Zadaca – Programiranje

# Zadatak

#include <iostream>

#include <stdexcept>

#include <vector>

#include <iomanip>

using namespace std;

typedef std::vector<std::vector<double>> Matrica;

Matrica KreirajMatricu(int br\_redova, int br\_kolona) {

return Matrica(br\_redova, std::vector<double>(br\_kolona));

}

int BrojRedova(Matrica m) {

return m.size();

}

int BrojKolona(Matrica m) {

if (m.size() != 0) return m[0].size();

return 0;

}

Matrica UnesiMatricu(int br\_redova, int br\_kolona) {

auto m = KreirajMatricu(br\_redova, br\_kolona);

for (int i = 0; i < br\_redova; i++)

for (int j = 0; j < br\_kolona; j++) {

std::cout << "Element (" << i + 1 << "," << j + 1 << "): ";

std::cin >> m[i][j];

}

return m;

}

void IspisiMatricu(Matrica m, int sirina) {

for (int i = 0; i < BrojRedova(m); i++) {

for (int j = 0; j < BrojKolona(m); j++)

std::cout << std::setw(sirina) << m[i][j];

std::cout << std::endl;

}

cout << endl;

}

Matrica SaberiMatrice(Matrica m1, Matrica m2) {

auto m3 = KreirajMatricu(BrojRedova(m1), BrojKolona(m1));

if (BrojRedova(m1) != BrojRedova(m2) && BrojKolona(m1) != BrojKolona(m2)) throw domain\_error("Matrice nisu kompatibilne za sabiranje. Ponovite unos.");

cout << "Zbir vasih matrica izgleda ovako: " << endl;

for (int i = 0; i < BrojRedova(m1); i++)

for (int j = 0; j < BrojKolona(m1); j++) m3[i][j] = m1[i][j] + m2[i][j];

return m3;

}

Matrica PomnoziMatricuSkalarom(Matrica m, int skalar, int broj\_matrice) {

auto m1 = KreirajMatricu(BrojRedova(m), BrojKolona(m));

cout << "Proizvod skalara i vase " << broj\_matrice << ". matrice izgleda ovako: " << endl;

for (int i = 0; i < BrojRedova(m); i++)

for (int j = 0; j < BrojKolona(m); j++)

m1[i][j] = skalar \* m[i][j];

return m1;

}

Matrica PomnoziMatrice(Matrica m1, Matrica m2) {

auto m3 = KreirajMatricu(BrojRedova(m1), BrojKolona(m2));

if (BrojRedova(m1) != BrojRedova(m2) || BrojKolona(m1) != BrojKolona(m2)) throw domain\_error("Matrice nisu kompatibilne za mnozenje. Ponovite unos.");

cout << "Proizvod vasih matrica izgleda ovako: " << endl;

for (int i = 0; i < BrojRedova(m1); i++)

for (int j = 0; j < BrojKolona(m2); j++) {

double suma = 0;

for (int k = 0; k < BrojRedova(m2); k++) suma += m1[i][k] \* m2[k][j];

m3[i][j] = suma;

}

return m3;

}

int main()

{

int r1, k1, r2, k2, skalar;

cout << "Unesite broj redova prve matrice: " << endl;

cin >> r1;

cout << "Unesite broj kolona prve matrice: " << endl;

cin >> k1;

Matrica a = UnesiMatricu(r1, k1);

cout << endl << "Vasa matrica izgleda ovako: " << endl;

IspisiMatricu(a, 3);

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

cout << "Unesite broj redova druge matrice: " << endl;

cin >> r2;

cout << "Unesite broj kolona druge matrice: " << endl;

cin >> k2;

Matrica b = UnesiMatricu(r2, k2);

cout << endl << "Vasa matrica izgleda ovako: " << endl;

IspisiMatricu(b, 3);

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

cout << "Unesite skalar kojim cete mnoziti obje matrice: " << endl;

cin >> skalar;

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

try {

IspisiMatricu(SaberiMatrice(a, b), 3);

}

catch (domain\_error izuzetak) {

cout << izuzetak.what() << endl;

}

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

IspisiMatricu(PomnoziMatricuSkalarom(a, skalar, 1), 3);

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

IspisiMatricu(PomnoziMatricuSkalarom(b, skalar, 2), 3);

cout << endl << "------------------------------------------------------------------------------------------------" << endl << endl << endl;

try {

IspisiMatricu(PomnoziMatrice(a, b), 3);

}

catch (domain\_error izuzetak) {

cout << izuzetak.what() << endl;

}

return 0;

}

# Zadatak

#include <iostream>

#include <vector>

#include <string>

#include <cwctype>

#include <stdexcept>

#include <algorithm>

using namespace std;

vector<string> RazvrstajRijeciPoDuzini(string recenica)

{

vector<string> v;

string rijec;

char slovo;

for (int i = 0; i < recenica.size(); i++)

{

slovo = recenica[i];

if (slovo == ' ' && rijec.size() != 0)

{

v.push\_back(rijec);

rijec.clear();

}

else if (iswalnum(slovo))

rijec += toupper(slovo);

}

v.push\_back(rijec);

for (int i=0; i<v.size(); i++)

if (v[i].size() > 99)

throw range\_error("Jedna od rijeci je preduga (preko 99 slova). Ponovite unos.");

vector<string> duzina(99);

for (int i = 0; i < duzina.size(); i++) {

for (int j = 0; j < v.size(); j++) {

if (v[j].size() == i+1)

duzina[i] += (v[j] + " ");

}

}

return duzina;

}

int main()

{

string recenica;

cout << "Unesite recenicu: ";

getline(cin, recenica);

int d = 1;

try {

vector<string> v = RazvrstajRijeciPoDuzini(recenica);

for (string i : v) {

if(i != "")

cout << "Duzina " << d << ": " << i << endl;

d++;

}

}

catch(range\_error izuzetak) {

cout << izuzetak.what() << endl;

}

return 0;

}

# Zadatak

#include <iostream>

#include <stdexcept>

#include <new>

#include <cmath>

using namespace std;

long double\* DinamickiNiz(int n) {

long double\* niz = new long double[n];

if (n == 0) throw range\_error("Broj elemenata mora biti pozitivan");

for (int i = 0; i < n; i++) {

niz[i] = pow(2, i);

}

return niz;

}

int main() {

int n;

cout << "Unesite broj elemenata niza: ";

cin >> n;

long double\* ptr = DinamickiNiz(n);

try {

cout << "Vas niz je: " << endl;

for (int i = 0; i < n; i++)

cout << ptr[i] << endl;

cout << endl;

delete[] ptr;

}

catch(range\_error izuzetak) {

cout << izuzetak.what() << endl;

}

catch(bad\_alloc izuzetak) {

cout << "Alokacija nije uspjela" << endl;

}

return 0;

}