

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

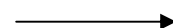
```
#include <iostream>
#include <complex>
using namespace std;
void f(int *x, int *y, char z) { cout << *x++ << z << *--y << " "; }
int g(double x, int y = 2) { return x * y + 1; }
double g(double x, double y, int z = 1) { return x + y * z; }
int &h(int f(double, int), int &x, int &y, int &z) {
    cout.width(7);
    cout << endl << x << y << z << endl;
    x = y + z; z = x + y;
    cout << x << y << z << endl;
    z = f(x, y);
    return x;
}
void t(complex<double> x, int *y) { cout << double(*y) + x << endl; }
int main() {
    enum x {y, z = 4, u, v = 9, w};
    int a[] = {6, 7, 4, 2, 8, 5, 9, 1, 4, 8};
    int *p(&a[4]), *q(p - 1), *r(a + 7);
    cout << y << z << u << v << w << endl;
    t(4, new int(1)); t(complex<double>(4, 4), new int(1));
    cout << g(5,8) << " " << g(5.8) << " " << g(5,8.) << " "
        << g(8,5,5.8) << " " << g(8.5,5.8) << " " << g(8,5.5,8) << endl;
    while(r >= q) f(q, r, '-');
    ++h(g, a[7], *a, p[-2]);
    cout << a[7] << " " << *a << " " << p[-2] << endl;
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja za vektor cijelih brojeva koji joj se proslijeđuje kao parametar vraća kao rezultat vektor sa istim brojem elemenata, ali čiji su elementi u odnosu na ulazni vektor transformirani na način da im je svaka parna cifra uvećana za 1, a svaka neparna cifra umanjena za 1. Naprimjer, ukoliko je neki element ulaznog vektora glasio 354126079, odgovarajući element u vektoru koji predstavlja rezultat treba da glasi 245037168. Slično, od broja -354126079 nastao bi broj -245037168. Napisanu funkciju demonstrirajte u isječku programa koji prikazuje kako se koristi napisana funkcija.

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. 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)

ROT13 je primitivni sistem šifriranja koji se često koristi na web stranicama kada se neki tekst ne želi da učini čitljivim na prvi pogled (npr. odgovor na neko pitanje koje je napisano neposredno ispod), ali da ga je po potrebi ipak moguće lako dešifrirati. Prema ovom sistemu, slova engleskog alfabeta iz opsega od A do M zamjenjuju se sa odgovarajućim slovima iz opsega od N do Z (tj. A se zamjenjuje sa N, B sa O, C sa P itd.) i obrnuto (dakle, N se zamjenjuje sa A itd.). Pri tome velika slova ostaju velika, mala slova ostaju mala, dok znakovi koji nisu slova (cifre, znaci interpunkcije, itd.) ostaju kakvi jesu. Interesatno je da se kod ovog sistema dešifriranje vrši istim postupkom kao i šifriranje (što je očigledno iz opisa postupka). Napišite funkciju "ROT13" koja kao parametar prima neki tekst (tipa "string") a koja transformira taj tekst prema ROT13 postupku. Funkcija ne vraća nikakav rezultat, već samo utiče na izmjenu stringa koji joj je proslijeđen kao parametar. Funkcija ne smije koristiti nikakve pomoćne stringove, nizove, vektore itd. Napišite i kratki isječak programa koji za neki tekst unesen sa tastature ispisuje njegovu transformaciju po ROT13 sistemu.

Zadatak 5 (3 poena)

Napišite generičku funkciju koja obavlja "kompaktifikaciju" dinamički alocirane matrice na način koji će biti opisan. Funkcija prima 3 parametra nazvana "a", "m" i "n". "a" je dvojni pokazivač preko kojeg se pristupa elementima neke dinamički alocirane dvodimenzionalne matrice za koju se pretpostavlja da je alocirana postupkom *fragmentirane alokacije* (tip elemenata matrice nije unaprijed poznat), dok su "m" i "n" njeni brojevi redova odnosno kolona respektivno. Funkcija treba da dinamički alocira novu matricu istog formata kao i početna matrica, ali ovaj put postupkom *kontinualne alokacije*, zatim da prepíše elemente početne matrice u novostvorenu matricu, oslobodi memoriju koju je zauzimala početna matrica i na kraju, da preusmjeri pokazivač "a" da pokazuje na novokreiranu matricu. U slučaju da nova alokacija ne uspije, funkcija treba da baci izuzetak, pri čemu tada izvorna matrica ne smije biti obrisana, niti pokazivač "a" smije biti izmijenjen. Vodite računa da funkcija ne smije stvoriti curenje memorije ni pod kojim uvjetima.

Zadatak 6 (2,5 poena)

Generička funkcija "max_element" iz biblioteke "algorithm" obično se koristi sa dva parametra, ali se može koristiti i sa tri parametra $p1$, $p2$ i f . Parametri $p1$ i $p2$ kao što je uobičajeno omeđuju blok koji se pretražuje ($p1$ pokazuje na početak a $p2$ pokazuje tačno iza kraja bloka), dok je f funkcija kriterija kojom se pobliže može specificirati šta znači "najveći". Recimo, moguće je specificirati da je jedan broj "veći" od drugog ukoliko ima veću sumu cifara. Uz takav kriterij, broj 39 je recimo "veći" od broja 124, tako da bi funkcija "max_element" uz takav kriterij zapravo pronašla broj sa najvećom sumom cifara. Funkcija kriterija radi na sljedeći način. Prilikom poređenja, umjesto prostog testiranja oblika $y > x$, ova varijanta funkcije "max_element" poziva $f(x, y)$ i smatra da je y veći od x ako i samo ako poziv $f(x, y)$ daje "tačno" kao rezultat. Funkcija kao rezultat vraća pokazivač (ili iterator, zavisno već od toga šta su $p1$ i $p2$) na mjesto u bloku gdje se nalazi "najveći" element (uz zadani kriterij). Napišite kako bi mogla izgledati implementacija ovakve funkcije "max_element" uz pretpostavku da se koristi potpuna dedukcija. Funkcija se mora zasnivati na pokazivačkoj/iteratorskoj aritmetici (tj. nije dozvoljeno indeksirati pokazivač, ili koristiti trivijalnu simulaciju indeksiranja oblika " $*(p+i)$ "). Napišite i kratki isječak programa koji će sa tastature unijeti 10 stringova, smjestiti ih u dek i na kraju ispisati najduži uneseni string. Za pronalaženje najdužeg stringa, iskoristite napisanu funkciju uz odgovarajuću funkciju kriterija.

Zadatak 7 (2 poena)

Pretpostavimo da je potrebno sortirati niz kompleksnih brojeva uz pomoć funkcije "sort" u takav poredak da brojevi sa manjim realnim dijelom dođu ispred brojeva sa većim realnim dijelom, ali da među brojevima sa jednakim realnim dijelom brojevi sa manjim imaginarnim dijelom dođu ispred brojeva sa većim imaginarnim dijelom. Napišite funkciju kriterija koja omogućava da funkcija "sort" obavi ovakvo sortiranje. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi 10 kompleksnih 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.

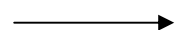
```
#include <iostream>
#include <complex>
using namespace std;
void f(int *x, int *y, char z) { cout << *x++ << z << *--y << " "; }
int g(double x, int y = 3) { return x * y - 1; }
double g(double x, double y, int z = 2) { return x + y * z; }
int &h(int f(double, int), int &x, int &y, int &z) {
    cout.width(5);
    cout << endl << x << y << z << endl;
    x = y + z; z = x + y;
    cout << x << y << z << endl;
    z = f(x, y);
    return x;
}
void t(complex<double> x, int *y) { cout << double(*y) - x << endl; }
int main() {
    enum x {y, z = 3, u, v = 8, w};
    int a[] = {3, 5, 2, 8, 7, 1, 3, 9, 0, 7};
    int *p(&a[3]), *q(p - 1), *r(a + 7);
    cout << y << z << u << v << w << endl;
    t(6, new int(2)); t(complex<double>(6, 6), new int(2));
    cout << g(4,6) << " " << g(4.6) << " " << g(4,6.) << " "
        << g(6,4,4.6) << " " << g(6.4,4.6) << " " << g(6,4.4,6) << endl;
    while(r >= q) f(q, r, '-');
    ++h(g, a[6], *a, p[-2]);
    cout << a[6] << " " << *a << " " << p[-2] << endl;
    return 0;
}
```

Zadatak 2 (2,5 poena)

Napišite funkciju koja za vektor cijelih brojeva koji joj se prosljeđuje kao parametar vraća kao rezultat vektor sa istim brojem elemenata, ali čiji su elementi u odnosu na ulazni vektor transformirani na način da im je svaka cifra manja od 5 uvećana za 1, a svaka cifra veća ili jednaka od 5 umanjena za 1. Naprimjer, ukoliko je neki element ulaznog vektora glasio 3541260, odgovarajući element u vektoru koji predstavlja rezultat treba da glasi 4452351. Slično, od broja -3541260 nastao bi broj -4452351. Napisanu funkciju demonstrirajte u isječku programa koji prikazuje kako se koristi napisana funkcija.

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. 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)

ROT13 je primitivni sistem šifriranja koji se često koristi na web stranicama kada se neki tekst ne želi da učini čitljivim na prvi pogled (npr. odgovor na neko pitanje koje je napisano neposredno ispod), ali da ga je po potrebi ipak moguće lako dešifrirati. Prema ovom sistemu, slova engleskog alfabeta iz opsega od A do M zamjenjuju se sa odgovarajućim slovima iz opsega od N do Z (tj. A se zamjenjuje sa N, B sa O, C sa P itd.) i obrnuto (dakle, N se zamjenjuje sa A itd.). Pri tome velika slova ostaju velika, mala slova ostaju mala, dok znakovi koji nisu slova (cifre, znaci interpunkcije, itd.) ostaju kakvi jesu. Interesatno je da se kod ovog sistema dešifriranje vrši istim postupkom kao i šifriranje (što je očigledno iz opisa postupka). Napišite funkciju "ROT13" koja kao parametar prima neki tekst (tipa "string") a koja kao rezultat daje taj tekst transformiran prema ROT13 postupku (parametar koji se prosljeđuje funkciji ostaje neizmijenjen). Funkcija ne smije koristiti nikakve pomoćne stringove, nizove, vektore itd. Napišite i kratki isječak programa koji za neki tekst unesen sa tastature ispisuje njegovu transformaciju po ROT13 sistemu.

Zadatak 5 (3 poena)

Napišite generičku funkciju koja obavlja "kompaktifikaciju" dinamički alocirane matrice na način koji će biti opisan. Funkcija prima 3 parametra nazvana "a", "m" i "n". "a" je dvojni pokazivač preko kojeg se pristupa elementima neke dinamički alocirane dvodimenzionalne matrice za koju se pretpostavlja da je alocirana postupkom *fragmentirane alokacije* (tip elemenata matrice nije unaprijed poznat), dok su "m" i "n" njeni brojevi redova odnosno kolona respektivno. Funkcija treba da dinamički alocira novu matricu istog formata kao i početna matrica, ali ovaj put postupkom *kontinualne alokacije*, zatim da prepíše elemente početne matrice u novostvorenu matricu, oslobodi memoriju koju je zauzimala početna matrica i na kraju, da preusmjeri pokazivač "a" da pokazuje na novokreiranu matricu. U slučaju da nova alokacija ne uspije, funkcija treba da baci izuzetak, pri čemu tada izvorna matrica ne smije biti obrisana, niti pokazivač "a" smije biti izmijenjen. Vodite računa da funkcija ne smije stvoriti curenje memorije ni pod kojim uvjetima.

Zadatak 6 (2,5 poena)

Generička funkcija "min_element" iz biblioteke "algorithm" obično se koristi sa dva parametra, ali se može koristiti i sa tri parametra $p1$, $p2$ i f . Parametri $p1$ i $p2$ kao što je uobičajeno omeđuju blok koji se pretražuje ($p1$ pokazuje na početak a $p2$ pokazuje tačno iza kraja bloka), dok je f funkcija kriterija kojom se pobliže može specificirati šta znači "najmanji". Recimo, moguće je specificirati da je jedan broj "manji" od drugog ukoliko ima manju sumu cifara. Uz takav kriterij, broj 124 je recimo "manji" od broja 39, tako da bi funkcija "min_element" uz takav kriterij zapravo pronašla broj sa najmanjom sumom cifara. Funkcija kriterija radi na sljedeći način. Prilikom poređenja, umjesto prostog testiranja oblika $x < y$, ova varijanta funkcije "min_element" poziva $f(x, y)$ i smatra da je x manji od y ako i samo ako poziv $f(x, y)$ daje "tačno" kao rezultat. Funkcija kao rezultat vraća pokazivač (ili iterator, zavisno već od toga šta su $p1$ i $p2$) na mjesto u bloku gdje se nalazi "najmanji" element (uz zadani kriterij). Napišite kako bi mogla izgledati implementacija ovakve funkcije "min_element" uz pretpostavku da se koristi potpuna dedukcija. Funkcija se mora zasnivati na pokazivačkoj/iteratorskoj aritmetici (tj. nije dozvoljeno indeksirati pokazivač, ili koristiti trivijalnu simulaciju indeksiranja oblika " $*(p+i)$ "). Napišite i kratki isječak programa koji će sa tastature unijeti 10 stringova, smjestiti ih u dek i na kraju ispisati najkraći uneseni string. Za pronalaženje najkraćeg stringa, iskoristite napisanu funkciju uz odgovarajuću funkciju kriterija.

Zadatak 7 (2 poena)

Pretpostavimo da je potrebno sortirati niz kompleksnih brojeva uz pomoć funkcije "sort" u takav poredak da brojevi sa većim realnim dijelom dođu ispred brojeva sa manjim realnim dijelom, ali da među brojevima sa jednakim realnim dijelom brojevi sa većim imaginarnim dijelom dođu ispred brojeva sa manjim imaginarnim dijelom. Napišite funkciju kriterija koja omogućava da funkcija "sort" obavi ovakvo sortiranje. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi 10 kompleksnih brojeva u neki dek, nakon čega se vrši njihovo sortiranje u skladu sa opisanim kriterijem te ispis tako sortiranog deka na ekran.