

Palindromi

Napišite sučelje i implementaciju za strukture **Palindrom** i **Palindromi**. Struktura **Palindromi** predstavlja skup palindroma, gdje palindrom definiramo kao niz znakova koji se čitaju s lijeva isto kao i zdesna. Struktura **Palindrom** predstavlja sam palindrom. Sučelje treba spremiti u datoteku **palindromi.h**, a implementaciju u datoteku **palindromi.cpp**.

Palindrom – varijable, konstruktori i funkcije članice

- **nužne varijable:** string palindrom, int brojPojavljivanja, int duljina
- **Palindrom()** - inicijalizira varijablu palindrom na prazan string i varijablu brojPojavljivanja na nulu.
- **Palindrom(string p, int bP)** - inicijalizira varijablu palindrom na dani string p, brojPojavljivanja na bP te prilagođava varijablu duljina

Palindromi - varijable, konstruktori i funkcije članice

- **nužne varijable:** int brojPalindroma, polje palindromi
- **Palindromi()** - inicijalizira varijablu brojPalindroma na nulu
- **Palindromi(string nizZnakova)** – radi isto što i funkcija postavi
- **void postavi(string nizZnakova)** – za dani nizZnakova pronalazi sve palindrome u tom nizu. Palindrome sprema u varijablu članicu palindromi kao polje struktura Palindrom.
- **string kaoString(void)** – vraća sve palindrome kao string. Za znak razmaka koristiti ";". Palindromi trebaju biti sortirani silazno po broju pojavljivanja. Ako dva palindroma imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **string najduzi(void)** – vraća sve palindrome najveće pronađene duljine kao string. Za znak razmaka koristiti ";". Palindromi trebaju biti sortirani silazno po broju pojavljivanja. Dakle, ako su najduži pronađeni palindromi "ana" (5 puta) i "ovo" (3 puta) treba vratiti string "ana;ovo". Ako dva palindroma imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **string najkraci(void)** – isto kao i funkcija najduzi samo treba vratiti string najkraćih pronađenih palindroma.
- **Palindromi zajednicki(string nizZnakova)** – vraća strukturu Palindromi koja sadrži sve palindrome zajedničke za nizZnakova i trenutnu strukturu. Palindromi u strukturi trebaju biti sortirani silazno po broju pojavljivanja u stringu nizZnakova. Ako dva palindroma imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski.
- **Palindromi zajednicki(Palindromi P2)** – vraća strukturu Palindromi koja sadrži sve palindrome zajedničke za strukturu P2 i trenutnu strukturu. Palindromi u strukturi trebaju biti sortirani silazno po broju pojavljivanja u strukturi Palindromi P2 (varijablu brojPojavljivanja nove strukture postaviti na odgovarajuće vrijednosti iz varijable P2). Ako dva palindroma imaju istu frekvenciju pojavljivanja treba ih sortirati leksikografski
- **string oznaci(string nizZnakova)** – za dani nizZnakova (dan malim slovima) označiti velikim slovima sve palindrome unutra nizZnakova zajedničke za nizZnakova i trenutnu strukturu. Ostali znakovi moraju ostati zapisani kao mali.

Napomene:

- Treba zanemariti moguće praznine između znakova.
- Minimalna duljina palindroma s kojima smijete raditi jest 3.
- Palindrom je uvijek neparne duljine.
- Podskupove palindroma ne smatramo novim palindromom. Dakle, niz "ananana" sadrži samo jedan palindrom "ananana". Međutim, palindromi ne moraju biti disjunktni ("abab" sadrži dva palindroma "aba" i "bab").
- Broj palindroma (zajedno s podskupovima) u danom nizu znakova bit će manji od 100!
- za prebacivanje u velika slova možete koristiti funkciju `std::toupper`
- testni primjer nalazi se i na Degiorgi forumu pod temom "(2013) 1. zadaca palindromi".

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 palindromi".