

O'ZBEKISTON RESPUBLIKASI

OLIY TA'LIM, FAN VA INOVATSIYALAR

VAZIRLIGI

MIRZO ULUGBEK NOMIDAGI

O ZBEKISTON MILLIY UNIVERSITETI

AMALIY MATEMATIKA VA INTELLEKTUAL

TEXNALOGIYALARI FAKULTETI

AMALIY MATEMATIKA

YO’NALISHI 1-BOSQICH TALABASI

ABDURAIMOV ABDULLOHNING

# MALAKAVIY AMALIYOT HISOBOTI

**Tekshirdi: Jaxongir R.**

**Topshirdi: Abduraimov A.**

* 1 Пример

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

public class Program

{

public static double TriangleArea(double x1, double y1, double x2, double y2, double x3, double y3)

{

// Uchburchak yuzasini hisoblash

double side1 = Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2));

double side2 = Math.Sqrt(Math.Pow(x3 - x2, 2) + Math.Pow(y3 - y2, 2));

double side3 = Math.Sqrt(Math.Pow(x1 - x3, 2) + Math.Pow(y1 - y3, 2));

// Uchburchak to'g'ri tuzilganligini tekshirish

if (side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1)

{

double s = (side1 + side2 + side3) / 2;

double area = Math.Sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

return area;

}

else

{

throw new ArgumentException("Uchburchak Xato tuzilgan");

}

}

public static string CompareTriangleAreas(double x1, double y1, double x2, double y2, double x3, double y3,

double x1\_, double y1\_, double x2\_, double y2\_, double x3\_, double y3\_)

{

// Birinchi uchburchakning yuzasini hisoblash

double area1 = TriangleArea(x1, y1, x2, y2, x3, y3);

// Ikkinchi uchburchakning yuzasini hisoblash

double area2 = TriangleArea(x1\_, y1\_, x2\_, y2\_, x3\_, y3\_);

// Uchburchaklarning yuzalari taqqoslash

if (area1 > area2)

{

return "Birinchi uchburchak katta yuzaga ega";

}

else if (area1 < area2)

{

return "Ikkinchi uchburchak katta yuzaga ega";

}

else

{

return "Uchburchaklar teng yuzaga ega";

}

}

public static void Main()

{

double x1, y1, x2, y2, x3, y3;

double x1\_, y1\_, x2\_, y2\_, x3\_, y3\_;

// Birinchi uchburchakning koordinatalarini kiritish

Console.WriteLine("Birinchi uchburchakning x1 qiymati: ");

x1 = double.Parse(Console.ReadLine());

Console.WriteLine("Birinchi uchburchakning y1 qiymati: ");

y1 = double.Parse(Console.ReadLine());

Console.WriteLine("Birinchi uchburchakning x2 qiymati: ");

x2 = double.Parse(Console.ReadLine());

Console.WriteLine("Birinchi uchburchakning y2 qiymati: ");

y2 = double.Parse(Console.ReadLine());

Console.WriteLine("Birinchi uchburchakning x3 qiymati: ");

x3 = double.Parse(Console.ReadLine());

Console.WriteLine("Birinchi uchburchakning y3 qiymati: ");

y3 = double.Parse(Console.ReadLine());

// Ikkinchi uchburchakning koordinatalarini kiritish

Console.WriteLine("Ikkinchi uchburchakning x1 qiymati: ");

x1\_ = double.Parse(Console.ReadLine());

Console.WriteLine("Ikkinchi uchburchakning y1 qiymati: ");

y1\_ = double.Parse(Console.ReadLine());

Console.WriteLine("Ikkinchi uchburchakning x2 qiymati: ");

x2\_ = double.Parse(Console.ReadLine());

Console.WriteLine("Ikkinchi uchburchakning y2 qiymati: ");

y2\_ = double.Parse(Console.ReadLine());

Console.WriteLine("Ikkinchi uchburchakning x3 qiymati: ");

x3\_ = double.Parse(Console.ReadLine());

Console.WriteLine("Ikkinchi uchburchakning y3 qiymati: ");

y3\_ = double.Parse(Console.ReadLine());

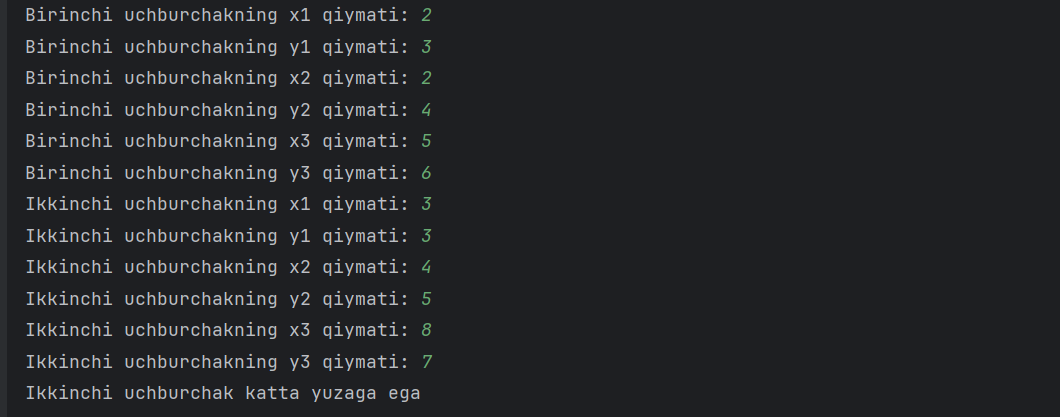
// Uchburchaklarning yuzasini taqqoslash va natijani chop etish

string result = CompareTriangleAreas(x1, y1, x2, y2, x3, y3, x1\_, y1\_, x2\_, y2\_, x3\_, y3\_);

Console.WriteLine(result);

}

}



* 2 пример

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

public class Program

{

public static double[,] MatrixPower(double[,] matrix, int power)

{

double[,] result = new double[matrix.GetLength(0), matrix.GetLength(1)];

if (power == 0)

{

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

result[i, j] = (i == j) ? 1 : 0;

}

}

}

else if (power == 1)

{

Array.Copy(matrix, result, matrix.Length);

}

else

{

result = MatrixMultiply(matrix, matrix);

for (int p = 2; p < power; p++)

{

result = MatrixMultiply(result, matrix);

}

}

return result;

}

public static double[,] MatrixMultiply(double[,] matrix1, double[,] matrix2)

{

int rows1 = matrix1.GetLength(0);

int cols1 = matrix1.GetLength(1);

int rows2 = matrix2.GetLength(0);

int cols2 = matrix2.GetLength(1);

if (cols1 != rows2)

{

throw new ArgumentException("Matritsalar hajmlari mos emas.");

}

double[,] result = new double[rows1, cols2];

for (int i = 0; i < rows1; i++)

{

for (int j = 0; j < cols2; j++)

{

double sum = 0;

for (int k = 0; k < cols1; k++)

{

sum += matrix1[i, k] \* matrix2[k, j];

}

result[i, j] = sum;

}

}

return result;

}

public static void Main()

{

int n;

Console.WriteLine("Matritsaning o'lchamini kiriting: ");

n = int.Parse(Console.ReadLine());

// Matritsa elementlarini olish

double[,] matrix = new double[n, n];

Console.WriteLine("Matritsaning elementlarini kiriting:");

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

{

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

{

Console.Write($"Matrix[{i}][{j}]: ");

matrix[i, j] = double.Parse(Console.ReadLine());

}

}

int power;

Console.WriteLine("Ildiz darajasini kiriting: ");

power = int.Parse(Console.ReadLine());

double[,] result = MatrixPower(matrix, power);

Console.WriteLine($"{power}-darajali matritsa:");

for (int i = 0; i < result.GetLength(0); i++)

{

for (int j = 0; j < result.GetLength(1); j++)

{

Console.Write(result[i, j] + " ");

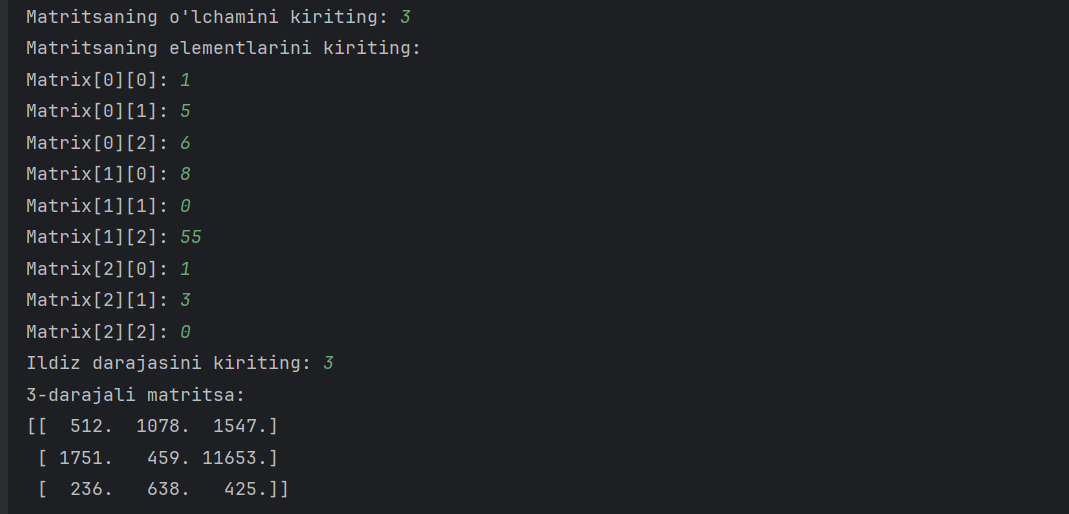
}

Console.WriteLine();

}

}

}



* 3 Пример

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication32

public enum Hafta\_kuni

{

yaksh = 0,

dush = 1,

sesh = 2,

chor = 3,

paysh = 4,

juma = 5,

shanba = 6

}

public class Program

{

public static string[] Yil = new string[365];

static Program()

{

for (int i = 0; i < 365; i++)

{

Yil[i] = ((Hafta\_kuni)((i + 1) % 7)).ToString();

}

}

public static string TogriKuniTop(int k)

{

return Yil[k - 1].Substring(0, 1).ToUpper() + Yil[k - 1].Substring(1);

}

public static void Main(string[] args)

{

while (true)

{

Console.Write("k-kunini kiriting: [365] => k");

int k = int.Parse(Console.ReadLine());

if (k <= 365 && k > 0)

{

string k\_kuni = TogriKuniTop(k);

Console.WriteLine($"{k}-kuni: {k\_kuni}");

break;

}

else

{

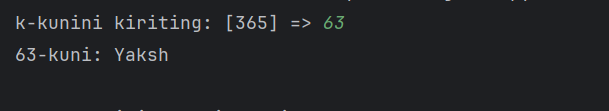
Console.WriteLine("k-kunini xato kiritdingiz");

}

}

}

}



* 4 Пример

using System;

using System.Linq;

public class Program

{

public static void Main(string[] args)

{

double[] x = { 1.0, 2.0, 3.0, 4.0, 5.0 };

// a) Har bir elementga 1 qo'shilgan vektor

double[] a\_vector = x.Select(elem => elem + 1).ToArray();

// b) Vektor elementlari teskari tartibda joylashtirilgan

double[] b\_vector = x.Reverse().ToArray();

// c) Vektor elementlari ko'paytirilgan

double c\_vector = x.Aggregate(1.0, (acc, elem) => acc \* elem);

// d) Vektor elementlari vergul orqali ajratilgan

string d\_vector = string.Join(", ", x);

// e) Vektor elementlari chakana ravishda chapga o'tkazilgan

Console.Write("p taga chapga surilsin: ");

int p = int.Parse(Console.ReadLine());

double[] e\_vector = x.Skip(p).Concat(x.Take(p)).ToArray();

Console.WriteLine("a) Har bir elementga 1 qo'shilgan vektor: [" + string.Join(", ", a\_vector) + "]");

Console.WriteLine("b) Vektor elementlari teskari tartibda joylashtirilgan: [" + string.Join(", ", b\_vector) + "]");

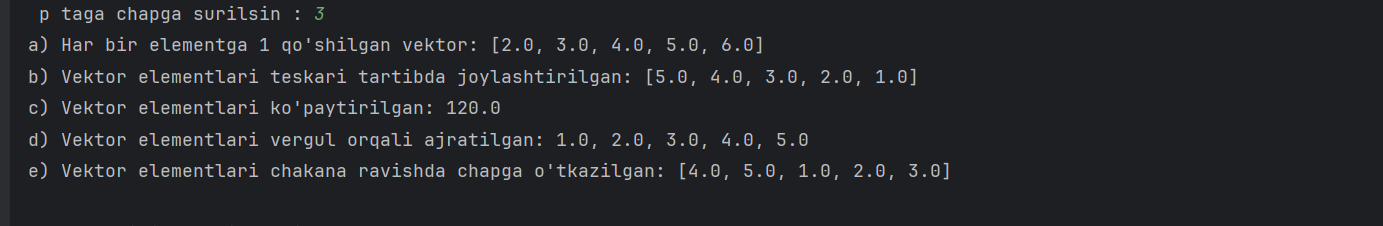
Console.WriteLine("c) Vektor elementlari ko'paytirilgan: " + c\_vector);

Console.WriteLine("d) Vektor elementlari vergul orqali ajratilgan: [" + d\_vector + "]");

Console.WriteLine("e) Vektor elementlari chakana ravishda chapga o'tkazilgan: [" + string.Join(", ", e\_vector) + "]");

}

}



* 5 Пример

using System;

using System.Linq;

public class Program

{

public static string OzgartirishA(string ketma\_ketlik)

{

string[] sozlar = ketma\_ketlik.Split();

string[] natija = new string[sozlar.Length];

for (int i = 0; i < sozlar.Length; i++)

{

string soz = sozlar[i];

char oxirgi\_harf = soz.Last();

int oxirgi\_harf\_soni = soz.Count(c => c == oxirgi\_harf);

if (oxirgi\_harf\_soni == 1)

{

natija[i] = soz;

}

else

{

string takrorlanmagan\_harflar = string.Join("", soz.Where(c => c != oxirgi\_harf).Distinct());

natija[i] = takrorlanmagan\_harflar + oxirgi\_harf;

}

}

return string.Join(" ", natija);

}

public static string OzgartirishB(string ketma\_ketlik)

{

string[] sozlar = ketma\_ketlik.Split();

string[] natija = new string[sozlar.Length];

for (int i = 0; i < sozlar.Length; i++)

{

natija[i] = new string(sozlar[i].Distinct().ToArray());

}

return string.Join(" ", natija);

}

public static string OzgartirishD(string ketma\_ketlik)

{

string[] sozlar = ketma\_ketlik.Split();

string[] natija = new string[sozlar.Length];

for (int i = 0; i < sozlar.Length; i++)

{

string soz = sozlar[i];

if (soz.Length % 2 != 0)

{

int orta\_index = soz.Length / 2;

natija[i] = soz.Substring(0, orta\_index) + soz.Substring(orta\_index + 1);

}

else

{

natija[i] = soz;

}

}

return string.Join(" ", natija);

}

public static void Main(string[] args)

{

string ketma\_ketlik = "assalomu alaykum aziz dindoshim.";

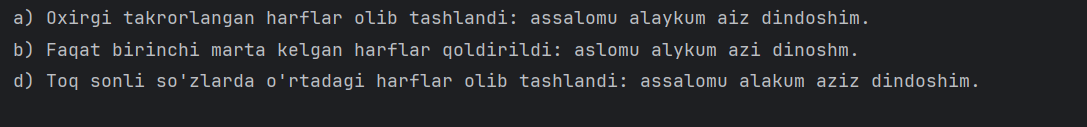
Console.WriteLine("a) Oxirgi takrorlangan harflar olib tashlandi: " + OzgartirishA(ketma\_ketlik));

Console.WriteLine("b) Faqat birinchi marta kelgan harflar qoldirildi: " + OzgartirishB(ketma\_ketlik));

Console.WriteLine("d) Toq sonli so'zlarda o'rtadagi harflar olib tashlandi: " + OzgartirishD(ketma\_ketlik));

}

}



* 6 Пример

using System;

public class Program

{

public static double Daraja(double x, int n)

{

if (n == 0)

{

return 1;

}

else if (n < 0)

{

return 1 / Daraja(x, -n);

}

else if (n % 2 == 0)

{

double temp = Daraja(x, n / 2);

return temp \* temp;

}

else

{

return x \* Daraja(x, n - 1);

}

}

public static void Main(string[] args)

{

Console.Write("sonni kiriting = ");

double x = double.Parse(Console.ReadLine());

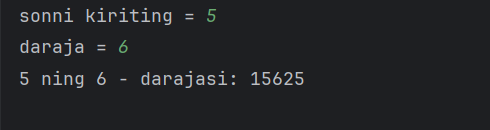
Console.Write("daraja = ");

int n = int.Parse(Console.ReadLine());

Console.WriteLine($"{x} ning {n} - darajasi: " + Daraja(x, n));

}

}



* 7 Пример

using System;

public class Kompleks

{

public double Haqiqiy { get; }

public double Imkoniy { get; }

public Kompleks(double haqiqiy, double imkoniy)

{

Haqiqiy = haqiqiy;

Imkoniy = imkoniy;

}

public static Kompleks Qoshish(Kompleks kompleks1, Kompleks kompleks2)

{

double haqiqiy\_javob = kompleks1.Haqiqiy + kompleks2.Haqiqiy;

double imkoniy\_javob = kompleks1.Imkoniy + kompleks2.Imkoniy;

return new Kompleks(haqiqiy\_javob, imkoniy\_javob);

}

public static Kompleks Ayirish(Kompleks kompleks1, Kompleks kompleks2)

{

double haqiqiy\_javob = kompleks1.Haqiqiy - kompleks2.Haqiqiy;

double imkoniy\_javob = kompleks1.Imkoniy - kompleks2.Imkoniy;

return new Kompleks(haqiqiy\_javob, imkoniy\_javob);

}

public static Kompleks Kopaytirish(Kompleks kompleks1, Kompleks kompleks2)

{

double haqiqiy\_javob = kompleks1.Haqiqiy \* kompleks2.Haqiqiy - kompleks1.Imkoniy \* kompleks2.Imkoniy;

double imkoniy\_javob = kompleks1.Haqiqiy \* kompleks2.Imkoniy + kompleks1.Imkoniy \* kompleks2.Haqiqiy;

return new Kompleks(haqiqiy\_javob, imkoniy\_javob);

}

public override string ToString()

{

return $"{Haqiqiy} + {Imkoniy}i";

}

public static void Main(string[] args)

{

Kompleks a = new Kompleks(3, 2);

Kompleks b = new Kompleks(1, 5);

// Qo'shish

Console.WriteLine("Qo'shish: " + Qoshish(a, b));

// Ayirish

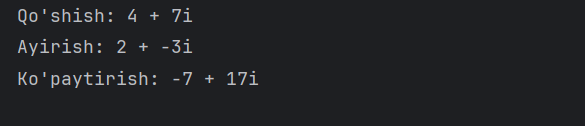
Console.WriteLine("Ayirish: " + Ayirish(a, b));

// Ko'paytirish

Console.WriteLine("Ko'paytirish: " + Kopaytirish(a, b));

}

}



* 8 Пример

using System;

class KOMPLEKS

{

public double Real { get; set; }

public double Imag { get; set; }

public KOMPLEKS(double real, double imag)

{

Real = real;

Imag = imag;

}

public override string ToString()

{

return Imag >= 0 ? $"{Real} + {Imag}i" : $"{Real} - {Math.Abs(Imag)}i";

}

public static KOMPLEKS operator +(KOMPLEKS a, KOMPLEKS b)

{

return new KOMPLEKS(a.Real + b.Real, a.Imag + b.Imag);

}

public static KOMPLEKS operator -(KOMPLEKS a, KOMPLEKS b)

{

return new KOMPLEKS(a.Real - b.Real, a.Imag - b.Imag);

}

public static KOMPLEKS operator \*(KOMPLEKS a, KOMPLEKS b)

{

double realPart = a.Real \* b.Real - a.Imag \* b.Imag;

double imagPart = a.Real \* b.Imag + a.Imag \* b.Real;

return new KOMPLEKS(realPart, imagPart);

}

}

class KOMP\_KV\_TENGLAMA : KOMPLEKS

{

public double B { get; set; }

public double C { get; set; }

public KOMP\_KV\_TENGLAMA(double a, double b, double c) : base(a, 0)

{

B = b;

C = c;

}

public double Discriminant()

{

return B \* B - 4 \* Real \* C;

}

public Tuple<KOMPLEKS, KOMPLEKS> Roots()

{

double disc = Discriminant();

if (disc >= 0)

{

double root1 = (-B + Math.Sqrt(disc)) / (2 \* Real);

double root2 = (-B - Math.Sqrt(disc)) / (2 \* Real);

return Tuple.Create(new KOMPLEKS(root1, 0), new KOMPLEKS(root2, 0));

}

else

{

double realPart = -B / (2 \* Real);

double imagPart = Math.Sqrt(Math.Abs(disc)) / (2 \* Real);

KOMPLEKS root1 = new KOMPLEKS(realPart, imagPart);

KOMPLEKS root2 = new KOMPLEKS(realPart, -imagPart);

return Tuple.Create(root1, root2);

}

}

}

class Program

{

static void Main(string[] args)

{

KOMPLEKS a = new KOMPLEKS(1, 2);

KOMPLEKS b = new KOMPLEKS(3, 4);

Console.WriteLine("a = " + a);

Console.WriteLine("b = " + b);

Console.WriteLine("a + b = " + (a + b));

Console.WriteLine("a - b = " + (a - b));

Console.WriteLine("a \* b = " + (a \* b));

KOMP\_KV\_TENGLAMA quadraticEq = new KOMP\_KV\_TENGLAMA(2, 8, 9);

Console.WriteLine("\nKvadrat tenglama: " + quadraticEq.Real + "x^2 + " + quadraticEq.B + "x + " + quadraticEq.C + " = 0");

Console.WriteLine("Discriminant: " + quadraticEq.Discriminant());

var roots = quadraticEq.Roots();

Console.WriteLine("Ildizlari: " + roots.Item1 + ", " + roots.Item2);

}

}

