

I PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA A)

Zadatak 1 (5 poena)

Prikažite *tačan izgled ekrana* na kraju izvršavanja ovog C++ programa, uz kratko obrazloženje zbog čega su rezultati onakvi kakvi jesu. Radi nedostatka prostora, izostavljene su “#include” direktive na početku (inače, uključene su biblioteke “iostream”, “string”, “complex”, “algorithm”, “functional”, “set” i “iterator”). Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
void P(int &x, int y, int &z) { x -= y; y -= z; z -= x; }
void P(int &x, int y, int &&z) { x += y; y += z; z += x; }
template <typename T>
    std::complex<double> Q(T &x, T &y) { return {*--y, *x++}; }
int R(std::function<int(int)> x, int y) { return x(x(y)); }
int main() {
    int x(4), y(6), z(1);
    P(z, x + 1, y); std::cout << x << " " << y << " " << z << std::endl;
    x = 4; y = 6; z = 1;
    P(z, x, y + 1); std::cout << x << " " << y << " " << z << std::endl;
    std::set<int> s{2, 7, 3, 6, 9, 0, 0, 4, 6, 2, 5};
    std::set<int>::iterator p(s.begin()), q(s.end());
    while(p != q) std::cout << Q(p, q) << std::endl;
    x = 4; y = 1; z = R([&y](int z){ return y++ * z; }, x);
    std::cout << x << " " << y << " " << z << std::endl;
    std::string k("yxcvbn"), l("poiuztrew"), m(1);
    std::copy(k.begin(), k.end(), l.begin());
    std::copy(k.begin(), k.end(), std::inserter(m, m.begin()));
    std::cout << k << " " << l << " " << m << std::endl;
    return 0;
}
```

Zadatak 2 (2 poena)

Napišite funkciju koja kao parametar prima dva vektora, nazovimo ih “A” i “B”. Elementi vektora “A” su realni brojevi, a elementi vektora “B” su cijeli brojevi. Funkcija treba da kreira novi vektor čiji su elementi oni elementi vektora “A” čije su pozicije (indeksi) određeni elementima vektora “B”. Na primjer, ukoliko vektor “B” sadrži elemente 3, 5, 2, 0 i 5, funkcija treba da kreira i vrati kao rezultat vektor čiji su elementi redom treći, peti, drugi, nulti i peti element vektora “A”. U slučaju da vektor “B” sadrži vrijednosti koje ne mogu biti indeksi niti jednog elementa iz vektora “A”, funkcija treba da baci izuzetak tipa “range_error” uz odgovarajući prateći tekst.

Zadatak 3 (2 poena)

Napišite funkciju koja kao parametre prima neku matricu organiziranu kao vektor vektora realnih brojeva, a koja kao rezultat daje logičku vrijednost “tačno” ili “netačno” u ovisnosti od toga da li je ta matrica gornja trougaona matrica ili ne (gornja trougaona matrica ima sve elemente ispod glavne dijagonale jednake nuli). Ukoliko parametara nema formu matrice (tj. ukoliko svi redovi odgovarajućeg vektora vektora nemaju jednak broj elemenata), funkcija treba baciti izuzetak tipa “domain_error” uz prateći tekst “Parametar nema formu matrice!”. Napišite i mali isječak programa u kojem ćete sa tastature unijeti matricu formata $m \times n$ pri čemu se m i n također unose sa tastature, a koji će nakon toga na ekranu ispisati informaciju da li je unesena matrica gornja trougaona ili ne (tu informaciju ćete dobiti pozivom napisane funkcije).

Zadatak 4 (2,5 poena)

Napravite funkciju koja ima jedan parametar stringovnog tipa, a koja vrši transformaciju prvog znaka u svakoj riječi tog stringa iz malo u veliko slovo (ukoliko znak nije malo slovo, on ostaje netaknut). Na primjer, ukoliko se funkciji prosljedi string "s" čiji je sadržaj " Dobio sam 25 maraka ", nakon završetka funkcije sadržaj stringa "s" treba biti " Dobio Sam 25 Maraka ". Funkcija ne vraća nikakav rezultat.

Zadatak 5 (2 poena)

Napišite funkciju sa 4 parametra f , $xmin$ i $xmax$ i dx . Parametar f je funkcija koja prima realni broj kao parametar i vraća realan broj kao rezultat (koja prima realni broj kao parametar i vraća realan broj kao rezultat), dok su $xmin$, $xmax$ i dx realni brojevi. Funkcija treba da kao rezultat vrati listu koja se sastoji od vrijednosti funkcije f za sve vrijednosti argumenta od $xmin$ do $xmax$ sa korakom dx , odnosno vrijednosti $f(xmin)$, $f(xmin + dx)$, $f(xmin + 2dx)$, ..., $f(xmax)$. Napisanu funkciju demonstrirajte u isječku programa koji će izračunati vrijednosti funkcije $f(x) = 3x^2 - 5$ za sve vrijednosti x od 0 do 10 sa korakom 0.1 i ispisati ih na ekran.

Zadatak 6 (2 poena)

Neka su date dvije sekvence brojeva a_1, a_2, \dots, a_n i b_1, b_2, \dots, b_n iste dužine. Broj $a_1b_1 + a_2b_2 + \dots + a_nb_n$ naziva se *skalarni proizvod* ove dvije sekvence. Napravite generičku funkciju koja kao prva dva parametra prima pokazivače ili iteratore koji omeđuju prvu sekvencu, a kao treći parametar pokazivač ili iterator koji pokazuje na početak druge sekvence (pokazivač ili iterator iza kraja sekvence nije potreban, s obzirom da sekvence imaju istu dužinu). Funkcija kao rezultat treba da vrati skalarni proizvod ove dvije sekvence. Za elemente sekvenci se ništa ne pretpostavlja osim da se elementi prve sekvence mogu množiti sa elementima druge sekvence i da se tako formirani proizvodi mogu sabirati. Također, treba predvidjeti da se ove dvije sekvence mogu nalaziti u kontejnerskim strukturama različite vrste (npr. jedna sekvenca može biti u nizu, a druga u deku). Napisanu funkciju demonstrirajte u isječku programa koji će naći i ispisati skalarni proizvod za dvije liste iste dužine sa fiksnim elementima.

Zadatak 7 (2 poena)

Poznato je funkcija "count_if" daje kao rezultat broj elemenata bloka omeđenog sa pokazivačima ili iteratorima p_1 i p_2 za koje funkcija f koja joj se prenosi kao treći parametar daje rezultat "true". Napišite isječak programa u kojem ćete iskoristiti funkciju "count_if" da u nekom fiksnom nizu od 10 cijelih brojeva prebrojite i ispišete koliko ima brojeva čija je suma cifara veća od n , pri čemu se n unosi sa tastature.

Zadatak 8 (2,5 poena)

Napišite generičku funkciju koja kao parametar prima pet parametra, nazvanih a , m , n , v i mod . Funkcija treba da dinamički alokira matricu sa m redova i n kolona, inicijalizira elemente matrice vrijednostima v i da smjesti u parametar a dvojni pokazivač kojim će se moći pristupiti elementima tako kreirane matrice. Peti parametar mod je logičkog tipa, i ukoliko mu je vrijednost "true", alokaciju treba izvršiti postupkom kontinualne alokacije, a u suprotnom postupkom fragmentirane alokacije. Kao rezultat, funkcija ne vraća ništa. U slučaju da alokacija ne uspije, funkcija treba da baci izuzetak tipa "bad_alloc". Također, funkcija se treba pobrinuti da ni u kom slučaju ne može doći do curenja memorije. Napišite i mali isječak glavnog programa u kojem ćete iskoristiti ovu funkciju da dinamički kreirate matricu formata 3×3 postupkom kontinualne alokacije popunjenu nulama, ispišete elemente tako kreirane matrice na ekran, i na kraju, oslobodite memoriju koju je zauzimala matrica. Obavezno predvidite hvatanje eventualno bačenih izuzetaka.

I PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA B)

Zadatak 1 (5 poena)

Prikažite *tačan izgled ekrana* na kraju izvršavanja ovog C++ programa, uz kratko obrazloženje zbog čega su rezultati onakvi kakvi jesu. Radi nedostatka prostora, izostavljene su “#include” direktive na početku (inače, uključene su biblioteke “iostream”, “string”, “complex”, “algorithm”, “functional”, “set” i “iterator”). Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
void P(int &x, int y, int &z) { x -= y; y -= z; z -= x; }
void P(int &x, int y, int &&z) { x += y; y += z; z += x; }
template <typename T>
    std::complex<double> Q(T &x, T &y) { return {*--y, *x++}; }
int R(std::function<int(int)> x, int y) { return x(x(y)); }
int main() {
    int x(9), y(2), z(5);
    P(z, x + 1, y); std::cout << x << " " << y << " " << z << std::endl;
    x = 9; y = 2; z = 5;
    P(z, x, y + 1); std::cout << x << " " << y << " " << z << std::endl;
    std::set<int> s{1, 6, 6, 9, 4, 0, 3, 7, 4, 1, 8};
    std::set<int>::iterator p(s.begin()), q(s.end());
    while(p != q) std::cout << Q(p, q) << std::endl;
    x = 2; y = 6; z = R([&y](int z){ return y++ * z; }, x);
    std::cout << x << " " << y << " " << z << std::endl;
    std::string k("mnbvcx"), l("lkjhgfdsa"), m(1);
    std::copy(k.begin(), k.end(), l.begin());
    std::copy(k.begin(), k.end(), std::inserter(m, m.begin()));
    std::cout << k << " " << l << " " << m << std::endl;
    return 0;
}
```

Zadatak 2 (2 poena)

Napišite funkciju koja kao parametar prima dva vektora, nazovimo ih “v” i “w”. Elementi vektora “v” su realni brojevi, a elementi vektora “w” su cijeli brojevi. Funkcija treba da kreira novi vektor čiji su elementi oni elementi vektora “v” čije su pozicije (indeksi) određeni elementima vektora “w”. Na primjer, ukoliko vektor “w” sadrži elemente 3, 5, 2, 0 i 5, funkcija treba da kreira i vrati kao rezultat vektor čiji su elementi redom treći, peti, drugi, nulti i peti element vektora “v”. U slučaju da vektor “w” sadrži vrijednosti koje ne mogu biti indeksi niti jednog elementa iz vektora “v”, funkcija treba da baci izuzetak tipa “range_error” uz odgovarajući prateći tekst.

Zadatak 3 (2 poena)

Napišite funkciju koja kao parametre prima neku matricu organiziranu kao vektor vektora realnih brojeva, a koja kao rezultat daje logičku vrijednost “tačno” ili “netačno” u ovisnosti od toga da li je ta matrica donja trougaona matrica ili ne (donja trougaona matrica ima sve elemente iznad glavne dijagonale jednake nuli). Ukoliko parametara nema formu matrice (tj. ukoliko svi redovi odgovarajućeg vektora vektora nemaju jednak broj elemenata), funkcija treba baciti izuzetak tipa “domain_error” uz prateći tekst “Parametar nema formu matrice!”. Napišite i mali isječak programa u kojem ćete sa tastature unijeti matricu formata $m \times n$ pri čemu se m i n također unose sa tastature, a koji će nakon toga na ekranu ispisati informaciju da li je unesena matrica donja trougaona ili ne (tu informaciju ćete dobiti pozivom napisane funkcije).

Zadatak 4 (2,5 poena)

Napravite funkciju koja ima jedan parametar stringovnog tipa, a koja vrši transformaciju prvog znaka u svakoj riječi tog stringa iz malo u veliko slovo (ukoliko znak nije malo slovo, on ostaje netaknut). Na primjer, ukoliko se funkciji prosljedi string "s" čiji je sadržaj " Dobio sam 25 maraka ", nakon završetka funkcije sadržaj stringa "s" treba biti " Dobio Sam 25 Maraka ". Funkcija ne vraća nikakav rezultat.

Zadatak 5 (2 poena)

Napišite funkciju sa 4 parametra f , $xmin$ i $xmax$ i dx . Parametar f je funkcija koja prima realni broj kao parametar i vraća realan broj kao rezultat (koja prima realni broj kao parametar i vraća realan broj kao rezultat), dok su $xmin$, $xmax$ i dx realni brojevi. Funkcija treba da kao rezultat vrati listu koja se sastoji od vrijednosti funkcije f za sve vrijednosti argumenta od $xmin$ do $xmax$ sa korakom dx , odnosno vrijednosti $f(xmin)$, $f(xmin + dx)$, $f(xmin + 2dx)$, ..., $f(xmax)$. Napisanu funkciju demonstrirajte u isječku programa koji će izračunati vrijednosti funkcije $f(x) = 1 / (x + 3)$ za sve vrijednosti x od 0 do 5 sa korakom 0.25 i ispisati ih na ekran.

Zadatak 6 (2 poena)

Neka su date dvije sekvence brojeva a_1, a_2, \dots, a_n i b_1, b_2, \dots, b_n iste dužine. Broj $a_1b_1 + a_2b_2 + \dots + a_nb_n$ naziva se *skalarni proizvod* ove dvije sekvence. Napravite generičku funkciju koja kao prva dva parametra prima pokazivače ili iteratore koji omeđuju prvu sekvencu, a kao treći parametar pokazivač ili iterator koji pokazuje na početak druge sekvence (pokazivač ili iterator iza kraja sekvence nije potreban, s obzirom da sekvence imaju istu dužinu). Funkcija kao rezultat treba da vrati skalarni proizvod ove dvije sekvence. Za elemente sekvenci se ništa ne pretpostavlja osim da se elementi prve sekvence mogu množiti sa elementima druge sekvence i da se tako formirani proizvodi mogu sabirati. Također, treba predvidjeti da se ove dvije sekvence mogu nalaziti u kontejnerskim strukturama različite vrste (npr. jedna sekvenca može biti u nizu, a druga u deku). Napisanu funkciju demonstrirajte u isječku programa koji će naći i ispisati skalarni proizvod za dvije liste iste dužine sa fiksnim elementima.

Zadatak 7 (2 poena)

Poznato je funkcija "count_if" daje kao rezultat broj elemenata bloka omeđenog sa pokazivačima ili iteratorima p_1 i p_2 za koje funkcija f koja joj se prenosi kao treći parametar daje rezultat "true". Napišite isječak programa u kojem ćete iskoristiti funkciju "count_if" da u nekom fiksnom nizu od 10 cijelih brojeva prebrojite i ispišete koliko ima brojeva čija je suma cifara manja od n , pri čemu se n unosi sa tastature.

Zadatak 8 (2,5 poena)

Napišite generičku funkciju koja kao parametar prima pet parametra, nazvanih a , m , n , v i mod . Funkcija treba da dinamički alokira matricu sa m redova i n kolona, inicijalizira elemente matrice vrijednostima v i da smjesti u parametar a dvojni pokazivač kojim će se moći pristupiti elementima tako kreirane matrice. Peti parametar mod je logičkog tipa, i ukoliko mu je vrijednost "true", alokaciju treba izvršiti postupkom fragmentirane alokacije, a u suprotnom postupkom kontinualne alokacije. Kao rezultat, funkcija ne vraća ništa. U slučaju da alokacija ne uspije, funkcija treba da baci izuzetak tipa "bad_alloc". Također, funkcija se treba pobrinuti da ni u kom slučaju ne može doći do curenja memorije. Napišite i mali isječak glavnog programa u kojem ćete iskoristiti ovu funkciju da dinamički kreirate matricu formata 3×3 postupkom kontinualne alokacije popunjenu nulama, ispišete elemente tako kreirane matrice na ekran, i na kraju, oslobodite memoriju koju je zauzimala matrica. Obavezno predvidite hvatanje eventualno bačenih izuzetaka.

II PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA A)

Svi zadaci odnose se na program koji je dat u Prilogu.

Zadatak 1 (0,5 poena)

Šta će na ekran ispisati program iz Priloga kada se pokrene?

Zadatak 2 (1 poen)

U programu iz Priloga postoji curenje memorije. Pronađite ga i riješite ovaj problem.

Zadatak 3 (1,5 poen)

Kopiranje i međusobno dodjeljivanje primjeraka klase “Liga” u programu iz Priloga ne može se obavljati bez neželjenih posljedica. Izvršite neophodne izmjene da se eliminiraju sve te nepoželjne posljedice. Pri tome se pobrinite da optimizirate rad ukoliko se pojavi potreba za kopiranjem privremenih objekata tipa te klase (recimo, rezultata neke funkcije koja vraća takve objekte).

Zadatak 4 (2,5 poena)

Podaci o timovima u ovom programu čuvaju se unutar primjeraka strukture “Tim”, što krši principe objektno zasnovanog programiranja (narušava se skrivanje informacija i enkapsulacija). Izmijenite strukturu “Tim” u klasu čiji su svi atributi privatni. U klasu ćete uvesti konstruktor čiji je jedini parametar ime tima, koji inicijalizira ime tima na postavljenu vrijednost, a sve ostale atribute na inicijalizira na nulu. Pored toga ćete uvesti pristupne metode kojima se može saznati ime tima, broj poena, te gol razlika (razlika između broja datih i broja primljenih golova), kao i metode za registraciju utakmice i ispis podataka o timu. Metoda za registraciju utakmice prihvata broj datih i primljenih golova kao parametre, i na osnovu toga ažurira atribute (pobjeda vrijedi 3 boda, a neriješen rezultat 1 bod). Konačno, metoda za ispis treba redom da ispiše (u jednom redu ekrana) ime tima, broj utakmica, broj pobjeda, broj neriješenih utakmica, broj poraza, broj datih golova, broj primljenih golova i broj poena. Za ime tima treba zauzeti prostor od 20 znakova, u kojem ime treba da bude ispisano poravnato ulijevo, a za sve brojne podatke prostor od 4 znaka na ekranu, pri čemu ispis svakog brojčanog podatka treba biti poravnat udesno unutar predviđenog prostora.

Zadatak 5 (1 poen)

Nakon prepravki obavljenih u Zadatku 4, program više neće raditi (tačnije, neće se kompajlirati). Izvršite sve neophodne prepravke u ostatku programa da on proradi.

Zadatak 6 (3 poena)

Klasa “Liga” rukuje sa dinamički alociranim nizom čiji su elementi objekti tipa “Tim”. Klasa se može učiniti mnogo fleksibilnijom i efikasnijom ukoliko se i sami objekti tipa “Tim” dinamički alociraju, pri čemu će se njihove adrese čuvati u dinamički alociranom nizu pokazivača na objekte tipa “Tim”. Obavite takve izmjene u klasi “Liga”, zadržavajući njen interfejs neizmijenjenim (ostatak programa treba da korektno radi bez ikakvih izmjena sa tako izmijenjenom klasom). Pretpostavite da klasa “Liga” uzima u vlasništvo sve timove koji su u njoj “pohranjeni”.

Zadatak 7 (1,5 poena)

Dodajte u program mogućnost da se nekom timu pohranjenom u ligi može pristupati navođenjem njegovog imena unutar uglastih zagrada. Na primjer, ukoliko je “liga” neka liga (tj. objekat tipa “Liga”), tada “liga[“Jedinstvo”]” treba da predstavlja tim pohranjen u toj ligi čiji je naziv “Jedinstvo”, ukoliko takav postoji u ligi. Ukoliko traženi tim ne postoji u ligi, treba baciti prikladan izuzetak (u ovom i svim narednim zadacima, pretpostavite da je riješen Zadatak 6.).

Zadatak 8 (1,5 poen)

Dodajte u program mogućnost da se u ligu mogu dodavati novi timovi pomoću operatora "+=". Na primjer, ukoliko je "liga" neka liga, konstrukcijom poput "liga += "Velež"" kreira se novi tim imena "Velež" koji se upisuje u ligu.

Zadatak 9 (1 poen)

Dodajte u program mogućnost da se dva tima mogu porediti pomoću operatora "<" i ">", pri čemu je bolji (veći) onaj tim koji ima više poena, odnosno onaj sa većom gol razlikom u slučaju da oba tima imaju isti broj poena.

Zadatak 10 (1,5 poen)

Dodajte u program mogućnost da se liga može zadavati navođenjem imena timova u vitičastim zagradama, odnosno konstrukcijom poput

```
Liga liga{"Željezničar", "Sarajevo", "Velež", "Čelik", "Jedinstvo",  
"Zrinjski", "Sloboda", "Borac"};
```

Zadatak 11 (2,5 poena)

Dodajte u klasu "Liga" metode za registraciju utakmice i za ispis tabele. Metoda za registraciju utakmice treba da kao parametre prima imena timova koji su odigrali utakmicu kao i broj golova koji su dali jedan i drugi tim (uz bacanje izuzetka ukoliko navedeni timovi ne postoje). Metoda za ispis tabele treba da ispiše tabelu lige sortiranu u opadajućem poretku po broju bodova (ili gol razlici u slučaju kada timovi imaju isti broj poena).

Zadatak 12 (2,5 poena)

Dodajte u klasu "Liga" metode koje vrše pohranjivanje sadržaja lige i obnovu njenog sadržaja iz binarne datoteke čije se ime navodi kao parametar.

Prilog:

```
#include <iostream>
#include <cstring>
#include <stdexcept>

struct Tim {
    char ime_tima[20];
    int br_odigranih, br_pobjeda, br_nerijesenih,
        br_poraza, br_datih, br_primljenih, br_poena;
};

class Liga {
    int broj_timova, kapacitet;
    Tim *timovi;
public:
    explicit Liga(int kapacitet) : kapacitet(kapacitet), broj_timova(0),
        timovi(new Tim[kapacitet]) {}
    ~Liga() { delete[] timovi; }
    void RegistrirajTim(Tim &t) {
        if(broj_timova >= kapacitet) throw std::range_error("Popunjen kapacitet!");
        timovi[broj_timova++] = t;
    }
    Tim &DajTim(int redni_broj) { return timovi[redni_broj - 1]; }
    Tim DajTim(int redni_broj) const { return timovi[redni_broj - 1]; }
};

int main() {
    Tim *t1, *t2;
    t1 = new Tim; t2 = new Tim;
    std::strcpy(t1->ime_tima, "Sarajevo");
    std::strcpy(t2->ime_tima, "Zeljeznicar");
    Liga liga(10);
    liga.RegistrirajTim(*t2); liga.RegistrirajTim(*t1);
    std::cout << liga.DajTim(1).ime_tima << std::endl;
    std::cout << liga.DajTim(2).ime_tima << std::endl;
    return 0;
}
```

II PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA B)

Svi zadaci odnose se na program koji je dat u Prilogu.

Zadatak 1 (0,5 poena)

Šta će na ekran ispisati program iz Priloga kada se pokrene?

Zadatak 2 (1 poen)

U programu iz Priloga postoji curenje memorije. Pronađite ga i riješite ovaj problem.

Zadatak 3 (1,5 poen)

Kopiranje i međusobno dodjeljivanje primjeraka klase “Liga” u programu iz Priloga ne može se obavljati bez neželjenih posljedica. Izvršite neophodne izmjene da se eliminiraju sve te nepoželjne posljedice. Pri tome se pobrinite da optimizirate rad ukoliko se pojavi potreba za kopiranjem privremenih objekata tipa te klase (recimo, rezultata neke funkcije koja vraća takve objekte).

Zadatak 4 (2,5 poena)

Podaci o timovima u ovom programu čuvaju se unutar primjeraka strukture “Tim”, što krši principe objektno zasnovanog programiranja (narušava se skrivanje informacija i enkapsulacija). Izmijenite strukturu “Tim” u klasu čiji su svi atributi privatni. U klasu ćete uvesti konstruktor čiji je jedini parametar ime tima, koji inicijalizira ime tima na postavljenu vrijednost, a sve ostale atribute na inicijalizira na nulu. Pored toga ćete uvesti pristupne metode kojima se može saznati ime tima, broj poena, te gol razlika (razlika između broja datih i broja primljenih golova), kao i metode za registraciju utakmice i ispis podataka o timu. Metoda za registraciju utakmice prihvata broj datih i primljenih golova kao parametre, i na osnovu toga ažurira atribute (pobjeda vrijedi 3 boda, a neriješen rezultat 1 bod). Konačno, metoda za ispis treba redom da ispiše (u jednom redu ekrana) ime tima, broj utakmica, broj pobjeda, broj neriješenih utakmica, broj poraza, broj datih golova, broj primljenih golova i broj poena. Za ime tima treba zauzeti prostor od 20 znakova, u kojem ime treba da bude ispisano poravnato ulijevo, a za sve brojne podatke prostor od 4 znaka na ekranu, pri čemu ispis svakog brojčanog podatka treba biti poravnat udesno unutar predviđenog prostora.

Zadatak 5 (1 poen)

Nakon prepravki obavljenih u Zadatku 4, program više neće raditi (tačnije, neće se kompajlirati). Izvršite sve neophodne prepravke u ostatku programa da on proradi.

Zadatak 6 (3 poena)

Klasa “Liga” rukuje sa dinamički alociranim nizom čiji su elementi objekti tipa “Tim”. Klasa se može učiniti mnogo fleksibilnijom i efikasnijom ukoliko se i sami objekti tipa “Tim” dinamički alociraju, pri čemu će se njihove adrese čuvati u dinamički alociranom nizu pokazivača na objekte tipa “Tim”. Obavite takve izmjene u klasi “Liga”, zadržavajući njen interfejs neizmijenjenim (ostatak programa treba da korektno radi bez ikakvih izmjena sa tako izmijenjenom klasom). Pretpostavite da klasa “Liga” uzima u vlasništvo sve timove koji su u njoj “pohranjeni”.

Zadatak 7 (1,5 poena)

Dodajte u program mogućnost da se nekom timu pohranjenom u ligi može pristupati navođenjem njegovog imena unutar uglastih zagrada. Na primjer, ukoliko je “liga” neka liga (tj. objekat tipa “Liga”), tada “liga[“Jedinstvo”]” treba da predstavlja tim pohranjen u toj ligi čiji je naziv “Jedinstvo”, ukoliko takav postoji u ligi. Ukoliko traženi tim ne postoji u ligi, treba baciti prikladan izuzetak (u ovom i svim narednim zadacima, pretpostavite da je riješen Zadatak 6.).

Zadatak 8 (1,5 poen)

Dodajte u program mogućnost da se u ligu mogu dodavati novi timovi pomoću operatora "+=". Na primjer, ukoliko je "liga" neka liga, konstrukcijom poput "liga += "Velež"" kreira se novi tim imena "Velež" koji se upisuje u ligu.

Zadatak 9 (1 poen)

Dodajte u program mogućnost da se dva tima mogu porediti pomoću operatora "<" i ">", pri čemu je bolji (veći) onaj tim koji ima više poena, odnosno onaj sa većom gol razlikom u slučaju da oba tima imaju isti broj poena.

Zadatak 10 (1,5 poen)

Dodajte u program mogućnost da se liga može zadavati navođenjem imena timova u vitičastim zagradama, odnosno konstrukcijom poput

```
Liga liga{"Željezničar", "Sarajevo", "Velež", "Čelik", "Jedinstvo",  
"Zrinjski", "Sloboda", "Borac"};
```

Zadatak 11 (2,5 poena)

Dodajte u klasu "Liga" metode za registraciju utakmice i za ispis tabele. Metoda za registraciju utakmice treba da kao parametre prima imena timova koji su odigrali utakmicu kao i broj golova koji su dali jedan i drugi tim (uz bacanje izuzetka ukoliko navedeni timovi ne postoje). Metoda za ispis tabele treba da ispiše tabelu lige sortiranu u opadajućem poretku po broju bodova (ili gol razlici u slučaju kada timovi imaju isti broj poena).

Zadatak 12 (2,5 poena)

Dodajte u klasu "Liga" metode koje vrše pohranjivanje sadržaja lige i obnovu njenog sadržaja iz binarne datoteke čije se ime navodi kao parametar.

Prilog:

```
#include <iostream>
#include <cstring>
#include <stdexcept>

struct Tim {
    char ime_tima[20];
    int br_odigranih, br_pobjeda, br_nerijesenih,
        br_poraza, br_datih, br_primljenih, br_poena;
};

class Liga {
    int broj_timova, kapacitet;
    Tim *timovi;
public:
    explicit Liga(int kapacitet) : kapacitet(kapacitet), broj_timova(0),
        timovi(new Tim[kapacitet]) {}
    ~Liga() { delete[] timovi; }
    void RegistrirajTim(Tim &t) {
        if(broj_timova >= kapacitet) throw std::range_error("Popunjen kapacitet!");
        timovi[broj_timova++] = t;
    }
    Tim &DajTim(int redni_broj) { return timovi[redni_broj - 1]; }
    Tim DajTim(int redni_broj) const { return timovi[redni_broj - 1]; }
};

int main() {
    Tim *t1, *t2;
    t1 = new Tim; t2 = new Tim;
    std::strcpy(t1->ime_tima, "Sarajevo");
    std::strcpy(t2->ime_tima, "Zeljeznicar");
    Liga liga(10);
    liga.RegistrirajTim(*t2); liga.RegistrirajTim(*t1);
    std::cout << liga.DajTim(1).ime_tima << std::endl;
    std::cout << liga.DajTim(2).ime_tima << std::endl;
    return 0;
}
```