

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 PRIHVAĆEN). Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
#include <iostream>
#include <set>

template <typename T, typename F>
void f(T p, T q, F r) { while(p != q) r(*--q); }

int &g(int &p, int q, int &r) {
    std::cout << p << " " << q << " " << r << std::endl;
    p = q + r; q = p + r; r = p + q;
    std::cout << p << " " << q << " " << r << std::endl;
    return p;
}

void h(int **p, int *q) { *p = q; }

int main() {
    int a[] {3, 5, 4, 6, 2, 3, 7}, *p(a + 4), q(8), r(4);
    f(a, a + 5, [&r](int x) { std::cout << x << ", " << r << " "; r += x; });
    std::cout << r << std::endl;
    g(*(a + 2), a[4], p[-2]) *= 2;
    for(int p : a) std::cout << p << " "; std::cout << std::endl;
    h(&p, &q); std::cout << p[0] << " ";
    h(&p, new int(2)); std::cout << p[0] << std::endl;
    std::set<int> s;
    s.insert(3); s.insert(5); s.insert(2); s.insert(3); s.insert(7);
    for(auto i = s.begin(); i != s.end(); i++) std::cout << *i << " ";
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a kao rezultat vraća vektor logičkih vrijednosti iste dimenzije, pri čemu je i -ti element vektora koji se vraća kao rezultat logička vrijednost “tačno” ukoliko je i -ti element ulaznog vektora prost broj, inače je logička vrijednost “netačno” (prosti brojevi su oni cijeli brojevi koji imaju tačno dva djelioca u skupu prirodnih brojeva, jedinicu i samog sebe). Napisanu funkciju demonstrirajte u isječku programa koji unosi slijed od “n” brojeva sa tastature (“n” se također unosi sa tastature), smješta ih u vektor, šalje taj vektor napisanoj funkciji i na kraju koristi njen rezultat da izračuna i ispiše koliko među unesenim brojevima ima prostih brojeva.

Zadatak 3 (2,5 poena)

Minimax vrijednost matrice po redovima definira se tako što se za svaki red matrice nađe najveća vrijednost u tom redu, a onda se uzme najmanja od tako nađenih najvećih vrijednosti po redovima. Analogno se definira i *maximin vrijednost matrice po redovima*. Napišite funkciju koja kao prvi parametar prima neku matricu organiziranu kao vektor vektora realnih brojeva, a koja u svoj drugi i treći parametar smješta minimax odnosno maximin vrijednosti po redovima za tu matricu. U slučaju da prvi parametar nema oblik matrice (tj. ukoliko njegovi redovi nemaju isti broj elemenata), treba baciti izuzetak tipa “domain_error” uz prateći tekst “Parametar nema formu matrice”. Funkcija ne vraća nikakav rezultat. Za realizaciju funkcije nije dozvoljeno koristiti nikakve pomoćne nizove ili vektore. Napišite i kratki isječak programa koji prikazuje kako se koristi napisana funkcija (uključujući i hvatanje eventualno bačenog izuzetka).



Zadatak 4 (2,5 poena)

Napišite funkciju čiji je prvi parametar neki string, a drugi parametar neka funkcija koja prima znak kao parametar i vraća znak kao rezultat. Funkcija treba da transformira string zadan prvim parametrom tako što će na svaki njegov znak koji nije razmak primijeniti funkciju zadanu drugim parametrom (preciznije, svaki znak koji nije razmak zamjenjuje se rezultatom te funkcije) i pri tome ne vraća nikakav rezultat. Napisanu funkciju demonstrirajte u isječku koji neki string unesen sa tastature transformira tako da se svaki znak koji nije razmak mijenja znakom koji mu slijedi po ASCII poretku (npr. "A" postaje "B", itd.) i ispisuje tako transformirani string. Prikažite dva rješenja, jedno koje koristi imenovanu funkciju kao parametar, i drugo koje koristi lambda funkciju kao parametar.

Zadatak 5 (2,5 poena)

Napišite generičku funkciju "KopirajBezDuplikata" sa tri parametra "p1", "p2" i "p3". Funkcija prepisuje elemente bloka omeđenog sa "p1" i "p2" ("p1" pokazuje na početak a "p2" tačno iza kraja bloka) u blok na čiji početak pokazuje "p3", ali izostavljajući elemente koji su jednaki nekom elementu koji se već ranije pojavljivao u bloku (tj. izostavljajući duplikate). Funkcija treba da bude zasnovana na potpunoj dedukciji, tako da kao parametre prihvata i pokazivače i iteratore. Pored toga, funkcija bi trebala da se zasniva samo na osnovnom skupu operacija koje podržavaju svi tipovi iteratora (tako da treba da radi sa iteratorima svih vrsta, ne samo iteratorima sa direktnim pristupom). Parametri "p1" i "p2" moraju biti istog tipa, dok se tip parametra "p3" može razlikovati od njih. Kao rezultat, funkcija treba da vrati pokazivač/iterator koji pokazuje tačno iza kraja određiškog bloka (nakon obavljenog prepisivanja). Napisanu funkciju demonstrirajte u isječku programa koji će neki string unesen sa tastature prepisati u drugi string izostavljajući duplikate prisutne u prvom stringu i na kraju ispisati novodobijeni string (korištenje umetača je dozvoljeno i poželjno).

Zadatak 6 (3 poena)

Napišite funkciju sa jednim parametrom "n" koja vrši dinamičku alokaciju grbave matrice čiji prvi red ima 1 element, drugi 2 elementa, i tako dalje do posljednjeg reda koji će imati "n" redova i popunjava tako formirane redove elementima Pascalovog trougla (trokuta). Dakle, prvi red treba sadržavati element 1, drugi red elemente 1 i 1, treći red elemente 1, 2 i 1, četvrti red elemente 1, 3, 3 i 1, peti red elemente 1, 4, 6, 4 i 1, itd. (prvi i posljednji element u svakom redu jednaki su 1, a svaki ostali element jednak je zbiru elementa koji se nalazi tačno iznad njega i njegovog susjeda sa lijeve strane). Za alokaciju treba koristiti postupak *kontinualne alokacije*. Funkcija kao rezultat treba da vrati dvostruki pokazivač preko kojeg se može izvršiti pristup elementima alocirane matrice. U slučaju da se kao parametar zada negativan broj ili nula, funkcija treba da baci tekst "Pogrešan parametar" kao izuzetak, dok u slučaju da alokacija ne uspije, treba baciti izuzetak koji se sastoji od teksta "Kreiranje nije uspelo". Pri tome, ni u kom slučaju ne smije doći do curenja memorije. Napisanu funkciju demonstrirajte u isječku koji koristi napisanu funkciju da kreira dinamičku grbavu matricu koja sadrži Pascalov trougao (trokut) sa "n" redova, pri čemu se "n" unosi sa tastature i ispisuje tako kreiranu grbavu matricu na ekran. Porebno je predviditi i hvatanje eventualno bačenih izuzetaka, kao i oslobađanje prostora koji je matrica zauzimala nakon završetka rada.

Zadatak 7 (2 poena)

Razlika između najveće i najmanje cifre nekog broja naziva se njegov *raspon* (npr. broj 6735 ima raspon 4, jer je $7-3=4$). Napišite isječak programa koji će jedan fiksni dek od 10 četverocifrenih brojeva (brojeve definirajte po svojoj volji) sortirati tako da brojevi sa većim rasponom dođu ispred brojeva sa manjim rasponom i ispisati tako sortirani dek. U slučaju da dva broja imaju isti raspon, broj koji je manji treba doći prije broja koji je veći. Za sortiranje koristiti funkciju "sort" iz biblioteke "algorithm". Poželjno bi bilo da kao funkciju kriterija upotrijebite lambda funkciju, ali ukoliko ne znate to uraditi, prihvatite se i rješenje sa imenovanom funkcijom kriterija.

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 <set>

template <typename T, typename F>
void f(T p, T q, F r) { while(p != q) r(*p++); }

int &g(int &p, int q, int &r) {
    std::cout << p << " " << q << " " << r << std::endl;
    p = q - r; q = p - r; r = p - q;
    std::cout << p << " " << q << " " << r << std::endl;
    return p;
}

void h(int **p, int *q) { *p = q; }

int main() {
    int a[]{4, 2, 7, 1, 3, 6, 5}, *p(a + 2), q(3), r(6);
    f(a, a + 5, [&r](int x) { std::cout << x << ", " << r << " "; r += x; });
    std::cout << r << std::endl;
    g(*(a + 1), a[3], p[-1]) *= 4;
    for(int p : a) std::cout << p << " "; std::cout << std::endl;
    h(&p, &q); std::cout << p[0] << " ";
    h(&p, new int(9)); std::cout << p[0] << std::endl;
    std::set<int> s;
    s.insert(8); s.insert(5); s.insert(2); s.insert(5); s.insert(6);
    for(auto i = s.begin(); i != s.end(); i++) std::cout << *i << " ";
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a kao rezultat vraća vektor logičkih vrijednosti iste dimenzije, pri čemu je i -ti element vektora koji se vraća kao rezultat logička vrijednost “tačno” ukoliko je i -ti element ulaznog vektora složen broj, inače je logička vrijednost “netačno” (složeni brojevi su oni cijeli brojevi različiti od nule koji imaju barem tri različita djelioca u skupu prirodnih brojeva, dakle barem još jedan djelioc osim jedinice i samog sebe). Napisanu funkciju demonstrirajte u isječku programa koji unosi slijed od “n” brojeva sa tastature (“n” se također unosi sa tastature), smješta ih u vektor, šalje taj vektor napisanoj funkciji i na kraju koristi njen rezultat da izračuna i ispiše koliko među unesenim brojevima ima složenih brojeva.

Zadatak 3 (2,5 poena)

Minimax vrijednost matrice po kolonama definira se tako što se za svaku kolonu matrice nađe najveća vrijednost u toj koloni, a onda se uzme najmanja od tako nađenih najvećih vrijednosti po kolonama. Analogno se definira i *maximin vrijednost matrice po kolonama*. Napišite funkciju koja kao prvi parametar prima neku matricu organiziranu kao vektor vektora realnih brojeva, a koja u svoj drugi i treći parametar smješta minimax odnosno maximin vrijednosti po kolonama za tu matricu. U slučaju da prvi parametar nema oblik matrice (tj. ukoliko njegovi redovi nemaju isti broj elemenata), treba baciti izuzetak tipa “domain_error” uz prateći tekst “Parametar nema formu matrice”. Funkcija ne vraća nikakav rezultat. Za realizaciju funkcije nije dozvoljeno koristiti nikakve pomoćne nizove ili vektore. Napišite i kratki isječak programa koji prikazuje kako se koristi napisana funkcija (uključujući i hvatanje eventualno bačenog izuzetka).



Zadatak 4 (2,5 poena)

Napišite funkciju čiji je prvi parametar neki string, a drugi parametar neka funkcija koja prima znak kao parametar i vraća znak kao rezultat. Funkcija treba da transformira string zadan prvim parametrom tako što će na svaki njegov znak koji nije razmak primijeniti funkciju zadanu drugim parametrom (preciznije, svaki znak koji nije razmak zamjenjuje se rezultatom te funkcije) i pri tome ne vraća nikakav rezultat. Napisanu funkciju demonstrirajte u isječku koji neki string unesen sa tastature transformira tako da se svaki znak koji nije razmak mijenja znakom koji mu prethodi po ASCII poretку (npr. "B" postaje "A", itd.) i ispisuje tako transformirani string. Prikažite dva rješenja, jedno koje koristi imenovanu funkciju kao parametar, i drugo koje koristi lambda funkciju kao parametar.

Zadatak 5 (2,5 poena)

Napišite generičku funkciju "KopirajBezDuplikata" sa tri parametra "p1", "p2" i "p3". Funkcija prepisuje elemente bloka omeđenog sa "p1" i "p2" ("p1" pokazuje na početak a "p2" tačno iza kraja bloka) u blok na čiji početak pokazuje "p3", ali izostavljajući elemente koji su jednaki nekom elementu koji se već ranije pojavljivao u bloku (tj. izostavljajući duplikate). Funkcija treba da bude zasnovana na potpunoj dedukciji, tako da kao parametre prihvata i pokazivače i iteratore. Pored toga, funkcija bi trebala da se zasniva samo na osnovnom skupu operacija koje podržavaju svi tipovi iteratora (tako da treba da radi sa iteratorima svih vrsta, ne samo iteratorima sa direktnim pristupom). Parametri "p1" i "p2" moraju biti istog tipa, dok se tip parametra "p3" može razlikovati od njih. Kao rezultat, funkcija treba da vrati pokazivač/iterator koji pokazuje tačno iza kraja određiđenog bloka (nakon obavljenog prepisivanja). Napisanu funkciju demonstrirajte u isječku programa koji će neki string unesen sa tastature prepisati u drugi string izostavljajući duplikate prisutne u prvom stringu i na kraju ispisati novodobijeni string (korištenje umetača je dozvoljeno i poželjno).

Zadatak 6 (3 poena)

Napišite funkciju sa jednim parametrom "n" koja vrši dinamičku alokaciju grbave matrice čiji prvi red ima 1 element, drugi 2 elementa, i tako dalje do posljednjeg reda koji će imati "n" redova i popunjava tako formirane redove elementima Pascalovog trougla (trokuta). Dakle, prvi red treba sadržavati element 1, drugi red elemente 1 i 1, treći red elemente 1, 2 i 1, četvrti red elemente 1, 3, 3 i 1, peti red elemente 1, 4, 6, 4 i 1, itd. (prvi i posljednji element u svakom redu jednaki su 1, a svaki ostali element jednak je zbiru elementa koji se nalazi tačno iznad njega i njegovog susjeda sa lijeve strane). Za alokaciju treba koristiti postupak *kontinualne alokacije*. Funkcija kao rezultat treba da vrati dvostruki pokazivač preko kojeg se može izvršiti pristup elementima alocirane matrice. U slučaju da se kao parametar zada negativan broj ili nula, funkcija treba da baci tekst "Parametar mora biti pozitivan" kao izuzetak, dok u slučaju da alokacija ne uspije, treba baciti izuzetak koji se sastoji od teksta "Alokacija nije uspjela". Pri tome, ni u kom slučaju ne smije doći do curenja memorije. Napisanu funkciju demonstrirajte u isječku koji koristi napisanu funkciju da kreira dinamičku grbavu matricu koja sadrži Pascalov trougao (trokut) sa "n" redova, pri čemu se "n" unosi sa tastature i ispisuje tako kreiranu grbavu matricu na ekran. Porebno je predvidjeti i hvatanje eventualno bačenih izuzetaka, kao i oslobađanje prostora koji je matrica zauzimala nakon završetka rada.

Zadatak 7 (2 poena)

Razlika između najveće i najmanje cifre nekog broja naziva se njegov *raspon* (npr. broj 6735 ima raspon 4, jer je $7-3=4$). Napišite isječak programa koji će jedan fiksni deo od 8 petocifrenih brojeva (brojeve definirajte po svojoj volji) sortirati tako da brojevi sa manjim rasponom dođu ispred brojeva sa većim rasponom i ispisati tako sortirani deo. U slučaju da dva broja imaju isti raspon, broj koji je veći treba doći prije broja koji je manji. Za sortiranje koristiti funkciju "sort" iz biblioteke "algorithm". Poželjno bi bilo da kao funkciju kriterija upotrijebite lambda funkciju, ali ukoliko ne znate to uraditi, prihvata se i rješenje sa imenovanom funkcijom kriterija.