

- (fseek, ftell) Napisati program koji u datoteci koja se zadaje kao prvi argument komandne linije zamenjuje sva pojavljivanja niske koja se zadaje kao drugi argument komandne linije niskom koja se zadaje kao treći argument komandne linije.
- Argumenti komandne linije su a b -f ime\_datoteke ili samo a b, tim redosledom. a i b su dve date niske, niska a je kraca od niske b, a ime\_datoteke datoteka u koju treba upisati rezultat. Ukoliko nije zadata, rezultat pisati na standardnom izlazu. Kao rezultat upisati izmenjenu nisku b, tako sto je njen prvi deo zamenjen niskom a. Ukoliko je doslo do greske ispisati -1.  
Primer 1: ./a.out pera2 abrakadabra -f u prva.txt se upisuje: pera2adabra  
Primer 2: prva.txt ./a.out dan leto na standardni izlaz se ispisuje: dano
- Definisati strukturu **Tacka** za predstavljanje tačaka u ravni sa koordinatama tipa **double**.
  - Napisati funkciju **Rastojanje** koja izračunava rastojanje između dve tačke.
  - Napisati funkciju **Obim** koja izračunava obim poligona zadanog nizom tacaka.
  - Napisati funkciju **Povrsina** koja izračunava površinu poligona zadanog nizom tacaka (koristiti Heronov obrazac:  $P = \sqrt{s(s-a)(s-b)(s-c)}$  pri čemu je s poluobim trougla sa stranicama a, b i c).
  - Sa ulaza se unosi broj n i zatim n tačaka koje redom predstavljaju koordinate temena konveksnog prostog poligona. Koristiti dinamičku alokaciju memorije. Program poziva funkcije **Obim** i **Povrsina** i ispisuje na standardni izlaz obim i površinu poligona. Za neispravan unos ili neuspešnu alokaciju memorije program ispisuje broj -1. Pri ispisu vrednosti se zaokružuju na 4 decimale.

Primer 1:

Uneti broj tacaka:

4

Uneti koordinate tacaka:

0 0

0 4

6 4

6 0

Obim: 20.0000

Povrsina: 24.0000

Primer 3:

Uneti broj tacaka:

6

Uneti koordinate tacaka:

-1 1

0 2

1 1

1 -1

0 -2

-1 -1

Obim: 9.6569

Povrsina: 6.0000

Primer 2:

Uneti broj tacaka:

4

Uneti koordinate tacaka:

-1 1

2 3

4 1

0 -7

Obim: 23.4405

Povrsina: 25.0000

Primer 4:

Uneti broj tacaka:

-3

Izlaz: -1

- Definisati strukturu **student** koja opisuje ime i prezime studenta kao i njegov broj poena na ispitu. Napisati funkciju **int ocena(student s)** koja za datog studenta, na osnovu broja poena automatski određuje zaključnu ocenu, na osnovu trenutno važećih pravila na fakultetu. Napisati funkciju **int položili(student\* niz)** koja u nizu studenata određuje ukupan broj studenata koji su položili ispit. Napisati program koji omogućava unos podataka o studentima sa standardnog ulaza (za svakog studenta se unosi ime, prezime i broj poena, odvojeni proizvoljnim brojem belina, a unos se završava kada se unese prazan red), nakon čega se za svakog studenta ispisuje ime prezime i ocena, i ispisuje se ukupan broj studenata koji su položili ispit kao i ukupan broj studenata koji su pali ispit.

Primeri:

Ulaz:

Ulaz:

Ulaz:

Pera Peric 42  
Milos Milanovic 78  
Jelna Milojkovic 98  
Natasa Kuzmanovic 88

Izlaz:

Pera Peric 5  
Milos Milovanovic 8  
Jelna Milojkovic 10  
Natasa Kuzmanovic 9

polozilo: 3  
palo: 1

Jelisaveta Dragicevic 12  
Isidora Velisavljevic 23  
Mika Petrovic 67  
Jadranka Kovacevic 77  
Marko Markovic 14

Izlaz:

Jelisaveta Dragicevic 5  
Isidora Velisavljevic 5  
Mika Petrovic 7  
Jadranka Kovacevic 8  
Marko Markovic 5

polozilo: 2  
palo: 3

Zika Perovic 55  
Isidora Glisic 76  
Djordje Lazarevic 100

Izlaz:

Zika Perovic 6  
Isodora Glisic 8  
Djordje Lazarevic 10

polozilo: 3  
palo: 0

---

Ulaz:

Milos Glogovac 72  
Marija Kocic 99  
Misa Alimpic 46  
Pavle Joksic 87  
Jelena Teovanovic 14  
Ivana Milicevic 62

Izlaz:

Milos Glogovac 8  
Marija Kocic 10  
Misa Alimpic 5  
Pavle Joksic 9  
Jelena Teovanovic 5  
Ivana Milicevic 7

polozilo: 4  
palo: 2