

PROGRAMIRANJE II K1 (18.04.2019.) – GRUPA A

- ❶ (5 bodova) Napisati funkciju **concat** koja kao obavezne parametre prihvata pokazivač na funkciju (parametar **f**) i broj **n**, a zatim **n** cijelih brojeva (neobavezni parametri).

Parametar **f** treba da pokazuje na funkciju koja kao parametar prihvata cijeli broj, a kao rezultat vraća vrijednost 0 ako neki uslov nije ispunjen za dati cijeli broj, a vrijednost 1 ako taj uslov jeste ispunjen za dati cijeli broj.

Funkcija **concat** treba da cijele brojeve (neobavezni parametri) za koje je ispunjen neki uslov koji se provjerava korištenjem parametra **f**, pretvori u stringove i spoji ih u jedan dinamički string koji se vraća kao rezultat izvršavanja funkcije. Ako dati uslov nije ispunjen niti za jedan cijeli broj, funkcija treba da vrati **NULL**.

Prototip funkcije **concat** je:

```
char* concat(int (*f)(int), int n, ...);
```

- ❷ (5 bodova) Dat je sljedeći tip podatka:

```
typedef struct {  
    char proizvođač[16], model[16], gorivo[11];  
    int godiste, kilometraza, cijena;  
} VOZILO;
```

Napisati glavnu funkciju (**main**) u kojoj treba iz datoteke pročitati binarno upisane podatke o nepoznatom broju vozila i u tekstualnu datoteku upisati podatke o onim vozilima čija je cijena manja od ili jednaka zadatoj cijeni, a godište veće od ili jednako zadatom godištu. Podaci se u tekstualnu datoteku upisuju u formatiranom obliku.

Argumenti komandne linije:

- prvi – naziv binarne (ulazne) datoteke,
- drugi – naziv tekstualne (izlazne) datoteke
- treći – zadata cijena i
- četvrti – zadato godište.

Primjer tekstualne (izlazne) datoteke:

PROIZVOĐJAC	MODEL	GORIVO	GOD.	KILOM.	CIJENA
Audi	A8	Dizel	2011	173000	49500
Mercedes-Benz	E200	Dizel	2011	200000	25000
Citroen	C4	Dizel	2012	180000	12000

Pretpostaviti da su implementirane sljedeće funkcije za upis header-a i footer-a u tekstualnu datoteku:

```
void header(FILE *dat);  
void footer(FILE *dat);
```

- ❸ (5 bodova) Napisati funkciju **strsort** koja **niz** od **n** stringova sortira u rastućem redoslijedu korištenjem insert-sort algoritma. Pri tome, parametar **k** predstavlja pokazivač na funkciju koja vrši poređenje dva stringa prema nekom kriterijumu po kojem se vrši sortiranje. Funkcija za poređenje dva stringa vraća negativnu vrijednost ako je prvi string manji od drugog, vrijednost 0 ako su stringovi jednaki te pozitivnu vrijednost ako je prvi string veći od drugog. Prototip funkcije je:

```
void strsort(char **niz, int n,  
             int (*k)(const char *, const char *));
```

- ❹ (5 bodova) Napisati rekurzivnu funkciju koja nad cjelobrojnim nizom podataka vrši eksponencijalnu pretragu. Svaki novi segment eksponencijalne pretrage se definiše u novom rekurzivnom pozivu. Ukoliko je tražena vrijednost u traženom opsegu, nastaviti pretragu sekvencijalno. Ukoliko je početak/kraj eksponencijalnog opsega izvan opsega niza, izvršiti sekvencijalnu pretragu u dijelu niza koji je preostao. U slučaju uspješne pretrage funkcija vraća vrijednost 1, a u suprotnom vrijednost 0. Prototip funkcije je:

```
int pretrazi (int *niz, int n, int begin,  
             int end, int kljuc);
```

Pri tome je **niz** – adresa početka niza, **n** – dužina niza, **begin** – početak eksponencijalnog opsega, **end** – kraj eksponencijalnog opsega i **kljuc** – ključ pretrage. Smatra se da su vrijednosti **begin** i **end** unesene ispravno prilikom poziva funkcije i da je niz sortiran.