Strukture podataka i algoritmi

Projekat 1

Za projekat 1 student može odabrati jednu od četiri ponuđene teme. Temu je potrebno odabrati i prijaviti najkasnije do 10.12.2019 (uključivo). Temu student bira i prijavljuje na linku https://docs.google.com/forms/d/e/1FAIpQLSerLSUVx1qzH5IUr45GPCTaUpTdEvt0ByQv455aUew4TlIHLA/viewform. Završen projekat je potrebno poslati predmetnom asistentu na mail najkasnije 25.12.2019. (do ponoći). Odbrana projekta će se održati 27.12.2019. i trajat će gotovo čitav dan. Dan prije odbrane će biti objavljena satnica u kojem terminu će koji student braniti projekat, a studenti u mailu u kojem šalju projekat mogu naznačiti koji im termin eventualno ne odgovara za odbranu.

Pored traženih stvari u svakoj temi, podrazumijeva se da student (ukoliko je to neophodno za klasu) implementira osnovne stvari da bi klasa bila funkcionalna i upotrebljiva (konstruktor kopije, operator dodjele, destruktor, pomjerajući konstruktor i pomjerajući operator dodjele). Također, potrebno je da student napiše *main* funkciju u kojoj će pokazati kako funkcionišu sve tražene funkcionalnosti. Ovu *main* funkciju posmatrajte kao dio koji pokazujete klijentu koji je zainteresovan za Vaš proizvod, te želite da mu pokažete kako on radi i kako se koristi. Dakle, treba da bude jasno napisano (da se klijent može snaći u tom kodu i samostalno vršiti neka testiranja), te da pokažete kako sve ispravno funkcioniše u različitim situacijama (potencijalno nezgodnim). Naravno, klijent ne vidi samu implementaciju klase.

Tema 1 (10 bodova)

Potrebno je razviti klasu RealanBroj koja podržava operacije sa realnim brojevima proizvoljne veličine. Klasu implementirati kao dvije liste, od kojih jedna predstavlja cifre prije decimalnog zareza, a druga cifre nakon zareza. Pri kreiranju klase ne koristiti tip *list* iz STL. Klasa treba da ima dva konstruktora sa jednim parametrom, od kojih jedan prima objekat tipa double, a drugi objekat tipa string. Konstruktori treba da postave objekat na proslijeđenu vrijednost, pri čemu kod konstruktora koji prima string treba da se izvrši provjera te baci izuzetak ako proslijeđeni string ne predstavlja realan broj. Potrebno je preklopiti unarni operator –, kao i binarne operatore $+$, $-$, $*$, $+=$, $-=$, $*=$. Ukoliko je broj jednak 1,5, u klasi ne treba da se bespotrebno čuvaju dodatne nule (nakon cifre 5). Također, potrebno je preklopiti operatore << i >>, pri čemu operator << ispisuje broj na ekran, dok operator >> podržava unos objekata tipa RealanBroj sa tastature. Pri tome, operator >> treba podržati unos realnog broja kao izraza koji može sadržavati osnovne matematičke operacije, ali i operaciju ^ koja predstavlja stepenovanje. Na primjer, ako je broj objekat tipa RealanBroj, i ukoliko je unos $(3.2 - 2 * (0.8 - 0.2)\text{^}2) * 5$, naredba *cin>>broj* u varijablu *broj* treba da smjesti $12{,}4$ (stepenovanje ima veći prioritet u odnosu na množenje). Ukoliko izraz nije pravilno napisan, potrebno je baciti izuzetak. Ukoliko se za množenje implementira algoritam koji za dva broja koji imaju po n cifara radi u vremenu $\theta(n^2)$ moguće je osvojiti maksimalno 8 bodova.

Tema 2 (9 bodova)

Polinom $x^{999}+1$ nema smisla čuvati kao vektor koeficijenata, jer bi za čuvanje koeficijenata trebao vektor sa 1000 elemenata. Rad sa ovakvim polinomima bi se sveo na sabiranje i množenje nula. Rješenje je u implementaciji polinoma kao povezane liste (ne koristite tip *list* iz STL). Potrebno je podržati operatore << i >>, pri čemu operator << ispisuje polinom na ekran (od članova sa manjim stepenima do onih sa većim, npr. $1-x+2x\text{^}4$, u ispisu ne treba da bude znak „ $*$ “), dok operator >> omogućava unos objekata tipa *Polinom* sa tastature, tako da unos $(3x-(-x-1)\text{^}2)(x^3-2x^2+1)''$ treba podržati i smjestiti polinom $-6x^3+10x^2-10x+4$ u odgovarajuću varijablu. U slučaju nepravilnog unosa treba baciti izuzetak. Znakovi koji se mogu naći u unosu (pored brojeva) su x, $+$, $-$,$*$, ^, (,), $'$ pri čemu stepenovanje ima veći prioritet od množenja. Znak $'$ se može naći samo iza zatvorene zagrade ili iza drugog znaka $'$. Također, izraz tipa $3x$ je validan, ali izraz tipa $x3$ nije (mada $x*3$ jeste, kao i izraz $5*x*3$). Implementirati sve funkcije koje su potrebne da bi se omogućile pomenute funkcionalnosti.

Tema 3 (11 bodova)

Implementirati klasu *Matrica* koja predstavlja matricu proizvoljnog formata. Potrebno je preklopiti operatore << i >>, pri čemu operator << ispisuje matricu na ekran, dok operator >> podržava unos objekata tipa *Matrica* sa tastature. Format za unošenje matrice 2×3 je $[1\ 2\ 3;\ 4\ 5\ 6]$. Međutim, potrebno je podržati i unos složenijih izraza, kao npr. $[1\ 2;3\ 4]\text{^}3*(4*[1\ 2\ 3;4\ 5\ 6]+[7\ 8;\ 9\ 1;\ 2\ 3]\text{^}T)-[1\ 3;7\ 2]\text{^}-1*\ [7\ 8\ 1;1\ 2\ 3]*\ I3$. Ukoliko izraz nije validan iz nekog razloga (npr. ne slažu se dimenzije matrica), potrebno je baciti izuzetak. Također, potrebno je implementirati funkciju koja računa determinantu matrice (ukoliko se radi o kvadratnoj matrici). Implementirati sve funkcije koje su potrebne da bi se omogućile pomenute funkcionalnosti. Ukoliko se ne implementira brzi algoritam za množenje matrica moguće je osvojiti maksimalno 9 bodova.

Tema 4 (9 bodova)

U mnogim praktičnim primjenama pojavljuju se matrice čija su većina elemenata nule. Ovakve matrice nema smisla čuvati kao nizove nizova, jer bi se rad sa njima sveo na sabiranje i množenje nula. Implementirati klasu *Matrica* koja matricu čuva kao listu listi (ali ne koristiti tip *list* iz STL). Na primjer, ako matrica formata 1000×1000 sadrži samo 4 nenulta elementa i to $a_{2,10}=5, a_{2,400}=15, a_{100,50}=25, a_{700,800}=35$, onda klasa *Matrica* treba čuvati samo listu od 3 liste (od kojih prva ima dva elementa 5 i 15, a preostale dvije po jedan, 25 odnosno 35), vodeći evidenciju na koji se red koja lista odnosi, ali i unutar tih listi treba za svaki element voditi evidenciju u kojoj je koloni. Za tu klasu podržati sabiranje, oduzimanje, množenje i stepenovanje matrica, kao i transponovanje matrice. Potrebno je podržati konstruktor sa dva parametra tipa *int* (kreira matricu datih dimenzija popunjenu nulama), te konstruktor koji prima dimenzije matrice i vektor koji se sadrži od nenultih elemenata matrice, a elementi vektora su tipa *pair<pair<int,int>,double>,* pri čemu prvi element para predstavlja red i kolonu elementa, a drugi njegovu vrijednost. Na primjer, pomenuta matrica bi se trebala moći kreirati naredbom

Matrica M(1000,1000,{{{2,10},5},{{100,50},25},{{2,400},15},{{700,800},35}}).