

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 (prikaz bez obrazloženja biće shvaćen kao prepisivanje i NEĆE BITI PRIHVACĀEN). Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <complex>

using namespace std;

int p(double q, int p = 1) { return p * q; }
void q(int *&p, int *&q) { cout << *q-- << *++p << " "; }
double p(double r, double p, int q = 2) { return p - q * r; }
int &r(double (*u)(double), int &v, int &w, int &s) {
    cout << setw(6) << v << w << s << endl;
    v = w - s; s = v - w;
    cout << w << v << s << endl;
    w = (*u)(w);
    return v;
}

void s(const complex<double> &q) { cout << q + 1. << endl; }

int main() {
    enum a {b, c = 3, d, e = 7, f};
    int u[] = {9, 5, 2, 7, 3, 4, 6, 9, 2, 3};
    int *w(&u[3]), *x(u + 1), *y(w + 6);
    s(3); s(complex<double>(3, 3));
    while(x <= y) q(x, y);
    cout << endl << p(2, 3) << " " << p(3, 2, 2.3) << " " << p(3.2)
        << " " << p(2, 3.) << endl;
    r(sqrt, u[2], *u, w[-1]) += 2;
    cout << w[-1] << " " << *u << " " << u[2] << endl;
    cout << b << c << d << e << f << endl;
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja odstranjuje sve proste brojeve iz vektora cijelih brojeva koji se zadaje kao njen jedini parametar, zadržavajući ostale elemente u istom poretku kakvi su bili prije odstranjivanja. Funkcija ne vraća nikakvu vrijednost, već samo utiče na elemente vektora koji joj je prosljeđen kao parametar (koji pri tome, naravno, može promijeniti svoju veličinu). Pri tome, funkcija ne smije u svom radu kreirati i koristiti nikakve druge vektore ili nizove osim vektora koji joj je prosljeđen kao parametar, niti koristiti ikakve pomoćne funkcije. Također, nije dozvoljeno koristiti nikakve bibliotečke funkcije čija upotreba nije bila demonstrirana na predavanjima (poput funkcije “erase” primijenjene na vektore). Napisanu funkciju demonstrirajte u isječku programa koji će iz spiska brojeva koji se unose sa tastature odstraniti sve proste brojeve i ispisati sve brojeve nakon obavljenog odstranjivanja. Podsjetimo se da su prosti brojevi oni brojevi koji imaju tačno dva djelioca: jedinicu i samog sebe.

Zadatak 3 (2,5 poena)

Za neki element matrice kaŹemo da je *stacionaran* ukoliko je njegova vrijednost jednaka vrijednosti elemenata koji se nalaze lijevo, desno, iznad i ispod njega. Napišite funkciju koja kao parametar prima



neku matricu organiziranu kao vektor vektora, a kao rezultat daje logičku vrijednost “tačno” ili “netačno” u ovisnosti od toga da li matrica ima stacionarnih elemenata ili ne. Funkcija treba da radi sa matricama proizvoljnog tipa elemenata. U slučaju da parametar nema formu matrice (tj. ukoliko svi redovi nemaju jednak broj elemenata), funkcija treba da baci izuzetak. Napišite i kratki isječak programa u kojem ćete demonstrirati kako se može iskoristiti napisana funkcija, uz pretpostavku da su elementi matrice već uneseni (drugim riječima, ne trebate pisati dio programa koji unosi elemente matrice). Pretpostavite da su elementi matrice realni brojevi. Obavezno predvidite i hvatanje eventualno bačenog izuzetka. Napomena: s obzirom da matrica može zaista biti masivna struktura podataka, izbjegnite po svaku cijenu kopiranje formalnog parametra u stvarni.

Zadatak 4 (2,5 poena)

Napišite funkciju koja kao parametar prima jedan string (tj. parametar tipa “string”), a kao rezultat vraća vektor stringova u kojem pojedini stringovi predstavljaju individualne riječi unutar te rečenice. Pri tome se pod pojmom “riječ” podrazumijeva svaka skupina susjednih znakova među kojima nema razmaka, takva da se ispred prvog znaka nalazi razmak (osim ukoliko se prvi znak nalazi na samom početku stringa) i da se iza posljednjeg znaka nalazi razmak (osim ukoliko se posljednji znak nalazi na samom kraju stringa). Napisanu funkciju testirajte u isječku programa koji će za rečenicu unesenu sa tastature ispisati sve njene riječi, svaku riječ u posebnom redu.

Zadatak 5 (2,5 poena)

Napišite funkciju sa jednim cjelobrojnim parametrom n koja vrši dinamičku alokaciju “grbave matrice” sa n redova u kojoj prvi red sadrži samo element 1, drugi red elemente 1, 2 i 1, treći red elemente 1, 2, 3, 2 i 1, četvrti red elemente 1, 2, 3, 4, 3, 2 i 1, itd. Kao rezultat, funkcija vraća dvostruki pokazivač preko kojeg se može pristupiti elementima tako kreirane grbave matrice. Ukoliko je n negativan ili 0, funkcija treba baciti izuzetak. Funkcija također treba baciti izuzetak u slučaju da alokacija ne uspije, pazeći pri tome da ne dođe do curenja memorije ni u kakvim okolnostima. Napisanu funkciju iskoristite u isječku programa koji traži da se sa tastature unese broj n , zatim kreira “grbavu matricu” traženih svojstava sa n redova, te na kraju ispisuje njene elemente red po red nakon čega oslobađa prostor koji je zauzela ta matrica. Predvidjeti i hvatanje svih izuzetaka koji bi funkcija eventualno mogla baciti.

Zadatak 6 (2,5 poena)

Napravite generičku funkciju koja kao parametre prima dva pokazivača ili iteratora koji omeđuju neki blok podataka, te jednu funkciju koja prima jedan parametar i vraća rezultat istog tipa kao što je i parametar. Za elemente bloka se pretpostavlja da se mogu međusobno upoređivati. Funkcija treba da kao rezultat vrati pokazivač ili iterator koji pokazuje na element unutar bloka za koji funkcija koja joj je prenesena kao treći parametar ima maksimalnu vrijednost. Ukoliko takvih elemenata ima više, funkcija vraća pokazivač ili iterator na prvi od njih. Funkcija se mora zasnivati na pokazivačkoj odnosno iteratorskoj aritmetici (tj. nije dozvoljeno indeksirati pokazivač, ili koristiti trivijalnu simulaciju indeksiranja oblika “*(p+i)”). Napišite i kratki isječak programa koji će u nekom nizu od 10 elemenata ispisati element koji ima najveći sinus.

Zadatak 7 (2,5 poena)

Pretpostavimo da je potrebno sortirati niz brojeva uz pomoć funkcije “sort” u takav poredak da brojevi sa manje cifara dođu ispred brojeva sa više cifara, ali da među brojevima sa jednakim brojem cifara veći brojevi dolaze ispred manjih brojeva. Na primjer, broj 23 treba doći ispred broja 4387 (jer broj 23 ima manje cifara od broja 4387), ali među brojevima 5428 i 4387 koji imaju jednak broj cifara, broj 5428 treba doći ispred broja 4387. Napišite funkciju kriterija koja omogućava da funkcija “sort” obavi ovakvo sortiranje. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi 10 cijelih brojeva u neki dek, nakon čega se vrši njihovo sortiranje u skladu sa opisanim kriterijem te ispis tako sortiranog deka na ekran.

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 (prikaz bez obrazloženja biće shvaćen kao prepisivanje i NEĆE BITI PRIHVAĆEN). Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <complex>

using namespace std;

int p(double q, int p = 1) { return p * q; }
void q(int *&p, int *&q) { cout << *q-- << *++p << " "; }
double p(double r, double p, int q = 2) { return p - q * r; }
int &r(double (*u)(double), int &v, int &w, int &s) {
    cout << setw(5) << v << w << s << endl;
    v = w - s; s = v - w;
    cout << w << v << s << endl;
    w = (*u)(w);
    return v;
}

void s(const complex<double> &q) { cout << q - 1. << endl; }

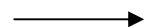
int main() {
    enum a {b, c = 2, d, e = 8, f};
    int u[] = {9, 5, 2, 7, 3, 4, 6, 9, 2, 3};
    int *w(&u[3]), *x(u + 1), *y(w + 6);
    s(2); s(complex<double>(2, 2));
    while(x <= y) q(x, y);
    cout << endl << p(3, 2) << " " << p(2, 3, 3.2) << " " << p(2.3)
        << " " << p(3, 2.) << endl;
    r(sqrt, u[2], *u, w[-1]) += 3;
    cout << w[-1] << " " << *u << " " << u[2] << endl;
    cout << b << c << d << e << f << endl;
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja odstranjuje sve složene brojeve iz vektora cijelih brojeva koji se zadaje kao njen jedini parametar, zadržavajući ostale elemente u istom poretku kakvi su bili prije odstranjivanja. Funkcija ne vraća nikakvu vrijednost, već samo utiče na elemente vektora koji joj je prosljeđen kao parametar (koji pri tome, naravno, može promijeniti svoju veličinu). Pri tome, funkcija ne smije u svom radu kreirati i koristiti nikakve druge vektore ili nizove osim vektora koji joj je prosljeđen kao parametar, niti koristiti ikakve pomoćne funkcije. Također, nije dozvoljeno koristiti nikakve bibliotečke funkcije čija upotreba nije bila demonstrirana na predavanjima (poput funkcije “erase” primijenjene na vektore). Napisanu funkciju demonstrirajte u isječku programa koji će iz spiska brojeva koji se unose sa tastature odstraniti sve složene brojeve i ispisati sve brojeve nakon obavljenog odstranjivanja. Podsjetimo se da su složeni brojevi oni brojevi koji imaju više od dva djelioca.

Zadatak 3 (2,5 poena)

Za neku matricu kažemo da je *nestacionarna* ukoliko niti jedan njen element nema osobinu da je njegova vrijednost jednaka vrijednosti elemenata koji se nalaze lijevo, desno, iznad i ispod njega. Napišite funkciju koja kao parametar prima neku matricu organiziranu kao vektor vektora, a kao



rezultat daje logičku vrijednost “tačno” ili “netačno” u ovisnosti od toga da li je matrica nestacionarna ili ne. Funkcija treba da radi sa matricama proizvoljnog tipa elemenata. U slučaju da parametar nema formu matrice (tj. ukoliko svi redovi nemaju jednak broj elemenata), funkcija treba da baci izuzetak. Napišite i kratki isječak programa u kojem ćete demonstrirati kako se može iskoristiti napisana funkcija, uz pretpostavku da su elementi matrice već uneseni (drugim riječima, ne trebate pisati dio programa koji unosi elemente matrice). Pretpostavite da su elementi matrice realni brojevi. Obavezno predvidite i hvatanje eventualno bačenog izuzetka. Napomena: s obzirom da matrica može zaista biti masivna struktura podataka, izbjegnite po svaku cijenu kopiranje formalnog parametra u stvarni.

Zadatak 4 (2,5 poena)

Napišite funkciju koja kao parametar prima jedan string (tj. parametar tipa “string”), a kao rezultat vraća vektor stringova u kojem pojedini stringovi predstavljaju individualne riječi unutar te rečenice. Pri tome se pod pojmom “riječ” podrazumijeva svaka skupina susjednih znakova među kojima nema razmaka, takva da se ispred prvog znaka nalazi razmak (osim ukoliko se prvi znak nalazi na samom početku stringa) i da se iza posljednjeg znaka nalazi razmak (osim ukoliko se posljednji znak nalazi na samom kraju stringa). Napisanu funkciju testirajte u isječku programa koji će za rečenicu unesenu sa tastature ispisati sve njene riječi, svaku riječ u posebnom redu.

Zadatak 5 (2,5 poena)

Napišite funkciju sa jednim cjelobrojnim parametrom n koja vrši dinamičku alokaciju “grbave matrice” sa n redova u kojoj prvi red sadrži samo element 1, drugi red elemente 2, 1 i 2, treći red elemente 3, 2, 1, 2 i 3, četvrti red elemente 4, 3, 2, 1, 2, 3 i 4, itd. Kao rezultat, funkcija vraća dvostruki pokazivač preko kojeg se može pristupiti elementima tako kreirane grbave matrice. Ukoliko je n negativan ili 0, funkcija treba baciti izuzetak. Funkcija također treba baciti izuzetak u slučaju da alokacija ne uspije, pazeći pri tome da ne dođe do curenja memorije ni u kakvim okolnostima. Napisanu funkciju iskoristite u isječku programa koji traži da se sa tastature unese broj n , zatim kreira “grbavu matricu” traženih svojstava sa n redova, te na kraju ispisuje njene elemente red po red nakon čega oslobađa prostor koji je zauzela ta matrica. Predvidjeti i hvatanje svih izuzetaka koji bi funkcija eventualno mogla baciti.

Zadatak 6 (2,5 poena)

Napravite generičku funkciju koja kao parametre prima dva pokazivača ili iteratora koji omeđuju neki blok podataka, te jednu funkciju koja prima jedan parametar i vraća rezultat istog tipa kao što je i parametar. Za elemente bloka se pretpostavlja da se mogu međusobno upoređivati. Funkcija treba da kao rezultat vrati pokazivač ili iterator koji pokazuje na element unutar bloka za koji funkcija koja joj je prenesena kao treći parametar ima minimalnu vrijednost. Ukoliko takvih elemenata ima više, funkcija vraća pokazivač ili iterator na prvi od njih. Funkcija se mora zasnivati na pokazivačkoj odnosno iteratorskoj aritmetici (tj. nije dozvoljeno indeksirati pokazivač, ili koristiti trivijalnu simulaciju indeksiranja oblika “*(p+i)”). Napišite i kratki isječak programa koji će u nekom nizu od 10 elemenata ispisati element koji ima najmanji kosinus.

Zadatak 7 (2,5 poena)

Pretpostavimo da je potrebno sortirati niz brojeva uz pomoć funkcije “sort” u takav poredak da brojevi sa više cifara dođu ispred brojeva sa manje cifara, ali da među brojevima sa jednakim brojem cifara manji brojevi dolaze ispred većih brojeva. Na primjer, broj 4387 treba doći ispred broja 23 (jer broj 4387 ima više cifara od broja 23), ali među brojevima 5428 i 4387 koji imaju jednak broj cifara, broj 4387 treba doći ispred broja 5428. Napišite funkciju kriterija koja omogućava da funkcija “sort” obavi ovakvo sortiranje. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi 10 cijelih brojeva u neki dek, nakon čega se vrši njihovo sortiranje u skladu sa opisanim kriterijem te ispis tako sortiranog deka na ekran.