DATOTEKE (FAJLOVI)

Strukture

```
struct Osoba
 { char ime [20];
  char prezime[20];
  int godiste;
char telefon[10];
 };
struct Osoba pera, mika, zika;
```

Strukture koristimo kada opisujemo složene pojmove, npr. osobu za koju navodimo dosta podataka poput imena, prezimena, godišta, telefona i drugih. Podaci mogu biti ugrađenih tipova koji su razmotreni u prethodnim poglavljima ili nove strukture sa svojim podacima.

Ovako definisane osobe *pera, mika* i *zika* imaju svoja imena, prezimena i ostale podatke koji su definisani u strukturi osoba, a za pristup pojedinačnim podacima koristi se tačka, npr. mika.ime, pera.telefon ili zika.godiste. Ovo je prirodan način jer tačno znamo kojoj osobi pripada koji detalj i ne postoji mogućnost mešanja.

Strukture takođe podržavaju operator dodele kao i drugi tipovi, pa zato možemo da napišemo mika=zika. Takođe, možemo dobiti adresu pomoću operatora &. Strukture nadalje možemo da koristimo kao i sve druge ugrađene tipove.

Strukture i typedef

Nepraktično je svaki put pisati struct Osoba kada treba da deklarišemo neke nove promenljive tipa Osoba. Srećom postoji elegantan način da se to zaobiđe pomoću *typedef.*

```
typedef struct
 { char ime[20];
   char prezime[20];
   int godiste;
   char telefon[10];
 } Osoba;
Osoba pera, mika, zika
```

Fajlovi

Fajlovi su bitni u svakom programskom jeziku jer omogućavaju da trajno sačuvamo vrednosti i time omogućimo kasniji nastavak rada sa tim istim podacima. Zanimljiv primer je adresar koji ne bi bio praktičan ako bismo pri svakom pokretanju sve podatke ponovo unosili.

Fajl otvaramo funkcijom fopen koja nam vraća pokazivač na strukturu **FILE.** Sve ostale funkcije za rad sa fajlovima koriste taj pokazivač umesto imena fajla. Sve funkcije osim fopen vraćaju int vrednost.

Mnoge funkcije za rad sa fajlovima (videti u tabeli) vraćaju konstantu EOF (End Of File) ukoliko nastane greška pri izvršavanju ili ako se dođe do kraja fajla.

Funkcija	**opis**
fopen(ime, rezim)	Otvaranje fajla datog imena. Režim "w"- pisanje, "r" - čitanje, "a" - dodavanje
fclose(f)	Zatvaranje fajla. (rezultat 0=uspešno, EOF neuspešno)
feof(f)	Ispitivanje da li je došlo do kraja fajla. (rezultat 0=nije kraj, >0 kraj)
fseek(f, kol, odakle)	Pozicioniranje. Odakle: SEEK _SET od početka, SEEK_CUR od trenutne pozicije i SEEK_END od kraja fajla
ftell(f)	Trenutna pozicija u fajlu. (viaća int vrednost ili -1 za grešku)
ferror(f)	Ispitivanje greške u prethodnoj operaciji. (0=nema greške)
getc(f)	Citanje znaka iz fajla. (vraća int ili EOF u slučaju greške)
putc(znak, f)	Upisivanje znaka u fajl. (isti karakter ili EOF u slučaju greške)

fgetc(f)	Čitanje znaka iz fajla. (isto kao getc**)**
fputc(znak, f)	Upisivanje znaka u fajl. (isto kao putc)
fgets(f, kol., string)	**Čitanje stringa iz fajla**. NULL ako nema više podataka.
fputs(string, f)	Upisivanje stringa u fajl. (vraća >0 ili EOF u slučaju greške)
fscanf(f,maska,prom.)	Čitanje promenljivih iz fajla. Vraća broj pročitanih znakova, negativnu vrednost za grešku ili EOF ako nema podataka**.**
fprintf(f,maska,prom.)	Upisivanje promenljivih u fajl. Vraća broj upisanih znakova il negativnu vrednost u slučaju greške
fflush(f)	Pražnjenje bafera. (0=uspešno, EOF neuspešno)

Tok je izvor ili odredište podatka, on se najčešće povezuje sa diskom, ekranom ili nekim drugim periferijskim uređajem. Tok se povezuje sa fajlom ili uređajem tako što prvo otvori tok, a na kraju ga zatvori. Nakon svakog pokretanja programa, automatski se otvaraju tri toka: **stdin, stderr, stdout.**
Tok **stdin** se vezu za **tastaturu**, a **stdout i stderr** za **ekran**. Sledeća dva izraza su ekvivalentna:

```
printf (" Zdravo ! ") ;     <=>     fprintf (stdout, " Zdravo!") ;
```

Funkcije system() i exit()

Funkcija *system* omogućava da izvršimo neku sistemsku funkciju poput npr. kopiranja ili brisanja fajla ili izrade direktorijuma. Dobar primer je system(''PAUSE'') koji zaustavlja rad programa dok korisnik ne pritisne neki taster.
Funkcija *exit* dovodi do prekida izvršavanja programa. Ona zatvara sve fajlove otvorene pomoću fopen. Kao indikator načina završetka programa mogu se koristiti vrednosti EXIT_FAILURE (nastala je greška) i EXIT_SUCCESS (normalan završetak programa).
Obe funkcije su definisane u stdlib.h.

Otvaranje datoteke:

```
FILE * fp;
fp = fopen("hello.txt", "w");
if( fp == NULL)
printf("Greska pri otvaranju datoteke");
```

mod	značenje
"r"	Otvori datoteku za čitanje (eng. read). Ako datoteka ne postoji fopen() vraća NULL.
"w"	Otvori datoteku za pisanje (eng. write). Ako ne postoji datoteka zadanog imena kreira se nova datoteka. Ako postoji datoteka zadanog imena njen se sadržaj briše i kreira prazna datoteka (novi podaci će se zapisivati počevši od početka datoteke).
"a"	Otvori datoteku za dopunu sadržaja (eng. append) . Ako ne postoji datoteka zadanog imena kreira se nova datoteka. Ako postoji datoteka zadanog imena, novi podaci će se dodavati na kraj datoteke.
"r+"	Otvori datoteku za čitanje i pisanje . Ako ne postoji datoteka zadanog imena kreira se nova datoteka. Ako postoji datoteka zadanog imena njen se sadržaj briše i kreira prazna datoteka (novi podaci će se zapisivati od početka datoteke)..
"w+"	Isto kao r+
"a+"	Otvori datoteku za čitanje i dopunu . Ako ne postoji datoteka zadanog imena kreira se nova datoteka. Ako postoji datoteka zadanog imena u nju se vrši upis na kraju datoteke..
"b"	Ako se iza slova w, r ili a još zapiše slovo 'b' to označava da se datoteku treba otvoriti u binarnom modu, inače se datoteka otvara u tekstualnom modu.

rb, wb, ab, rb+,wb+, ab+ ili r+b, w+b…

DATOTEKE ZADACI

//1.Provera da li postoji datoteka druga na disku D, otvori, ispise na monitoru i zatvori.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *f;
char *n="D:/druga.txt";
char c;
if( (f=fopen(n,"r")) == NULL ){
printf("Datoteka <%s> nije uspesno otvorena.\n", n);
exit(1);
}
printf("Datoteka <%s> je uspesno otvorena.\n", n);
printf(">>Tekstualni sadrzaj datoteke <%s>...\n", n);
while((c=fgetc(f))!=EOF)
        putchar(c);
printf("\n>>Kraj sadrzaja datoteke <%s>...\n", n);
fclose(f);
return 0;
}
```

//2 .datoteka ulaz.txt sadrzi cele brojeve. Izracunati njihov zbir i zapisati u izlaznu datoteku izlaz.txt.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *ul,*iz;
int a,zbir=0;
ul=fopen("F:/ulaz.txt","r");
iz=fopen("F:/izlaz.txt","w");
if (ul==NULL)
  {fprintf(stderr,"Neuspesno  otvaranje ulaz.txt");          // ili  fprintf(stdout,"Neuspesno  otvaranje ulaz.txt");
   exit(EXIT_FAILURE);                                       //ili  printf("Datoteka nije uspesno otvorena.\n"); exit(1);
  }
if (iz==NULL)
  {fprintf(stderr,"Neuspesno otvaranje izlaz.txt");
   exit(EXIT_FAILURE);
  }
while(fscanf(ul,"%d",&a)!=EOF)   zbir+=a;
fprintf(iz,"%d",zbir);
fclose(ul);
fclose(iz);
return 0;
}
```

//3.provera da li je datoteka prva uspesno otvorena i zatvorena i da na ekranu ispise sadrzaj datoteke prva

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *prva;
char c;
if( (prva=fopen("prva","r")) == NULL )   //da smo hteli da kreiramo datoteku umesto "r" pisali bi "w"
{
printf("Datoteka nije uspesno otvorena.\n");
exit(EXIT_FAILURE);
}
printf("Datoteka  je uspesno otvorena.\n");
while((c=fgetc(prva))!=EOF)
    putchar(c);
printf("Zatvaranje datoteke...\n");
if( fclose(prva)==EOF )
printf("\tNastupila je greska tokom zatvaranja!\n");
else
printf("\tZatvaranje uspesno!\n");
return 0;
}
```

//4.provera da li je datoteka prva uspesno otvorena i zatvorena i da na ekranu ispise sadrzaj stringa max duzine 20 sa fgets i puts

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *f;
char *n="F:/prva.txt";
char c[20];
if( (f=fopen(n,"r")) == NULL )   //da smo hteli da kreiramo datoteku umesto "r" pisali bi "w"
{
printf("Datoteka <%s> nije uspesno otvorena.\n", n);
exit(EXIT_FAILURE);
}
printf("Datoteka <%s> je uspesno otvorena.\n", n);
while(fgets(c,20,f)!=NULL)
    puts(c);
printf("Zatvaranje datoteke <%s>...\n", n);
if( fclose(f)==EOF )
printf("\tNastupila je greska tokom zatvaranja!\n");
else
printf("\tZatvaranje uspesno!\n");
return 0;
}
```

// 5. zad. Demonstrira se formatiran upis u tekstualnu datotke 5 realnih brojeva, koje unosi korisnik

```
#include <stdlib.h>
#include <stdio.h>
int main()
{
FILE *fp;
float data[5];
int i;
char filename[20];
puts("Otipkaj 5 realnih brojeva");
for (i = 0; i < 5; i++)
scanf("%f", &data[i]);
/* Dobavi ime datoteke, ali prethodno */
/* isprazni moguci višak znakova iz međuspremnika ulaza */
fflush(stdin);
puts("Otipkaj ime datoteka:");
gets(filename);
if ( (fp = fopen(filename, "w")) == NULL)
{
fprintf(stderr, "Greska pri otvaranju datoteke %s.",
filename);
exit(1);
}
/*Ispisi vrijednosti u datoteku i na standardni izlaz */
  for (i = 0; i < 5; i++)
          {           fprintf(fp, "\n podatak[%d] = %f", i, data[i]);
                      fprintf(stdout, "\n podatak[%d] = %f", i, data[i]);
           }
fclose(fp);
printf("\n Sada procitaj datoteku: %s, nekim editorom",filename);
return(0);}
```

Učitan broj u datoteci 67.0001 pročita se kao 67.000999. Očigledno je da se upretvaranju koje se vrši pri formatiranom unosu gubi na tačnosti.

Funkcija fscanf() je pogodna za formatirani unos brojeva, ali nije pogodna za unos stringova i znakova.

//6. Odrediti broj redova u tekstulnom faju čije ime unosi korisnik

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *ul;
int BrRed=0;
char ch,ime[100];
printf("Unesite putanju do fajla:");
scanf("%s",ime);
ul=fopen(ime,"r");
if (ul==NULL)
   {fprintf(stderr,"Neuspesno  otvaranje ulaz.txt");
    exit(EXIT_FAILURE);
   }
while((ch=fgetc(ul))!=EOF)
  if(ch=='\n') BrRed++;
printf("Broj redova je %d.",BrRed);
fclose(ul);
return 0;}
```

//7.Korisnik unosi putanje za 2 fajla, iskopirati sadrzaj prvog u drugi fajl, bez brisanja sadrzaja drugog fajla.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *prva,*druga;
char ch,ime1[100],ime2[100];
printf("Unesite putanju do fajla:");
scanf("%s",ime1);
printf("Unesite putanju do fajla:");
scanf("%s",ime2);
prva=fopen(ime1,"r");
druga=fopen(ime2,"a");
if (prva==NULL|| druga==NULL)
  {fprintf(stderr,"Neuspesno  otvaranje fajlova");
   exit(EXIT_FAILURE);
  }
while((ch=fgetc(prva))!=EOF)
 fputc(ch,druga);
fclose(prva);
fclose(druga);
return 0;
}
```

//8. Program koji demonstrira dodavanje sadržaja u datoteku (funkcija append).

/* Program demonstrira "a" - append mod datoteka - nadovezivanje */

```
#include <stdio.h>

main(){
        FILE* datoteka;
        int godina = 2010;

/* Otvaramo datoteku za nadovezivanje i proveravamo da li je doslo do greske, ako datoteka ne postoji kreira novu */

        if ( (datoteka=fopen("D:/druga.txt","a"))==NULL)
         {
                fprintf(stderr,"Greska : nisam uspeo da otvorim dat.txt\n");
                return 1;
        }

        /* Upisujemo sadrzaj  u datoteku */
        fprintf(datoteka,"Zdravo svima\n");
        fprintf(datoteka,"Sve najbolje u %d. godini!!!\n",godina);
        /* Zatvaramo datoteku */
        fclose(datoteka);
        return 0;
}
```

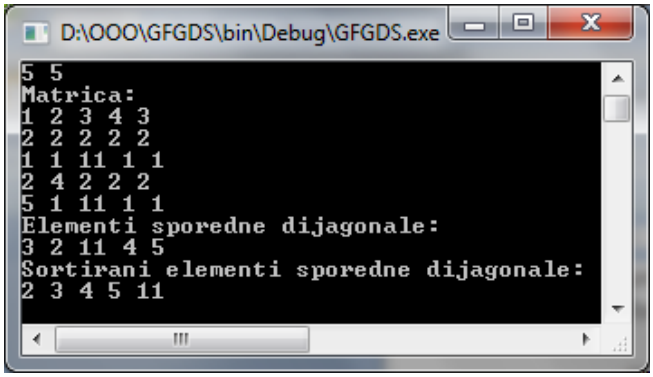
// 9. Napisati program za učitavanje onih redova tekstualne datoteke pod imenom ulaz.txt koji imaju više od 5 reči.

```
#include<stdio.h>
int main(){
	FILE *ulaz;
	int brojac_praznina;
	int i;
	char jedanRed[80];
	ulaz=fopen("D:/TEXT1.txt", "r");
	if(!ulaz){
		printf("Ulazna datoteka nije dostupna!");
		return 1;
	}
	while(!feof(ulaz)){
		fgets(jedanRed, 80, ulaz);
		brojac_praznina=1;
		for(i=0; jedanRed[i]!='\n'; i++)
			if(jedanRed[i]==' ')
				brojac_praznina++;
		if(brojac_praznina>5)
			printf("%s", jedanRed);
	}
	fclose(ulaz);
	return 0;
}
```

//10.Napisati program kojim se sadržaj fajla test.dat formiran od velikih slova alfabeta šifriran šalje u fajl sifra.dat. Znak se šifrira tako što se zamenjuje sledećim ASCII znakom, a znak Z zamenjuje sa A. NAPOMENA: zadatak radi samo sa VELIKIM slovima.

```
#include<stdio.h>
int main(int argc, char *argv[])
{
	int c;
	FILE *inf,*outf;
	inf=fopen("D:/test.txt","r");
	outf=fopen("D:/sifra.txt","w");
	while((c=fgetc(inf))!=EOF)
	{
		if('A'<=c && c<'Z')
			c++;
		else c='A';
		putc(c,outf);
	}
	fclose(inf);
	fclose(outf);
	return 0;
}
```

//11. U datoteci pod imenom matrica.txt je matrica u sledećem format: dimenzija matrice m n pa u sledecim redovima vrednosti elemenata po vrstama. Napisati program koji čita vrednosti elemenata matrice iz datoteke MATRICA.TXT i vrši sortiranje elemanata na sporednoj dijagonali u neopadajućem redosledu i tako dobijen niz upisuje u datoteku SORTIRAN.TXT.




```
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[]){
	int a[10][10],b[10],m,n,i=0,j=0;
	FILE *MATRICA;
	FILE *SORTIRAN;

	//otvaranje datoteke MATRICA
	if((MATRICA=fopen("D:/MATRICA.txt","r"))==NULL)
		printf("Datoteka ulaz.dat nije otvorena.");

	//kreiranje datoteke SORTIRAN
	if((SORTIRAN=fopen("D:/SORTIRAN.txt","w"))==NULL)
		printf("Datoteka izlaz.dat nije otvorena.");

	//ucitavanja dimenija matrice
	fscanf(MATRICA, "%d %d", &m,&n);
	printf("%d %d\n",m,n);

	if(m==n){
		//ucitavanje matrice
		for(i=0;i<n;i++)
			for(j=0;j<m;j++)
				fscanf(MATRICA,"%d",&a[i][j]);

		//ispis matrice
		printf("Matrica:\n");
		for(i=0;i<n;i++)
			for(j=0;j<m;j++){
				printf("%d ",a[i][j]);
				if(j==m-1)
					printf("\n");
			}
		//prepisivanje elemenata sporedne dijagonale u niz b
		for(i=0;i<n;i++)
			for(j=0;j<m;j++)
				if(i+j==m-1)
					b[i]=a[i][j];

		//ispis niza b
		printf("Elementi sporedne dijagonale:\n");
		for(i=0;i<n;i++)
			printf("%d ",b[i]);
		printf("\n");

		//sprtiranje niza b
		for(i=0;i<n-1;i++)
			for(j=i+1;j<n;j++)
				if (b[i]>b[j]){
```

```
                                        int tmp = b[i];
                                        b[i] = b[j];
                                        b[j] = tmp;
                                }
                //ispis sortiranog niza b
                printf("Sortirani elementi sporedne dijagonale:\n");
                for(i=0;i<n;i++)
                        printf("%d ",b[i]);
                printf("\n");
                //upis niza b u datoteku SORTIRAN
                for(i=0;i<n;i++)
                        fprintf(SORTIRAN,"%d ",b[i]);
        }
        else fprintf(SORTIRAN,"%s ","Matrica nije kvadratna");

        fclose(MATRICA);
        fclose(SORTIRAN);
        return 0;
}
```

//11. Program koji demonstrira kretanje po datoteci, tj. korišćenje funkcija koje vraćaju trenutnu poziciju u datoteci kao i funkciju za pozicioniranje na konkretnu poziciju u odnusu na početak, trenutnu poziciju ili kraj datoteke.

```
/* Demonstraacija  ftell() and rewind(). */
#include <stdlib.h>
#include <stdio.h>

#define BUFLEN 6

char msg[] = "trece sest gimnazijalci";

main(){
        FILE *fp;
        char buf[BUFLEN];

        if ( (fp = fopen("D:/TEXT1.txt", "w")) == NULL) {
                fprintf(stderr, "Error opening file.");
                exit(1);
        }

        if (fputs(msg, fp) == EOF) {
                fprintf(stderr, "Error writing to file.");
                exit(1);
        }

        fclose(fp);

        /* Otvaranju fajla za citanje. */

        if ( (fp = fopen("D:/TEXT1.txt", "r+w")) == NULL) {
                fprintf(stderr, "Error opening file.");
                exit(1);
        }
        /*trenutna pozicija u datoteci komandom ftell:
        long int ftell ( FILE * stream );
```

```
         */
        printf("\nOdmah po otvaranju, pozicija = %ld", ftell(fp));

        /* Citanje 5 karaktera. */

        fgets(buf, BUFLEN, fp);
        printf("\nPosle citanja \"%s\", pozicija = %ld", buf, ftell(fp));

        /* Citanje sledecih 5 karaktera */

        fgets(buf, BUFLEN, fp);
        printf("\n\nSledecih 5 karaktera su \"%s\", a pozicija je sad = %ld\n",
                        buf, ftell(fp));

        /* Premotavanje datoteke. */

        rewind(fp);

        printf("\nPosle premotavanja, pozicija se vratila na %ld", ftell(fp));

        /* Citanje 5 karaktera. */

        fgets(buf, BUFLEN, fp);
        printf("\ni citanje ponovo pocinje od pocetka: \"%s\"\n", buf);

        /*
        komanda fseek za pretragu na osnovu pozicije
        int fseek ( FILE * stream, long int offset, int origin );
origin moze biti nesto od sledecih konstanti:
        SEEK_SET        Pocetak fajla
        SEEK_CUR        Trenutna pozicija pokazivaca
        SEEK_END        Kraj fajla
         */

//postavlja pokazivac na 9 poziciju od pocetka fajla
        fseek ( fp , 9 , SEEK_SET );          fputs ( "DODATAK" , fp );

        printf("\nPosle dodavanja, pozicija je %ld \n", ftell(fp));
        fclose(fp);

        return(0);
}
```