

0.1 Programiranje 2, primer praktičnog dela ispita

Zadatak 1 Napisati program koji kao prvi argument komandne linije prima ime fajla u kome treba prebrojati sva pojavljivanja traženog stringa (bez preklapanja) koji se navodi kao drugi argument komandne linije (iskoristiti funkciju standardne biblioteke `strstr`). U slučaju bilo kakve greške ispisati -1 na standardni izlaz za greške. Pretpostaviti da linije datoteke neće biti duže od 127 karaktera.

Potpis funkcije `strstr`:

```
char *strstr(const char *haystack, const char *needle);
```

Funkcija traži prvo pojavljivanje podstringa `needle` u stringu `haystack`, i vraća pokazivač na početak podstringa, ili `NULL` ako podstring nije pronađen.

```
Poziv: ./a.out fajl.txt test      ./a.out      ./a.out fajl.txt foo      ./a.out fajl.txt aaa
Fajl:  Ovo je test primer.      (fajl ne postoji)      aaaaaaaaaaa
      U njemu se rec test
      javlja vise puta.
      testtesttest
Izlaz: 5                          -1          -1                          3
```

Zadatak 2 Na početku datoteke `trouglovi.txt` nalazi se broj trouglova čije su koordinate temena zapisane u nastavku datoteke. Napisati program koji učitava trouglove, i ispisuje ih na standardni izlaz sortirane po površini opadajuće (koristiti Heronov obrazac: $P = \sqrt{s * (s - a) * (s - b) * (s - c)}$, gde je s poluobim trougla). U slučaju bilo kakve greške ispisati -1 na standardni izlaz za greške. Ne praviti nikave pretpostavke o broju trouglova u fajlu, i proveriti da li je fajl ispravno zadat.

```
Datoteka: 4                          3                          (nema dateteke)      0
          0 0 0 1.2 1 0                1.2 3.2 1.1 4.3
          0.3 0.3 0.5 0.5 0.9 1
          -2 0 0 0 0 1
          2 0 2 2 -1 -1
Izlaz:    2 0 2 2 -1 -1                -1                          -1
          -2 0 0 0 0 1
          0 0 0 1.2 1 0
          0.3 0.3 0.5 0.5 0.9 1
```

Zadatak 3 Data je biblioteka za rad sa binarnim pretraživačkim stablima celih brojeva. Napisati funkciju

```
int f3(Cvor *koren, int n)
```

koja u datom stablu prebrojava čvorove na n -tom nivou, koji imaju tačno jednog potomka. Pretpostaviti da se koren nalazi na nivou 0. Ispravnost napisane funkcije testirati na osnovu biblioteke za rad sa stablima i zadate main funkcije koja najpre učitava nivo, a zatim i elemente samog stabla.

```
Ulaz:    1 5 3 6 1 4 7 9      2 5 3 6 1 0 4 7 9      0 4 2 5      3      -1 4 5 1 7
Izlaz:   1                          2                          0          0          0
```