju površinu prozora. U glavnom programu učitati podatke i ispisati vrednosti dobijene iz funkcija `Prozor` i `Vrata`.

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

```
struct prozor_vrata{
    char tip[6];
    float sirina,visina;};
```

```
int Vrata(struct prozor_vrata vp[],int n){
    int i,d=0;
    float k;
    for(i=0;i<n;i++)
        if(!strcmp(vp[i].tip,"VRATA")) {
            k=vp[i].sirina/30;
            if(k==(int)k) d=d+(int)k;
            else d=d+(int)k+1;
        }
    return d;
}
```

```
float Prozor(struct prozor_vrata vp[],int n){
    int i=0;
    float min=0;
    while(i<n && strcmp(vp[i].tip,"PROZOR")) i++;
    if(i<n) {
        min=vp[i].sirina*vp[i].visina;
        for(;i<n;i++)
            if(vp[i].sirina*vp[i].visina<min)
min=vp[i].sirina*vp[i].visina;
    }
    return min;
}
```

```
main()
struct prozor_vrata vp[100];
int i,n=0;
char s[6];
float s1,v1;
scanf("%s",s);
while(s!='\0'){
    strcpy(vp[n].tip,s);
    scanf("%f%f",&s1,&v1);
    vp[n].sirina=s1;
    vp[n++].visina=v1;

    scanf("%s",s);
}

printf("%d\n",Vrata(vp,n));
printf("%.2f\n",Prozor(vp,n));
}
```

Zadatak 5:

Napisati program u C-u koji kao parametar pri startovanju programa dobija ime tekstualne datoteke sa podacima o ugovorima sa firmama. Definirati strukturu firma koja sadrži sledeće podatke: ime firme, vrednost ugovora. U navedenoj datoteci se u prvoj liniji nalazi broj ugovora, a zatim se u svakom redu nalazi po jedan podatak o ugovorima (ime firme, vrednost). Program sadrži funkciju Ugovori koja dobija naziv firme a vraća ukupnu vrednost svih ugovora potpisanih sa tom firmom. U glavnom programu učitati podatke i za svaku firmu ispisati ukupnu vrednost ugovora.