Trigrami

Napišite sučelje i implementaciju za strukture **Trigram i Trigrami**. Struktura Trigram predstavlja niz od 3 znaka ("aaa", "xyz",...), a struktura Trigrami predstavljaju skup trigrama. Sučelje treba spremiti u datoteku **ngrami.h**, a implementaciju u datoteku **ngrami.cpp**.

Trigram – varijable, konstruktori i funkcije članice

- **nužne varijable:** string trigram, int brojPojavljivanja, int pogreska
- **Trigram()** - inicijalizira varijablu trigram na prazan string i varijablu brojPojavljivanja na nulu.
- **Trigram(string p, int bP)** - inicijalizira varijablu trigram na dani string p te brojPojavljivanja na bP. Ako je duljina stringa p razlicita od 3 treba postaviti varijablu pogreska na 1

Trigrami – varijable, konstruktori i funkcije članice

- **nužne varijable:** int brojTrigrama, polje trigrami
- **Trigrami()** - inicijalizira varijablu brojTrigrama na nulu
- **Trigrami(string nizZnakova)** – radi isto što i funkcija postavi
- **void postavi(string nizZnakova)** - za dani nizZnakova pronalazi sve trigrame u tom nizu. Trigrame sprema u varijablu članicu trigrami kao polje struktura Trigram.
- **string kaoString()** - vraća sve pronađene trigrame kao string. Za znak razmaka koristiti ";". Trigrami trebaju biti sortirani silazno po broju pojavljivanja. Dakle, ako su najduži pronađeni trigrami "aaa" (4 puta) i "aab" (7 puta) treba vratiti string "aab;aaa" Ako dva trirgama imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **Trigrami leksikografski()** – vraća strukturu Trigrami popunjenu leksikografski (kao u riječniku) sortiranim trigramima iz trenutne strukture.
- **string distribucija(int n)** – vraća string sastavljen od znakova iz n najčešćih trigrama sortiranih po frekvenciji pojavljivanja u najčešćim trigramima. Ako dva znaka imaju istu frekvenciju pojavljivanja unutar n najčešćih trigrama treba ih sortirati leksikografski. Primjer; za trigrame "aaa", "bbb", "cxx", "vvv" s učestalosti pojavljivanja 3,2,2,1 za distribucija(2) treba vratiti "abxc".
- **Trigrami zajednicki(string nizZnakova)** – vraća strukturu Trigrami koja sadrži sve trigrame zajedničke za nizZnakova i trenutnu strukturu. Trigrami u strukturi trebaju biti sortirani silazno po broju pojavljivanja u stringu nizZnakova. Ako dva trirgama imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **Trigrami zajednicki(Trigrami P2)** – vraća strukturu Trigrami koja sadrži sve trigrame zajedničke za P2 i trenutnu strukturu. Trigrami u strukturi trebaju biti sortirani silazno po broju pojavljivanja u strukturi P2. Ako dva trirgama imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **string oznaci(string nizZnakova)** – za dani nizZnakova (dan malim slovima) označi velikim slovima sva prva i zadnja pojavljivanja trigrama iz trenutne strukture. Ostali znakovi moraju ostati zapisani kao mali.

Napomene:

- Treba zanemariti moguće praznine između znakova.
- Broj različitih trigrama u danom nizu znakova bit će manji od 1000!
- Skup znakova koji se mogu pojaviti je neki podskup od znakova engleske abecede
- Trigrami se mogu preklapati (string "aaab" daje trigrame "aaa" i "aab")
- testni primjer nalazi se i na Degiorgi forumu pod temom "(2013) 1. zadaca trigrami".
- za prebacivanje u velika slova možete koristiti funkciju std::toupper
- nizZnakova bit će dan malim slovima

Opće napomene

- Struktura, funkcije i datoteke koje šaljete 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 šaljete 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 šaljete!

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 trigrami".