

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”, “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>
    void Q(T &x, T &y) { std::cout << *x++ << "," << *--y << " "; }
int R(std::function<int(int)> x, int y) { return x(x(y)); }
int main() {
    int x(3), y(5), z(2);
    P(z, x + 1, y); std::cout << x << " " << y << " " << z << std::endl;
    x = 3; y = 5; z = 2;
    P(z, x, y + 1); std::cout << x << " " << y << " " << z << std::endl;
    std::set<int> s{3, 5, 8, 2, 5, 6, 1, 5, 0, 9, 3};
    std::set<int>::iterator p(s.begin()), q(s.end());
    while(p != q) Q(p, q); std::cout << std::endl;
    x = 2; y = 3; z = R([&y](int z){ return y++ * z; }, x);
    std::cout << x << " " << y << " " << z << std::endl;
    std::string k("qwerty"), l("asdfghjkl"), 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 vektor cijelih brojeva (koji mogu biti i pozitivni i negativni), a koja kao rezultat vraća novi vektor čiji je i -ti element jednak najmanjoj cifri unutar i -tog elementa vektora koji je proslijeđen kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 352, 47132, -475, 98 i -647, funkcija treba da vrati kao rezultat vektor čiji su elementi 2, 1, 4, 8 i 4. Pri tome, nije dozvoljeno pisati nikakve pomoćne funkcije. Također, napišite i programski isječak u kojem ćete unijeti 10 brojeva sa tastature u vektor, a nakon toga pozvati napisanu funkciju i ispisati kolike su najmanje cifre unutar unesenih brojeva.

Zadatak 3 (2,5 poena)

Napišite funkciju koja kao parametre prima dvije matrice organizirane kao vektore vektora realnih brojeva, a koja kao rezultat daje novu matricu koja je jednaka proizvodu ove dvije matrice, ukoliko se one mogu množiti. Matrice se mogu množiti jedino ukoliko je broj kolona prve matrice jednak broju redova druge matrice. U tom slučaju je element na poziciji (i, j) u rezultirajućoj matrici jednak zbiru proizvoda svih elemenata i -tog reda prve matrice sa odgovarajućim elementima j -te kolone druge matrice. Ukoliko neki od 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 matrica!”. Također, ukoliko se matrice ne mogu množiti, treba baciti izuzetak istog tipa, uz prateći tekst “Matrice nisu saglasne za množenje!”. Napišite i mali isječak programa u kojem ćete izračunati proizvod za dvije fiksne matrice formata 3×3 (njihove elemente odaberite proizvoljno) i ispisati ga na ekran. Elemente rezultirajuće matrice treba ispisati red po red, pri čemu ćete za svaki element zauzeti prostor od 8 znakova na ekranu.

Zadatak 4 (2 poena)

Jedan od najprostijih sistema šifriranja teksta je tzv. *monoalfabetски sistem*, gdje se prosto svaki znak nekog teksta mijenja nekim drugim znakom u skladu sa nekim ključem. Recimo, tekst "abcbeaccbac" se prema ključu $a \rightarrow x, b \rightarrow c, c \rightarrow a$ pretvara u tekst "xcacaxaacxa". Napišite funkciju za šifriranje monoalfabetским sistemom. Prvi parametar je string koji se šifrira, dok je drugi parametar mapa koja predstavlja ključ. Mapa se sastoji od parova znakova, pri čemu prvi element svakog para predstavlja znak koji se transformira, a drugi element predstavlja znak u koji se vrši transformacija. Funkcija vraća kao rezultat šifrirani string (dok string koji joj je prenesen kao parametar ostaje neizmijenjen). Svi znakovi stringa za koje nije u ključu propisano kako se transformiraju ostaju neizmijenjeni. Napisanu funkciju demonstrirajte u isječku programa koji će ispisati na ekranu šifrirani tekst za neki tekst unesen sa tastature pomoću nekog ključa koji možete odabrati po volji.

Zadatak 5 (2 poena)

Napišite generičku funkciju sa 3 parametra x, f i n . x je vrijednost nekog nedefiniranog tipa T , f je funkcija koja prima parametar tipa T i vraća rezultat tipa T (ili nešto što se ponaša poput funkcije koja prima parametar tipa T i vraća rezultat tipa T), dok je n cijeli broj. Funkcija treba da kao rezultat vrati vrijednost $f(f(f(\dots(f(x))\dots))$ gdje se funkcija f uzastopno primjenjuje n puta, odnosno vrijednost koja se dobije kada se na argument x funkcija f primijeni n puta. Napisanu funkciju demonstrirajte u isječku programa koji će izračunati vrijednost $f(f(f(f(f(10))))))$ gdje je $f(x) = 3x^2 - 5$.

Zadatak 6 (2 poena)

Neka je data neka sekvenca brojeva $a_1, a_2, a_3, \dots, a_n$. Pod *kumulativom sumom* te sekvence naziva se nova sekvenca brojeva $a_1, a_1 + a_2, a_1 + a_2 + a_3, \dots, a_1 + a_2 + a_3 + \dots + a_n$. Napravite generičku funkciju koja kao prva dva parametra prima pokazivače ili iteratore koji omeđuju neki slijed brojeva (u stilu funkcija iz biblioteke "algorithm", a koja na odredište na koje pokazuje trećim parametrom (koji opet može biti pokazivač ili iterator) smješta novu sekvencu koja predstavlja kumulativnu sumu prve sekvence. Za elemente slijeda se ništa ne pretpostavlja osim da se mogu sabirati. Napisanu funkciju demonstrirajte u isječku programa koji će za neku listu sa fiksnim elementima izračunati njenu kumulativnu sumu i smjestiti je u neki vektor koji potom treba ispisati na ekran.

Zadatak 7 (2 poena)

Pretpostavimo da želimo sortirati niz kompleksnih brojeva tako da kompleksni broj z_1 dolazi ispred kompleksnog broja z_2 ako i samo ako je apsolutna vrijednost od z_1 manja od apsolutne vrijednosti od z_2 , ili ako su apsolutne vrijednosti od z_1 i z_2 jednake, ali je realni dio od z_1 manji od realnog dijela od z_2 . Definirajte odgovarajuću funkciju kriterija, i napišite isječak programa u kojem ćete pokazati kako biste deklarirali dek kompleksnih brojeva, napunili ga vrijednostima unesenim sa tastature i sortirali ga u traženi poredak korištenjem funkcije "sort" iz biblioteke "algorithm".

Zadatak 8 (2,5 poena)

Napišite generičku funkciju koja kao parametar prima vektor stringova, a koja kreira dinamički alociranu strukturu podataka koja sadrži iste informacije kao i zadani vektor stringova. Konkretnije, ta struktura podataka treba da se sastoji od dinamički alociranog niza pokazivača, kojih ima onoliko koliko ima stringova u vektoru, pri čemu će svaki od pokazivača da pokazuje na dinamički alocirani niz znakova koji će sadržavati iste znakove kao i odgovarajući string. Pri tome, taj niz znakova treba biti nul-terminiran (tj. iza njegovog posljednjeg znaka treba se nalaziti granični znak sa ASCII šifrom 0). Funkcija kao rezultat treba da vrati dvojni pokazivač na alocirani niz pokazivača, preko kojeg se može pristupiti kreiranoj strukturi podataka. U slučaju da alokacija ne uspije, funkcija treba da baci izuzetak tipa "bad_alloc" kao izuzetak. Također, funkcija se treba pobrinuti da ni u koj slučaju ne može doći do curenja memorije. Napišite i mali isječak glavnog programa (ne treba sve) u kojem ćete pozvati ovu funkciju nad jednim fiksnim vektorom stringova (elemente odaberite po volji) i nakon toga ispisati na ekranu sve informacije pohranjene u kreiranoj strukturi podataka (tj. ispis treba da bude isti kao da ste ispisali odgovarajuće stringove pohranjene u polaznom vektoru stringova).

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”, “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>
    void Q(T &x, T &y) { std::cout << *x++ << " " << *--y << " "; }
int R(std::function<int(int)> x, int y) { return x(x(y)); }
int main() {
    int x(6), y(1), z(4);
    P(z, x + 1, y); std::cout << x << " " << y << " " << z << std::endl;
    x = 6; y = 1; z = 4;
    P(z, x, y + 1); std::cout << x << " " << y << " " << z << std::endl;
    std::set<int> s{4, 6, 9, 3, 6, 7, 2, 6, 1, 0, 4};
    std::set<int>::iterator p(s.begin()), q(s.end());
    while(p != q) Q(p, q); std::cout << std::endl;
    x = 3; y = 4; z = R([&y](int z){ return --y * z; }, x);
    std::cout << x << " " << y << " " << z << std::endl;
    std::string k("asdfgh"), l("qwertyuiop"), 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 vektor cijelih brojeva (koji mogu biti i pozitivni i negativni), a koja kao rezultat vraća novi vektor čiji je i -ti element jednak najvećoj cifri unutar i -tog elementa vektora koji je proslijeđen kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 352, 47132, -475, 98 i -647, funkcija treba da vrati kao rezultat vektor čiji su elementi 5, 7, 7, 9 i 7. Pri tome, nije dozvoljeno pisati nikakve pomoćne funkcije. Također, napišite i programski isječak u kojem ćete unijeti 10 brojeva sa tastature u vektor, a nakon toga pozvati napisanu funkciju i ispisati kolike su najveće cifre unutar unesenih brojeva.

Zadatak 3 (2,5 poena)

Napišite funkciju koja kao parametre prima dvije matrice organizirane kao vektore vektora realnih brojeva, a koja kao rezultat daje novu matricu koja je jednaka proizvodu ove dvije matrice, ukoliko se one mogu množiti. Matrice se mogu množiti jedino ukoliko je broj kolona prve matrice jednak broju redova druge matrice. U tom slučaju je element na poziciji (i, j) u rezultirajućoj matrici jednak zbiru proizvoda svih elemenata i -tog reda prve matrice sa odgovarajućim elementima j -te kolone druge matrice. Ukoliko neki od 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 matrica!”. Također, ukoliko se matrice ne mogu množiti, treba baciti izuzetak istog tipa, uz prateći tekst “Matrice nisu saglasne za množenje!”. Napišite i mali isječak programa u kojem ćete izračunati proizvod za dvije fiksne matrice formata 3×3 (njihove elemente odaberite proizvoljno) i ispisati ga na ekran. Elemente rezultirajuće matrice treba ispisati red po red, pri čemu ćete za svaki element zauzeti prostor od 8 znakova na ekranu.

Zadatak 4 (2 poena)

Jedan od najprostijih sistema šifriranja teksta je tzv. *monoalfabetски sistem*, gdje se prosto svaki znak nekog teksta mijenja nekim drugim znakom u skladu sa nekim ključem. Recimo, tekst "abcbeaccbac" se prema ključu $a \rightarrow x, b \rightarrow c, c \rightarrow a$ pretvara u tekst "xcacaxaacxa". Napišite funkciju za šifriranje monoalfabetским sistemom. Prvi parametar je string koji se šifrira, dok je drugi parametar mapa koja predstavlja ključ. Mapa se sastoji od parova znakova, pri čemu prvi element svakog para predstavlja znak koji se transformira, a drugi element predstavlja znak u koji se vrši transformacija. Funkcija vraća kao rezultat šifrirani string (dok string koji joj je prenesen kao parametar ostaje neizmijenjen). Svi znakovi stringa za koje nije u ključu propisano kako se transformiraju ostaju neizmijenjeni. Napisanu funkciju demonstrirajte u isječku programa koji će ispisati na ekranu šifrirani tekst za neki tekst unesen sa tastature pomoću nekog ključa koji možete odabrati po volji.

Zadatak 5 (2 poena)

Napišite generičku funkciju sa 3 parametra x, f i n . x je vrijednost nekog nedefiniranog tipa T , f je funkcija koja prima parametar tipa T i vraća rezultat tipa T (ili nešto što se ponaša poput funkcije koja prima parametar tipa T i vraća rezultat tipa T), dok je n cijeli broj. Funkcija treba da kao rezultat vrati vrijednost $f(f(f(\dots(f(x))\dots))$ gdje se funkcija f uzastopno primjenjuje n puta, odnosno vrijednost koja se dobije kada se na argument x funkcija f primijeni n puta. Napisanu funkciju demonstrirajte u isječku programa koji će izračunati vrijednost $f(f(f(f(6))))$ gdje je $f(x) = 2x^2 + 7$.

Zadatak 6 (2 poena)

Neka je data neka sekvenca brojeva $a_1, a_2, a_3, \dots, a_n$. Pod *kumulativom sumom* te sekvence naziva se nova sekvenca brojeva $a_1, a_1 + a_2, a_1 + a_2 + a_3, \dots, a_1 + a_2 + a_3 + \dots + a_n$. Napravite generičku funkciju koja kao prva dva parametra prima pokazivače ili iteratore koji omeđuju neki slijed brojeva (u stilu funkcija iz biblioteke "algorithm", a koja na određite na koje pokazuje trećim parametrom (koji opet može biti pokazivač ili iterator) smješta novu sekvencu koja predstavlja kumulativnu sumu prve sekvence. Za elemente slijeda se ništa ne pretpostavlja osim da se mogu sabirati. Napisanu funkciju demonstrirajte u isječku programa koji će za neku listu sa fiksnim elementima izračunati njenu kumulativnu sumu i smjestiti je u neki dek koji potom treba ispisati na ekran.

Zadatak 7 (2 poena)

Pretpostavimo da želimo sortirati slijed kompleksnih brojeva tako da kompleksni broj z_1 dolazi ispred kompleksnog broja z_2 ako i samo ako je apsolutna vrijednost od z_1 veća od apsolutne vrijednosti od z_2 , ili ako su apsolutne vrijednosti od z_1 i z_2 jednake, ali je imaginarni dio od z_1 manji od imaginarnog dijela od z_2 . Definirajte odgovarajuću funkciju kriterija, i napišite isječak programa u kojem ćete pokazati kako biste deklarirali vektor kompleksnih brojeva, napunili ga vrijednostima unesenim sa tastature i sortirali ga u traženi poredak korištenjem funkcije "sort" iz biblioteke "algorithm".

Zadatak 8 (2,5 poena)

Napišite generičku funkciju koja kao parametar prima vektor stringova, a koja kreira dinamički alociranu strukturu podataka koja sadrži iste informacije kao i zadani vektor stringova. Konkretnije, ta struktura podataka treba da se sastoji od dinamički alociranog niza pokazivača, kojih ima onoliko koliko ima stringova u vektoru, pri čemu će svaki od pokazivača da pokazuje na dinamički alocirani niz znakova koji će sadržavati iste znakove kao i odgovarajući string. Pri tome, taj niz znakova treba biti nul-terminiran (tj. iza njegovog posljednjeg znaka treba se nalaziti granični znak sa ASCII šifrom 0). Funkcija kao rezultat treba da vrati dvojni pokazivač na alocirani niz pokazivača, preko kojeg se može pristupiti kreiranoj strukturi podataka. U slučaju da alokacija ne uspije, funkcija treba da baci izuzetak tipa "bad_alloc" kao izuzetak. Također, funkcija se treba pobrinuti da ni u koj slučaju ne može doći do curenja memorije. Napišite i mali isječak glavnog programa (ne treba sve) u kojem ćete pozvati ovu funkciju nad jednim fiksnim vektorom stringova (elemente odaberite po volji) i nakon toga ispisati na ekranu sve informacije pohranjene u kreiranoj strukturi podataka (tj. ispis treba da bude isti kao da ste ispisali odgovarajuće stringove pohranjene u polaznom vektoru stringova).

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

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

Zadatak 1 (1 poen)

Autor klase “GrafickiSistem” u programu datom u Prilogu htio je izbjeći probleme sa upravljanjem memorijom tako što je koristio vektor pametnih pokazivača za čuvanje pokazivača na grafičke elemente pohranjene u sistemu. Bez obzira na to, ovaj program i dalje ima curenje memorije. Napišite šta sve treba uraditi da se riješi ovaj problem. Pretpostavite pri tome da problemi ne nastaju usljed nedostatka memorije (tj. pretpostavimo da ne dolazi do bacanja izuzetka usljed nestašice memorije).

Zadatak 2 (2,5 poena)

Uz pretpostavku da je riješen problem opisan u Zadatku 1, u programu će postojati jedna klasa koja je takva da kopiranje i međusobno dodjeljivanje njenih primjeraka može dovesti do neželjenih posljedica. Ustanovite koja je to klasa i napišite sav potreban kod da se eliminišu 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 3 (1,5 poen)

Glavni program u programu iz priloga, nakon što kreira neke grafičke elemente i registrira ih u grafički sistem, trebao bi na ekranu ispisati osnovne informacije o registriranim objektima. Međutim, program ima bag zbog kojeg ispis nije u skladu sa očekivanjima. Šta će ovaj program uopće ispisati? Šta treba popraviti da ispis bude u skladu sa očekivanjima? Kako će tada ispis izgledati (navedite tačan izgled ekrana)?

Zadatak 4 (1 poen)

Konstruktor klase “Poligon” je donekle nesiguran, jer ukoliko alokacija niza “x” uspije a alokacija niza “y” ne uspije, imaćemo curenje memorije. Obavite izmjene kojima se rješava opisani problem.

Zadatak 5 (2 poena)

Funkcija “DodajTjeme” u klasi “Poligon” prima kao parametar redni broj tjemena, te x i y koordinate tjemena, ali ne provjerava da li je redni broj tjemena u dozvoljenom opsegu. Prepravite ovu funkciju da baca izuzetak tipa “range_error” uz prikladan prateći tekst ukoliko redni broj tjemena nije u legalnom opsegu. Također, dodajte još jednu verziju metode “DodajTjeme” koja umjesto x i y koordinata tjemena prima objekat tipa “Tacka” kao odgovarajući parametar preko kojeg se zadaje tjeme (boja tačke se ignorira), te dvije pristupne metode u klasu “Tacka” kojima se omogućava saznavanje informacija o x odnosno y koordinati tačke.

Zadatak 6 (2 poena)

Grafički sistem iz ovog programa ograničen je samo na dvije vrste objekata: tačke i poligone. Dodajte mogućnost da se u ovaj grafički sistem mogu dodavati i krugovi. Krugovi trebaju biti opisani sa bojom, centrom i poluprečnikom, pri čemu se centar može zadavati na dva načina: kao dva realna broja (x i y koordinata) ili kao objekat tipa “Tacka”. Treba biti podržano da se mogu saznavati informacije o centru kruga i poluprečniku, kao i ispis informacija o krugu u obliku “ $\{\{x, y\}, r\}$ ”.

Zadatak 7 (1 poen)

Dodajte mogućnost da postanu legalni izrazi poput “ $k * t$ ” i “ $t * k$ ” gdje je “ k ” neki realan broj, a “ t ” objekat tipa “Tacka”. Kao rezultat ovih izraza, treba da se dobije novi objekat tipa “Tacka” u kojem su obje koordinate pomnožene sa realnim brojem “ k ”.

Zadatak 8 (1,5 poen)

Dodajte mogućnost da se za podržane grafičke elemente može saznati informacija o njihovoj površini. Za površinu tačke uzeti da je jednaka 0, površina poligona računa se po formuli

$$P = \sum_{i=3}^n x_1 (y_{i-1} - y_i) + x_{i-1} (y_i - y_1) + x_i (y_1 - y_{i-1})$$

gdje je n broj tjemena, a (x_i, y_i) su odgovarajuće koordinate, dok se površina kruga računa po formuli koju bi trebali znati svi koji su uopće došli raditi ovaj ispit. Mora biti podržano da se površina ma kojeg elementa može dobiti bez da se unaprijed zna o kojem se tačno tipu elementa radi.

Zadatak 9 (1,5 poen)

Dodajte u klasu "Poligon" mogućnost da se informacije o tjemenu mogu saznati ili zadavati navođenjem rednog broja tjemena u uglastim zagradama, tako da ako je "p" neki objekat tipa "Poligon", izraz "p[3]" treba da predstavlja njegovo treće tjeme i to u vidu objekta tipa "Tačka". Treba omogućiti da se na ovaj način mogu ne samo iščitavati nego i mijenjati informacije o tjemenu, pri čemu izmjena treba biti podržana samo za slučaj nekonstantnih objekata tipa "Poligon".

Zadatak 10 (2 poena)

Mada se, zahvaljujući pametnim pokazivačima, objekti tipa "GrafickiSistem" mogu bezbjedno kopirati i međusobno dodjeljivati, takvo kopiranje je ipak zasnovano na plitkim kopijama (bezbjednost je ostvarena preko brojanja pristupa, koje rade pametni pokazivači). Napravite neophodne izmjene koje će omogućiti da se kopiranje objekata tipa "GrafickiSistem" vrši putem dubokog kopiranja. Obratite pažnju na neophodnost polimorfnog kopiranja.

Zadatak 11 (2 poena)

Funkcija "RegistrirajElement" kao parametar prima pokazivač na neki grafički element, te ga upisuje u kolekciju i preuzima vlasništvo nad njim (pretpostavljajući da je on dinamički alociran). Dodajte još jednu verziju funkcije "RegistrirajElement" koja umjesto pokazivača prima *referencu* na neki grafički element. Za razliku od prethodne funkcije, ova funkcija neće praviti nikakve pretpostavke kako je objekat kreiran i neće ga uzimati u vlasništvo. Umjesto toga, funkcija treba da dinamički kreira kopiju objekta koji se registrira i da preuzme vlasništvo nad tako kreiranom kopijom. Napomena: vjerovatno ćete se moći poslužiti nekim stvarima iz prethodnog zadatka.

Zadatak 12 (2 poena)

Napišite funkciju u klasi "GrafickiSistem" koja kao parametar ima ime datoteke, a omogućava da se podaci o grafičkim elementima pročitaju iz tekstualne datoteke. Datoteka je organizirana na sljedeći način. U prvom redu datoteke nalazi se broj grafičkih elemenata. Nakon toga, sljedeći podaci se redom ponavljaju za svaki element. Podaci za jedan element su u jednom redu. Na početku reda se nalazi znak "T", "P" ili "K", ovisno da li je u pitanju tačka, poligon ili krug. Dalje su svi podaci koji slijede međusobno razdvojeni razmacima. Za slučaj tačke, prvo se nalazi šifra boje (cijeli broj u opsegu od 0 do 6), te x i y koordinate tačke. Za slučaj poligona, imamo šifru boje, zatim broj tjemena, te koordinate tjemena (onoliko parova brojeva koliko ima tjemena). Za slučaj kruga imamo šifru boje, zatim koordinate centra, te na kraju poluprečnik. Ukoliko su već postojali podaci u sistemu, oni prethodno trebaju biti obrisani. Radi jednostavnosti, pretpostavite da datoteka sadrži isključivo ispravne podatke.

Prilog:

```
#include <iostream>
#include <vector>
#include <memory>

class GrafickiElement {
public:
    enum Boje {Crvena, Zelena, Zuta, Plava, Ljubicasta, Crna, Bijela};
    GrafickiElement(Boje boja) : boja(boja) {}
    Boje DajBoju() const { return boja; }
    void Ispisi() const { std::cout << boja; };
private:
    Boje boja;
};

class Tacka : public GrafickiElement {
    double x, y;
public:
    Tacka(GrafickiElement::Boje boja, double x, double y) :
        GrafickiElement(boja), x(x), y(y) {}
    void Ispisi() const { std::cout << "{" << x << "," << y << " "; }
};

class Poligon : public GrafickiElement {
    int broj_tjemena;
    double *x, *y;
public:
    Poligon(GrafickiElement::Boje boja, int broj_tjemena) :
        GrafickiElement(boja), broj_tjemena(broj_tjemena),
        x(new double[broj_tjemena]), y(new double[broj_tjemena]) {}
    void PostaviTjeme(int redni_broj, double x, double y) {
        Poligon::x[redni_broj - 1] = x; Poligon::y[redni_broj - 1] = y;
    }
    void Ispisi() const {
        std::cout << "{";
        for(int i = 0 ; i < broj_tjemena; i++) {
            std::cout << "{" << x[i] << "," << y[i] << " ";
            if(i != broj_tjemena - 1) std::cout << ",";
        }
        std::cout << " ";
    }
};

class GrafickiSistem {
    std::vector<std::shared_ptr<GrafickiElement>> elementi;
public:
    GrafickiSistem() {}
    void RegistrirajElement(GrafickiElement *e) {
        elementi.push_back(std::shared_ptr<GrafickiElement>(e));
    }
    void IspisiSve() {
        for(int i = 0; i < elementi.size(); i++) {
            elementi[i]->Ispisi();
            std::cout << std::endl;
        }
    }
};

int main() {
    GrafickiSistem s;
    s.RegistrirajElement(new Tacka(GrafickiElement::Crvena, 30, 20));
    Poligon *p(new Poligon(GrafickiElement::Plava, 4));
    p->PostaviTjeme(1, 10, 10); p->PostaviTjeme(2, 10, 50);
    p->PostaviTjeme(3, 50, 50); p->PostaviTjeme(4, 50, 10);
    s.RegistrirajElement(p);
    s.IspisiSve();
}
```

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

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

Zadatak 1 (1 poen)

Autor klase “GeometrijskaKolekcija” u programu datom u Prilogu htio je izbjeći probleme sa upravljanjem memorijom tako što je koristio vektor pametnih pokazivača za čuvanje pokazivača na geometrijske objekte pohranjene u sistemu. Bez obzira na to, ovaj program i dalje ima curenje memorije. Napišite šta sve treba uraditi da se riješi ovaj problem. Pretpostavite pri tome da problemi ne nastaju usljed nedostatka memorije (tj. pretpostavimo da ne dolazi do bacanja izuzetka usljed nestašice memorije).

Zadatak 2 (2,5 poena)

Uz pretpostavku da je riješen problem opisan u Zadatku 1, u programu će postojati jedna klasa koja je takva da kopiranje i međusobno dodjeljivanje njenih primjeraka može dovesti do neželjenih posljedica. Ustanovite koja je to klasa i napišite sav potreban kod da se eliminišu 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 3 (1,5 poen)

Glavni program u programu iz priloga, nakon što kreira neke geometrijske objekte i registrira ih u geometrijsku kolekciju, trebao bi na ekranu ispisati osnovne informacije o registriranim objektima. Međutim, program ima bag zbog kojeg ispis nije u skladu sa očekivanjima. Šta će ovaj program uopće ispisati? Šta treba popraviti da ispis bude u skladu sa očekivanjima? Kako će tada ispis izgledati (navedite tačan izgled ekrana)?

Zadatak 4 (1 poen)

Konstruktor klase “Mnogougao” je donekle nesiguran, jer ukoliko alokacija niza “x” uspije a alokacija niza “y” ne uspije, imaćemo curenje memorije. Obavite izmjene kojima se rješava opisani problem.

Zadatak 5 (2 poena)

Funkcija “DodajTjeme” u klasi “Mnogougao” prima kao parametar redni broj tjemena, te x i y koordinate tjemena, ali ne provjerava da li je redni broj tjemena u dozvoljenom opsegu. Prepravite ovu funkciju da baca izuzetak tipa “range_error” uz prikladan prateći tekst ukoliko redni broj tjemena nije u legalnom opsegu. Također, dodajte još jednu verziju metode “DodajTjeme” koja umjesto x i y koordinata tjemena prima objekat tipa “Tacka” kao odgovarajući parametar preko kojeg se zadaje tjeme (boja tačke se ignorira), te dvije pristupne metode u klasu “Tacka” kojima se omogućava saznavanje informacija o x odnosno y koordinati tačke.

Zadatak 6 (2 poena)

Geometrijska kolekcija iz ovog programa ograničena je samo na dvije vrste objekata: tačke i poligone. Dodajte mogućnost da se u ovu kolekciju mogu dodavati i krugovi. Krugovi trebaju biti opisani sa bojom, centrom i poluprečnikom, pri čemu se centar može zadavati na dva načina: kao dva realna broja (x i y koordinata) ili kao objekat tipa “Tacka”. Treba biti podržano da se mogu saznavati informacije o centru kruga i poluprečniku, kao i ispis informacija o krugu u obliku “ $\{\{x, y\}, r\}$ ”.

Zadatak 7 (1 poen)

Dodajte mogućnost da postanu legalni izrazi poput “ $k * t$ ” i “ $t * k$ ” gdje je “ k ” neki realan broj, a “ t ” objekat tipa “Tacka”. Kao rezultat ovih izraza, treba da se dobije novi objekat tipa “Tacka” u kojem su obje koordinate pomnožene sa realnim brojem “ k ”.

Zadatak 8 (1,5 poen)

Dodajte mogućnost da se za podržane geometrijske objekte može saznati informacija o njihovom obimu. Za obim tačke uzeti da je jednak 0, obim mnogougla računa se po formuli

$$O = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

gdje je n broj tjemena, a (x_i, y_i) su odgovarajuće koordinate, dok se obim kruga računa po formuli koju bi trebali znati svi koji su uopće došli raditi ovaj ispit. Mora biti podržano da se obim ma kojeg objekta može dobiti bez da se unaprijed zna o kojem se tačno tipu objekta radi.

Zadatak 9 (1,5 poen)

Dodajte u klasu "Mnogougao" mogućnost da se informacije o tjemenu mogu saznati ili zadavati navođenjem rednog broja tjemena u uglastim zgradama, tako da ako je "m" neki objekat tipa "Mnogougao", izraz "m[3]" treba da predstavlja njegovo treće tjeme i to u vidu objekta tipa "Tačka". Treba omogućiti da se na ovaj način mogu ne samo iščitavati nego i mijenjati informacije o tjemenu, pri čemu izmjena treba biti podržana samo za slučaj nekonstantnih objekata tipa "Mnogougao".

Zadatak 10 (2 poena)

Mada se, zahvaljujući pametnim pokazivačima, objekti tipa "GeometrijskaKolekcija" mogu bezbjedno kopirati i međusobno dodjeljivati, takvo kopiranje je ipak zasnovano na plitkim kopijama (bezbjednost je ostvarena preko brojanja pristupa, koje rade pametni pokazivači). Napravite neophodne izmjene koje će omogućiti da se kopiranje objekata tipa "GeometrijskaKolekcija" vrši putem dubokog kopiranja. Obratite pažnju na neophodnost polimorfnog kopiranja.

Zadatak 11 (2 poena)

Funkcija "RegistrirajObjekat" kao parametar prima pokazivač na neki geometrijski objekat, te ga upisuje u kolekciju i preuzima vlasništvo nad njim (pretpostavljajući da je on dinamički alociran). Dodajte još jednu verziju funkcije "RegistrirajObjekat" koja umjesto pokazivača prima *referencu* na neki geometrijski objekat. Za razliku od prethodne funkcije, ova funkcija neće praviti nikakve pretpostavke kako je objekat kreiran i neće ga uzimati u vlasništvo. Umjesto toga, funkcija treba da dinamički kreira kopiju objekta koji se registira i da preuzme vlasništvo nad tako kreiranom kopijom. Napomena: vjerovatno ćete se moći poslužiti nekim stvarima iz prethodnog zadatka.

Zadatak 12 (2 poena)

Napišite funkciju u klasi "GeometrijskaKolekcija" koja kao parametar ima ime datoteke, a omogućava da se podaci o geometrijskim objektima pročitaju iz tekstualne datoteke. Datoteka je organizirana na sljedeći način. U prvom redu datoteke nalazi se broj geometrijskih objekata. Nakon toga, sljedeći podaci se redom ponavljaju za svaki objekat. Podaci za jedan objekat su u jednom redu. Na početku reda se nalazi znak "T", "P" ili "K", ovisno da li je u pitanju tačka, poligon ili krug. Dalje su svi podaci koji slijede međusobno razdvojeni razmacima. Za slučaj tačke, prvo se nalazi šifra boje (cijeli broj u opsegu od 0 do 6), te x i y koordinate tačke. Za slučaj poligona, imamo šifru boje, zatim broj tjemena, te koordinate tjemena (onoliko parova brojeva koliko ima tjemena). Za slučaj kruga imamo šifru boje, zatim koordinate centra, te na kraju poluprečnik. Ukoliko su već postojali podaci u sistemu, oni prethodno trebaju biti obrisani. Radi jednostavnosti, pretpostavite da datoteka sadrži isključivo ispravne podatke.

Prilog:

```
#include <iostream>
#include <vector>
#include <memory>

class GeometrijskiObjekat {
public:
    enum Boje {Crvena, Zelena, Zuta, Plava, Ljubicasta, Crna, Bijela};
    GeometrijskiObjekat(Boje boja) : boja(boja) {}
    Boje DajBoju() const { return boja; }
    void Ispisi() const { std::cout << boja; };
private:
    Boje boja;
};

class Tacka : public GeometrijskiObjekat {
    double x, y;
public:
    Tacka(GeometrijskiObjekat::Boje boja, double x, double y) :
        GeometrijskiObjekat(boja), x(x), y(y) {}
    void Ispisi() const { std::cout << "{" << x << "," << y << "}"; }
};

class Mnogougao : public GeometrijskiObjekat {
    int broj_tjemena;
    double *x, *y;
public:
    Mnogougao(GeometrijskiObjekat::Boje boja, int broj_tjemena) :
        GeometrijskiObjekat(boja), broj_tjemena(broj_tjemena),
        x(new double[broj_tjemena]), y(new double[broj_tjemena]) {}
    void PostaviTjeme(int redni_broj, double x, double y) {
        Mnogougao::x[redni_broj - 1] = x; Mnogougao::y[redni_broj - 1] = y;
    }
    void Ispisi() const {
        std::cout << "{";
        for(int i = 0 ; i < broj_tjemena; i++) {
            std::cout << "{" << x[i] << "," << y[i] << "}";
            if(i != broj_tjemena - 1) std::cout << ",";
        }
        std::cout << "}";
    }
};

class GeometrijskaKolekcija {
    std::vector<std::shared_ptr<GeometrijskiObjekat>> objekti;
public:
    GeometrijskaKolekcija() {}
    void RegistrirajObjekat(GeometrijskiObjekat *obj) {
        objekti.push_back(std::shared_ptr<GeometrijskiObjekat>(obj));
    }
    void IspisiSve() {
        for(int i = 0; i < objekti.size(); i++) {
            objekti[i]->Ispisi();
            std::cout << std::endl;
        }
    }
};

int main() {
    GeometrijskaKolekcija g;
    g.RegistrirajObjekat(new Tacka(GeometrijskiObjekat::Crvena, 30, 20));
    Mnogougao *m(new Mnogougao(GeometrijskiObjekat::Plava, 4));
    m->PostaviTjeme(1, 10, 10); m->PostaviTjeme(2, 10, 50);
    m->PostaviTjeme(3, 50, 50); m->PostaviTjeme(4, 50, 10);
    g.RegistrirajObjekat(m);
    g.IspisiSve();
}
```