

Formatiranje teksta

Kažemo da je tekst dobro formatiran ako vrijede sljedeća pravila:

1. Riječi su odvojene točno jednim razmakom.
2. Rečenice istog paragrafa odvojene su točno dvama razmacima. (Smatramo da rečenica završava jednim od znakova '.', '!' ili '?', iza kojeg slijedi slijedi barem jedan razmak.)
3. Paragrafi su odvojeni točno jednom praznom linijom.
4. Zadan je N kao maksimalna duljina linija teksta. Niti jedna linija nije dulja od N i za svaku liniju vrijedi da je potpuno ispunjena. Linija L je potpuno ispunjena ako je prazna ili joj se ne može nadopisati početna riječ (ili više njih) iz sljedeće linije, tako da je još uvijek duljine najviše N. Također, linije počinju i završavaju bez praznina.

Napišite sučelje i implementaciju za strukturu **FTekst**. Sučelje treba spremiti u datoteku **ftekst.h**, a implementaciju u datoteku **ftekst.cpp**.

FTekst – varijable, konstruktori i funkcije članice

- **nužne varijable:** string tekst, int N
- **FTekst()** - inicijalizira varijablu tekst na prazan string
- **FTekst(string p)** – inicijalizira varijablu tekst na string p
- **string testPravilo1()** - vraća kao string popis linija, sortiranih uzlazno po broju linije, koje ne zadovoljavaju pravilo br. 1. Za znak razmaka koristiti ";". Dakle, ako linije 1 i 3 ne zadovoljavaju pravilo br. 1 treba vratiti "1;3".
- **string testPravilo2()** - vraća kao string popis paragrafa (rednih brojeva), sortiranih uzlazno po rednom broju paragrafa, koji ne zadovoljavaju pravilo br. 2. Za znak razmaka koristiti ";".
- **string testPravilo3()** - vraća kao string parove paragrafe koji ne zadovoljavaju pravilo br. 3. Za znak razmaka koristiti ";", a kao znak razmaka unutar parova koristiti ",". Npr. "1,2;5,6"
- **string testPravilo4(int duljinalinije)** -vraća kao string popis linija, sortiranih uzlazno po broju linije, koje ne zadovoljavaju pravilo br. 4. Za znak razmaka koristiti ";".
- **string rijeciVanLinije(int duljinalinije)** – vraća riječi koje bi se nalazile u različitim linijama (unutar paragrafa) prije i poslije formatiranja uz N=duljinalinije. Za znak razmaka koristiti ";". U povratnom stringu riječi trebaju biti sortirane po razlici u broju linija prije i poslije formatiranja, a riječi koje imaju istu razliku trebaju biti sortirane leksikografski (zanemariti razliku između velikih i malih slova).
- **int brojRijeciVanLinije(int duljinalinije)** – vraća broj riječi koje bi se nalazile u različitim linijama (unutar paragrafa) prije i poslije formatiranja uz N=duljinalinije
- **void format(int duljinalinije)** - formatira varijablu tekst po pravilima 1-4 uz N=duljinalinije
- **void dodaj(string p)** – dodaj string p na kraj teksta tako da se sačuva formatiranost

Napomene:

- kraj linije označen je znakom "\n"
- u ulaznom tekstu paragrafi su razdvojeni minimalno jednom praznom linijom
- ako kod funkcije dodaj tekst trenutne strukture ne završava s praznom linijom početak teksta koji dodajemo treba dodati kao dio zadnjeg paragrafa prvog teksta
- funkcije testPraviloX vraćaju prazan string ukoliko je sve po pravilu
- riječi definiramo kao sekvencijalni niz znakova različitih od ".", "?" i "!"
- testni primjer nalazi se na Degiorgi forumu pod temom "(2013) 1. zadaca formatiranje teksta".

Opće napomene

- Struktura, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija main()!
- nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od main-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u .h niti u .cpp datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa. No taj klijentski program ne šalžete!

Mole se studenti da ne kontaktiraju asistenta Ugrinu preko maila već da sva pitanja vezana uz zadatak postavljaju na Forumu pod temom "(2013) 1. zadaca formatiranje teksta".