POPRAVNI ISPIT IZ PREDMETA "TEHNIKE PROGRAMIRANJA" (I PARCIJALNI)

Zadatak 1. (5 poena)

Utvrdite šta će ispisati sljedeći program (napomenimo da ASCII kod znaka '4' glasi 52). Odgovor mora biti obrazložen:

```
#include <iostream>
#include <iomanip>
#include <cstring>
#include <complex>
#include <string>

using namespace std;

int f(char x) { return x; }
int f(int x) { return x + 1; }
int f(float x) { return 5 / x; }
int f(double x) { return x; }
int f(const char x[]) { return strlen(x); }
int f(string x) { return x.size() + 1; }

template <typename Tip>
int g(Tip x) { return f(x); }

int main() {
  string s("4");
  cout << g(4) << g<float>(4) << g(4.) << g('4') << g<int>('4') << endl;
  cout << g("4") << g(s) << g("444") << g<string>("444") << endl;
  int a(5), b(a), &c(a);
  const int d(a), &e(a), &f(a + 0);
  a += 2; cout << a << b << c << d << e << f << endl;
  double u(9); complex<double> v(u), w(-v);
  cout << sqrt(u) << setw(7) << sqrt(v) << " " << sqrt(w) << endl;
  return 0;
}
```

Zadatak 2. (2,5 poena)

Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja kao rezultat vraća novi vektor čiji su elementi faktorijeli odgovarajućih elemenata vektora koji je zadan kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 6, 3, 2, 7, 1, 0, 4 i 5, funkcija treba da kao rezultat vrati vektor čiji su elementi 720, 6, 2, 5040, 1, 1, 24 i 120. Ukoliko je neki od elemenata vektora negativan, funkcija treba da baci izuzetak. Napišite i kratki isječak programa u kojem ćete demonstrirati kako se koristi napisana funkcija (obavezno predvidite i hvatanje eventualno bačenog izuzetka).

Zadatak 3. (3 poena)

Napišite funkciju "`RazdvojiCifre`" sa tri parametra. Prvi parametar je neki cijeli broj. Funkcija treba da formira dva nova broja koji se sastoje respektivno od parnih i neparnih cifara polaznog broja, u istom redoslijedu u kojem se nalaze u polaznom broju. Novoformirane brojeve treba smjestiti redom u drugi i treći parametar funkcije. Na primjer, ukoliko se kao prvi parametar zada broj 32564719, u drugi i treći parametar treba da se redom smjeste brojevi 264 i 35719. Znak broja treba ignorirati, odnosno isti efekat se dobija ukoliko se kao prvi parametar zada broj –32564719. Napišite i kratki isječak programa u kojem ćete demonstrirati kako se koristi napisana funkcija.

Zadatak 4. (3 poena)

Nazovimo neku riječ "korektnom" ukoliko se u njoj ne pojavljuje skupina uzastopnih suglasnika duža od 2 slova. Na primjer, riječ "badlekuvje" je korektna, a riječ "balgftekrihvnu" nije korektna u smislu date definicije, jer sadrži 2 grupe suglasnika sa više od 2 uzastopna suglasnika ("lgft" i "hvn"). Napišite funkciju koja prihvata string kao parametar, a koja kao rezultat vraća logičku vrijednost "tačno" ukoliko string koji joj je proslijeđen kao parametar predstavlja korektnu riječ u smislu gore date definicije, a u suprotnom vraća logičku vrijednost "netačno". Napisanu funkciju demonstrirajte u isječku programa koji testira da li je riječ unesena sa tastature korektna ili ne.

Zadatak 5. (2,5 poena)

Napišite generičku funkciju sa 3 parametra *p1*, *p2* i *f*. Parametri *p1* i *p2* omeđuju neki blok podataka neodređenog tipa (*p1* pokazuje na prvi element bloka a *p2* pokazuje iza posljednjeg elementa bloka), dok je *f* funkcija koja kao parametar prima neki element istog tipa kao tip elemenata bloka, a vraća logičku vrijednost kao rezultat. Funkcija koju trebate napisati kao rezultat vraća sumu svih elemenata unutar bloka za koje funkcija *f* daje kao rezultat logičku vrijednost "tačno". Napisanu funkciju demonstrirajte u isječku programa koji ispisuje zbir svih pozitivnih brojeva u vektoru unesenom sa tastature.

Zadatak 6. (4 poena)

Napišite generičku funkciju koja kao svoj parametar prima dvodimenzionalnu strukturu (recimo matricu, ali pri čemu broj elemenata u svakom redu ne mora nužno biti isti) predstavljenu kao vektor vektōrâ čiji su elementi proizvoljnog tipa. Funkcija treba dinamički alokacirati prostor za dvodimenzionalnu strukturu identičnog oblika kao i parametar, zatim da u nju prepiše elemente dvodimenzionalne strukture predstavljene parametrom i, konačno, da kao rezultat vrati dvojni pokazivač preko kojeg se može izvršiti pristup elementima ove strukture. U slučaju da dođe do problema sa alokacijom memorije, funkcija treba baciti izuzetak. Pri tome, ni u kom slučaju ne smije doći do curenja memorije. Napisanu funkciju demonstrirajte u testnom programu koji kreira matricu formata 3×3 predstavljenu kao vektor vektōrâ, unosi njene elemente sa tastature, zatim poziva napisanu funkciju da kreira novu (dinamički kreiranu) dvodimenzionalnu strukturu, ispisuje njene elemente na ekran i, na kraju, oslobađa memoriju koju je zauzela novokreirana struktura.

POPRAVNI ISPIT IZ PREDMETA "TEHNIKE PROGRAMIRANJA" (II PARCIJALNI)

Zadatak 1. (8 poena)

Za potrebe neke aplikacije za dvodimenzionalnu kompjutersku grafiku, potrebno je razviti klasu "`Tacka`" koja predstavlja tačku u ravni, sa sljedećim interfejsom:

```
Tacka();
Tacka(double x, double y);
void Postavi(double x, double y);
void PostaviPolarno(double ro, double theta);
double DajX() const;
double DajY() const;
double DajRo() const;
double DajTheta() const;
void PostaviX(double x);
void PostaviY(double y);
void Transliraj(double delta_x, double delta_y);
void Rotiraj(double alpha, const Tacka &centar = Tacka());
```

Konstruktor bez parametara kreira tačku u koordinatnom početku, dok konstruktor sa dva parametra kreira tačku na osnovu zadanih Dekartovih koordinata x i y. Istu stvar radi i metoda "`Postavi`" (osim što omogućava naknadnu promjenu pozicije tačke. Metoda "`PostaviPolarno`" radi sličnu stvar samo na osnovu polarnih koordinata ρ i θ (podsjetimo se da je veza između Dekartovih i polarnih koordinata data formulama $x = \rho\cos\theta$, $y = \rho\sin\theta$ odnosno $\rho = \sqrt{x^2 + y^2}$, $\theta = \text{atan2}(x, y)$, gdje je "atan2" funkcija srodna funkciji "arkus tangens", a podržana je u jezicima C/C++ baš pod tim imenom). Metode "`DajX`", "`DajY`", "`DajRo`" i "`DajTheta`" vraćaju kao rezultat odgovarajuće Dekartove odnosno polarne koordinate tačke, dok metode "`PostaviX`" i "`PostaviY`" omogućavaju promjenu x odnosno y koordinate tačke (bez promjene preostale koordinate). Metoda "`Transliraj`" pomjera tačku za iznos Δx u smjeru x-ose i iznos Δy u smjeru y-ose, pri čemu se vrijednosti Δx i Δy navode kao parametri, dok metoda "`Rotiraj`" rotira tačku za ugao α koji se zadaje kao prvi parametar (u smjeru suprotnom od kazaljke na satu) oko tačke koja se zadaje kao drugi parametar. Drugi parametar kao podrazumijevanu vrijednost ima tačku kreiranu u koordinatnom početku, čime je postignuto da se može zadati samo jedan parametar (pri čemu se tada rotacija vrši oko koordinatnog početka). Za one koji nisu dovoljno upućeni (čitaj: većina koja izlazi na ovaj ispit), napomenimo da se rotacijom tačke (x, y) oko tačke (x_c, y_c) za ugao α dobija tačka (x', y') gdje je $x' = x_c + (x - x_c)\cos\alpha - (y - y_c)\sin\alpha$, $y' = y_c + (x - x_c)\sin\alpha + (y - y_c)\cos\alpha$.

Pored funkcija pobrojanih u interfejsu, potrebno je podržati i neke operatore. Operatori "`==`" i "`!=`" testiraju da li su dvije tačke jednake odnosno različite i vraćaju logičku vrijednost "tačno" ako jesu, odnosno "netačno" ako nisu. Operator "`!`" vraća logičku vrijednost "tačno" samo ako se primijeni na tačku koja se nalazi u koordinatnom početku, inače vraća logičku vrijednost "netačno". Konačno, operator "–" primijenjen na dvije tačke vraća kao rezultat njihovo međusobno rastojanje.

a) Implementirajte klasu sa navedenim osobinama, pri čemu ćete uzeti da su atributi klase Dekartove koordinate tačke x i y.

b) Pri razvoju aplikacije koja koristi klasu "`Tacka`", uočeno je da će neki vitalni algoritmi u aplikaciji mnogo brže raditi ukoliko se koordinate tačke interno čuvaju u polarnom koordinatnom sistemu, te da je pogodnije kao atribute klase čuvati njene polarne koordinate ρ i θ. Implementirajte ponovo klasu "`Tacka`" koristeći ovako izmijenjen dizajn, ali tako da nova klasa zadrži isti interfejs, odnosno da korisnici klase (bar sa aspekta njene upotrebe) ne primijete da je njena implementacija izmijenjena.

NAPOMENA: Ukoliko neke od metoda pametno napišete, nećete ih uopće morati mijenjati u novoj verziji klase (ideja je pisati metode tako da se oslanjaju na druge metode). U suprotnom, možda ćete morati iznova napisati sve metode, što nije baš relaksirajuće.

Zadatak 2. (12 poena)

Uprava ETF-a odlučila je da uvede automatizirani fast-food restoran. Za potrebe automatske obrade narudžbi, potrebno je razviti program zasnovan na skupini klasa, prema opisu koji slijedi.

Klasa "`Obrok`" opisuje najjednostavniju narudžbu, koja predstavlja neki obrok. Ova klasa od atributa sadrži naziv obroka (recimo "Burek"), cijenu obroka (realan broj) i broj indeksa studenta koji je naručio obrok. Pored ovih atributa, klasa sadrži samo konstruktor koji inicijalizira atribute na vrijednosti zadane parametrima, trivijalne pristupne metode koje vraćaju vrijednost svih atributa, metodu koja ispisuje podatke o narudžbi (ispis formatirajte prema vlastitoj želji) kao i metodu koja vraća ukupnu cijenu narudžbe. U slučaju klase "`Obrok`", ova metoda će samo vratiti cijenu obroka pohranjenu u atributu, ali treba predvidjeti da će ova metoda biti eventualno izmijenjena u složenijim klasama naslijeđenim iz klase "`Obrok`" kod kojih se ukupna cijena obroka formira na složeniji način. Pri tome, u slučaju posljednje dvije metode, treba biti omogućeno da se preko pokazivača na klasu "`Obrok`" uvijek pozivaju ispravne verzije ovih metoda, bez obzira na tip objekta na koji taj pokazivač pokazuje u trenutku poziva.

Klasa "`ObrokSaPicem`" opisuje narudžbu koja uz obrok uključuje i piće. Ova klasa naslijeđena je iz klase "`Obrok`", a od dodatnih atributa sadrži i naziv pića (recimo "Fanta") kao i cijenu pića. U skladu s tim, klasa ima i dvije nove trivijalne pristupne metode koje vraćaju vrijednosti novouvedenih atributa Konstruktor ove klase je izmijenjen u odnosu na konstruktor klase "`Obrok`" da omogući i inicijalizaciju dopunskih atributa. Metoda koja vraća ukupnu cijenu narudžbe modificirana je tako da vrati ukupnu cijenu koja uključuje kako cijenu obroka, tako i cijenu pića. Isto tako, modificirana je i metoda za ispis podataka o narudžbi, koja sada treba uključiti i dopunske informacije koje nisu postojale u klasi "`Obrok`", uključujući i informacije o ukupnoj cijeni.

Klasa "`Narudzbe`" predstavlja kolekciju narudžbi, izvedenu kao dinamički niz pokazivača na objekte tipa "`Obrok`" ili "`ObrokSaPicem`". Ova klasa sadrži konstruktor sa jednim parametrom, koji vrši odgovarajuću dinamičku alokaciju memorije, pri čemu parametar predstavlja maksimalan broj zahtjeva koji se mogu pohraniti. Pored toga, klasa sadrži i odgovarajući destruktor, koji oslobađa memoriju koju je objekat tipa "`Narudzbe`" zauzeo tokom svog života. Konstruktor kopije i preklopljeni operator dodjele nisu predviđeni, ali je kopiranje i međusobno dodjeljivanje primjeraka klase "`Narudzbe`" potrebno zabraniti. Od metoda, klasa "`Narudzbe`" posjeduje četiri metode. Prve dvije metode vrše kreiranje i dodavanje nove narudžbe, pri čemu se prva metoda odnosi na prostu narudžbu (obrok bez pića), dok se druga metoda odnosi na narudžbu obroka sa pićem. Parametri ovih metoda su isti kao i konstruktori klasa "`Obrok`" odnosno "`ObrokSaPicem`". One kreiraju novu narudžbu (u skladu sa navedenim parametrima) i smještaju ga u kolekciju. Treća metoda nema parametara. Ona vrši ispis podataka o prvoj primljenoj narudžbi na ekran (tj. narudžbi koja najduže čeka na isporuku). Pored toga, ova metoda vrši brisanje obrađene narudžbe iz kolekcije, tako da će sljedeći poziv ove metode ispisati sljedeću naredbu po redu, i tako dalje, sve dok se ne obrade sve narudžbe. U slulaju da se ova metoda pozove kada je kolekcija prazna, metoda treba da baci izuzetak. Posljednja metoda koju posjeduje klasa "`Narudzbe`" je metoda bez parametara koja daje kao rezultat logičku vrijednost "**`true`**" ako i samo ako objekat nad kojim je primijenjena predstavlja praznu kolekciju, a u suprotnom daje kao rezultat logičku vrijednost "**`false`**". Konačno, klasa "`Narudzbe`" posjeduje i preklopljeni operator "`[]`" koji kao rezultat daje ukupnu cijenu narudžbi koje je naručio student sa brojem indeksa navedenim unutar uglastih zagrada (tj. zbir cijena svih narudžbi tog studenta). Ukoliko student sa zadanim brojem indeksa nije ništa naručio, ovaj operator kao rezultat daje nulu.

Definirajte i implementirajte klase sa navedenim osobinama. Napisane klase demonstrirajte u testnom programu koji čita podatke o narudžbama iz tekstualne datoteke (format datoteke organizirajte po vlastitoj želji) i smješta ih u kolekciju, nakon čega ispisuje sve unesene narudžbe u redoslijedu pristizanja.