Euklidov algoritam za nzm u programskom jeziku C

Kovač, Borna

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:979126

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

EUKLIDOV ALGORITAM ZA NAJMANJU ZAJEDNIČU MJERU U PROGRAMSKOM JEZIKU C

Završni rad

Borna Kovač

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 18.09.2017.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Borna Kovač
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	3621, 10.10.2016.
OIB studenta:	92353118521
Mentor:	Doc.dr.sc. Tomislav Rudec
Sumentor:	Doc.dr.sc. Alfonzo Baumgartner
Sumentor iz tvrtke:	
Naslov završnog rada:	Euklidov algoritam za nzm u programskom jeziku C
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	18.09.2017.
Datum potvrde ocjene Odbora:	27.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 26.09.2017.

Ime i prezime studenta:	Borna Kovač
Studij:	Preddiplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	3621, 10.10.2016.
Ephorus podudaranje [%]:	0 %

Ovom izjavom izjavljujem da je rad pod nazivom**: Euklidov algoritam za nzm u programskom jeziku C**

izrađen pod vodstvom mentora Doc.dr.sc. Tomislav Rudec

i sumentora Doc.dr.sc. Alfonzo Baumgartner

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. O EUKLIDU 2
3. MATEMATIČKA OBRADA EUKLIDOVOG ALGORITMA 3
3.1. Postupak računanja Euklidovog algoritma 3
3.2. Dokaz o valjanosti 4
3.3. Euklidska podjela 5
3.4. Primjer Euklidovog algoritma za cijele brojeve 6
3.5. Prošireni Euklidov algoritam 7
3.6. Primjer Euklidovog algoritma za polinome 8
4. PREGLED KORIŠTENIH ALATA ZA IZRADU PROGRAMA 10
4.1. Microsoft Visual Studio 2017 10
4.2. Windows Forme 10
4.3. Microsoft Visual C# 15
5. IZRADA PROGRAMA 16
5.1. Program za računanje cijelim brojevima 17
5.1.1. Unos cijelih brojeva 18
5.1.2. Računanje najveće zajedničke mjere 19
5.2. Program za računanje polinomima 21
5.2.1. Unos koeficijenata 21
5.2.2. Računanje najveće zajedničke mjere 21
6. ZAKLJUČAK 29
LITERATURA 30
SAŽETAK 31
ŽIVOTOPIS 33

1. UVOD

Euklidov algoritam je efektivna metoda za određivanje najveće zajedničke mjere dva broja, odnosno najvećeg prirodnog broja koji dijeli oba dva zadana broja bez ostavljanja ostatka. Euklidov algoritam ima mnogo teorijskih i praktičnih primjena. Koristi se za skraćivanje razlomaka i pri dijeljenju u modularnoj aritmetici. Također se koristi u računarstvu za kriptografske protokole koji služe da bi se osigurala internetska komunikacija. Može se koristiti i za rješavanje diofantskih jednadžbi.

U ovom završnom radu prikazana je matematička obrada primjene Euklidovog algoritma za određivanje najveće zajedničke mjere dva cijela broja i dva polinoma drugog stupnja. Na temelju toga izrađen je računalni program u programskom jeziku C# koji se sastoji od dva dijela, računanje Euklidovog algoritma za dva cijela broja i računanje Euklidovog algoritma za dva polinoma drugog stupnja. Program nakon unesenih parametara prikazuje postupak primjene Euklidovog algoritma i krajnji rezultat, koji je najveća zajednička mjera ta dva broja ili polinoma.

1.1. Zadatak završnog rada

U završnom radu potrebno je napisati program u programskom jeziku C koji će dati izvještaj o najmanjoj zajedničkoj mjeri za prethodno upisane brojeve ili polinome u priklano sučelje koristeći Euklidov algoritam.

2. O EUKLIDU

Euklid, također i zvan Euklid iz Aleksandrije da se razlikuje od Euklida iz Megare, je bio grčki matematičar. Rođen je oko 330. godine prije Krista u Grčkoj. Euklida su također često zvali i „otac geometrije“. Bio je aktivan u Aleksandriji prilikom vladavine Ptolemeja I. Autor je mnogih djela, a neka od njih nisu sačuvana. *Elementi* je njegovo najpoznatije djelo, napisano u 13 knjiga. *Elementi* je služilo kao matematički udžbenik, pogotovo geometrijski, sve to ranog 20. stoljeća. U *Elementi*, Euklid za zasnovao principe geometrije od malog seta aksioma, zvana Euklidova geometrija. Također je pisao i o teoriji brojeva, sferičnoj geometriji i algebri. Sačuvana su i njegova djela: *Data*, *Optika* i *Katopika*. Najpoznatiji je po Euklidovom algoritmu koji služi za pronalazak najvećeg zajedničkog djelitelja (mjere) dva broja.

3. MATEMATIČKA OBRADA EUKLIDOVOG ALGORITMA

Euklidov algoritam računa najveću zajedničku mjeru (NZM) dva prirodna broja *a* i *b*. Najveća zajednička mjera *m* je najveći prirodni broj koji je dijeli i *a* i *b* bez ostavljanja ostatka. Najveća zajednička mjera se najčešće označava sa *m(a, b)* ili jednostavnije kao (a, b).

Ako je *m(a, b)* = 1, onda su *a* i *b* relativno prosti brojevi. Ovo svojstvo ne govori da su *a* i *b* kao sami prosti brojevi. Na primjer, ni broj 9 ni 25 nisu prosti brojevi, jer oba dva broja imaju 2 glavna faktora: $9 = 3 \cdot 3$ i $25 = 5 \cdot 5$. Ali brojevi 9 i 25 su relativno prosti, jer nijedan drugi broj osim 1 nije djeljiv sa oba dva broja.[1]

3.1. Postupak računanja Euklidovog algoritma

Euklidov algoritam nastavlja se u nizu koraka tako da se izlaz svakog koraka koristi kao ulaz za sljedeći. Neka je *k* broj koji broji korake u algoritmu, počevši od nule. Što znači da je početni korak: $k = 0$, a sljedeći je $k = 1$ i tako dalje.

Svaki korak počinje sa dva ne negativna ostatka r_{k-1} i r_{k-2}. Pošto algoritam osigurava da se ostatak postupno smanjuje nakon svakog koraka, r_{k-1} je uvijek manji od r_{k-2}. Cilj k-tog koraka je naći kvocijent q_k i ostatak r_k tako da zadovoljavaju jednadžbu:

$$r_{k-2} = q_k r_{k-1} + r_k \qquad (3-1)$$

tako da je $r_k < r_{k-1}$. Drugim riječima, višekratnici manjeg broja r_{k-1} su umanjeni od većeg broja r_{k-2} sve dok ostatak r_k ne bude manji od r_{k-1}. [1]

U početkom koraku ($k = 0$), ostaci r_{-2} i r_{-1} su jednaki *a* i *b*, brojevima za koje se traži najveća zajednička mjera. U idućem koraku ($k = 1$), ostaci su jednaki *b* i ostatku r_0 početnog koraka, i tako dalje. Tako da se algoritam može napisati kao slijed jednadžbi:

$$a = q_0 b + r_0 r_k \qquad (3-2)$$

$$b = q_1 r_0 + r_1 r_k \qquad (3-3)$$

$$r_0 = q_2 r_1 + r_2 r_k \quad (3-4)$$

$$r_1 = q_3 r_2 + r_3 r_k \quad (3-5)$$

$$\vdots$$

Ako je a manji od b, u prvom koraku algoritma se samo zamijene brojevi. Na primjer, ako je $a < b$, početni kvocijent q_0 je jednak nuli, a ostatak r_0 je jednak a. Tako da je r_k manji od njegovog prethodnika r_{k-1} za $k \geq 0$.

Pošto se ostaci smanjuju nakon svakog koraka, ali ne mogu nikad biti negativni, ostatak r_k mora eventualno biti jednak nuli, pri čemu se algoritam zaustavlja. Posljednji ostatak r_{k-1} koji nije nula je najveća zajednička mjera brojeva a i b. Broj k ne može biti beskonačan zato što postoji konačan broj ne negativnih brojeva između početnog ostatka r_0 i nule.

3.2. Dokaz o valjanosti

Valjanost Euklidovog algoritma može se dokazati argumentom u dva koraka. U prvom koraku, konačni ne nulti ostatak r_{k-1} je prikazan kako bi podijelio i a i b. Pošto je to zajednički djelitelj, mora biti manji ili jednak najvećoj zajedničkoj mjeri m. U drugom koraku, prikazano je da bilo koji zajednički djelitelj brojeva a i b, uključujući i m, mora biti djeljiv sa r_{k-1}. Stoga, m mora biti manji ili jednak r_{k-1}. Ova dva zaključka su nedosljedna osim ako je $r_{k-1} = m$.

Da bi se pokazao prvi korak, odnosno da r_{k-1} dijeli i a i b (u prvom koraku), prvo treba primijetiti da r_{k-1} dijeli svojeg prethodnika r_{k-2}

$$r_{k-1} = q_k r_{k-1} \quad (3-6)$$

pošto je konačni ostatak r_k jednak nuli. r_{k-1} također i dijeli svog idućeg prethodnika r_{k-3}

$$r_{k-3} = q_{k-1} r_{k-2} + r_{k-1} \quad (3-7)$$

zato što istovremeno dijeli oba uvjeta sa desne strane jednadžbe. Razmatrajući redom sve ostale prethodnike, zaključuje se da ih r_{k-1} dijeli sve, uključujući a i b. Nijedan od prethodnik ostataka r_{k-2},

r_{k-3}, itd. dijeli istovremeno i *a* i *b*, pošto ostavljaju ostatke. Pošto je r_{k-1} zajednički djelitelj brojeva *a* i *b*, $r_{k-1} \leq m$.

U drugom koraku, bilo koji prirodan broj *c* koji dijeli i *a* i *b* (drugim riječima, zajednički djelitelj *a* i *b*) mora dijeliti i ostatak r_k. Po definiciji, *a* i *b* mogu biti napisani kao višekratnici broja *c*: $a = xc$ i $b = yc$, gdje su *x* i *y* prirodni brojevi. Stoga, *c* dijeli početni ostatak r_0, jer vrijedi:

$$r_0 = a - q_0 b = xc - q_0 yc = (x - q_0 y)c. \qquad (3-8)$$

Analogni argument pokazuje da *c* također dijeli i sljedeće ostatke r_1i r_2, itd. Dakle, najveća zajednička mjera *m* mora dijeliti r_{k-1}, gdje je $m \leq r_{k-1}$.[1] Kako je prvi dio argumenta pokazao suprotno ($r_{k-1} \leq m$), slijedi da je $m = r_{k-1}$. Zato je *m* najveća zajednička mjera svih sljedećih parova:

$$m = nzm(a,b) = nzm(b,r_0) = nzm(r_0,r_1) = \cdots = nzm(r_{k-2},r_{k-1}) = r_{k-1}. \qquad (3-9)$$

3.3. Euklidska podjela

U svakom koraku *k*, Euklidov algoritam računa kvocijent q_k i ostatak r_k od dva broja r_{k-1} i r_{k-2}:

$$r_{k-2} = q_k r_{k-1} + r_k \qquad (3-10)$$

pri čemu je r_k po apsolutnoj vrijednosti strogo manji od r_{k-1}. Teorem koji podrazumijeva definiciju euklidske podjele osigurava da takav kvocijent i ostatak uvijek postoje i da su jedinstveni.

U Euklidovoj originalnoj verziji algoritma, kvocijent i ostatak se određuju ponavljanjem oduzimanja; to jest, r_{k-1} se oduzima od r_{k-2} sve dok ostatak r_k ne bude manji od r_{k-1}. Poslije toga se r_k i r_{k-1} razmjenjuju i proces se ponavlja. [1] Euklidska podjela smanjuje sve korake između te dvije razmijene u jedan korak, što je efikasnije. Štoviše, kvocijentima nije potreban, tako da se može zamijeniti euklidska podjela pomoću modulo operacije, koja daje samo ostatak. Tako da iteracija Euklidovog algoritma postaje jednostavno:

$$r_k = r_{k-2} mod r_{k-1}. \qquad (3-11)$$

3.4. Primjer Euklidovog algoritma za cijele brojeve

1) Euklidovim algoritmom nađite najveći zajednički djelitelj brojeva 3102 i 4002.

   Rješenje:

   $4002 = 1 \cdot 3102 + 900$

   $3102 = 3 \cdot 900 + 402$

   $900 = 2 \cdot 402 + 96$

   $402 = 4 \cdot 96 + 18$

   $96 = 5 \cdot 18 + 6$

   $18 = 3 \cdot 3$

   $m(4102, 3102) = 6$

2) Odredite $m = (252, 198)$ i prikažite *d* kao linearnu kombinaciju brojeva 198 i 252.

   Rješenje:

   $252 = 198 \cdot 1 + 54$

   $198 = 54 \cdot 3 + 36$

   $54 = 36 \cdot 1 + 18$

   $38 = 18 \cdot 2$

   $m(252, 198) = 18.$

   $18 = 54 - 38 \cdot 1 = 54 - (198 - 54 \cdot 3) = 4 \cdot 54 - 198 \cdot 1 = 4(252 - 198) - 198 = 4 \cdot 252 - 5 \cdot 198.$

3.5. Prošireni Euklidov algoritam

U aritmetici i programiranju, prošireni Euklidov algoritam je nastavak na Euklidov algoritam koji računa, osim najvećeg zajedničkog djelitelja dvaju cijelih brojeva *a* i *b*, i koeficijente Bezoutovog identiteta, koji su cijeli brojevi *x* i *y* tako da vrijedi:

$$ax + by = nzm(a, b). \qquad (3-12)$$

Prošireni Euklidov algoritam se također odnosi na algoritam koji računa najveću zajedničku mjeru dva univarijantna polinoma. Prilikom korištenja ovog algoritma, stupanj polinoma se smanjuje nakon svakog koraka sve dok nam ne ostane ostatak koji nije jednak nuli. Taj ostatak je najveća zajednička mjera tih polinoma.

Teorem

Za svaka dva polinoma $f(x) \neq 0$ i $g(x) \neq 0$ postoji jednoznačno određena zajednička mjera $m(f, g)$.

Ako je za polinome $f(x)$ i $g(x)$ $m(f, g) = 1$, onda kažemo da su oni relativno prosti polinomi.

Algoritam

Neka su f i g dva polinoma, $\mathrm{st}\, f \geq \mathrm{st}\, g$. Uzastopnom primjenom teorema o dijeljenju s ostatkom dobivamo sljedeći niz jednakosti:

$$f = gq_1 + r_1 \qquad (3-13)$$

$$g = r_1 q_2 + r_2 \qquad (3-14)$$

$$r_1 = r_2 q_3 + r_3 \qquad (3-15)$$

$$\vdots$$

$$r_{k-2} = r_{k-1} q_k + r_k \qquad (3-16)$$

$$r_{k-1} = r_k q_{k+1} \qquad (3-17)$$

Tada je $m(f, g)$ normirani polinom koji se dobije iz r_k. [2]

3.6. Primjer Euklidovog algoritma za polinome

1) $(x^2 + x - 6) \div (x^2 - x - 2) = 1$

$$\begin{array}{r} 1 \\ x^2 - x - 2 \overline{)\, \cancel{x^2} + x - 6} \\ \underline{-(\cancel{x^2} - x - 2)} \\ 2x - 4 \end{array}$$

$x^2 + x - 6 = (x^2 - x - 2)(1) + (x - 2)(2)$

$(x^2 - x - 2) \div (x - 2) = x + 1$

$$\begin{array}{r} x + 1 \\ x - 2 \overline{)\, \cancel{x^2} - x - 2} \\ \underline{-(\cancel{x^2} - 2x)} \\ \cancel{x} - \cancel{2} \\ \underline{-(\cancel{x} - \cancel{2})} \\ 0 \end{array}$$

$x^2 - x - 2 = (x - 2)(x + 1) + 0$

Najveća zajednička mjera: $x - 2$

2) $(x^2 + 7x + 6) \div (x^2 - 5x - 6) = 1$

$$\begin{array}{r} 1 \\ x^2 - 5x - 6 \overline{)\cancel{x^2} + 7x + 6} \\ \underline{-(\cancel{x^2} - 5x - 6)} \\ 12x + 12 \end{array}$$

$x^2 + 7x + 6 = (x^2 - 5x - 6)(1) + (x + 1)(12)$

$(x^2 - 5x - 6) \div (x + 1) = x - 6$

$$\begin{array}{r} x - 6 \\ x + 1 \overline{)\cancel{x^2} - 5x - 6} \\ \underline{-(\cancel{x^2} + x)} \\ \cancel{-6x - 6} \\ \underline{-(\cancel{-6x - 6})} \\ 0 \end{array}$$

$x^2 - 5x - 6 = (x + 1)(x - 6) + 0$

Najveća zajednička mjera: $x + 1$

4. PREGLED KORIŠTENIH ALATA ZA IZRADU PROGRAMA

Program za rješavanje i prikaz Euklidovog algoritma napisan je u razvojnom okruženju Microsoft Visual Studio 2017, u programskom jeziku C#.

4.1. Microsoft Visual Studio 2017

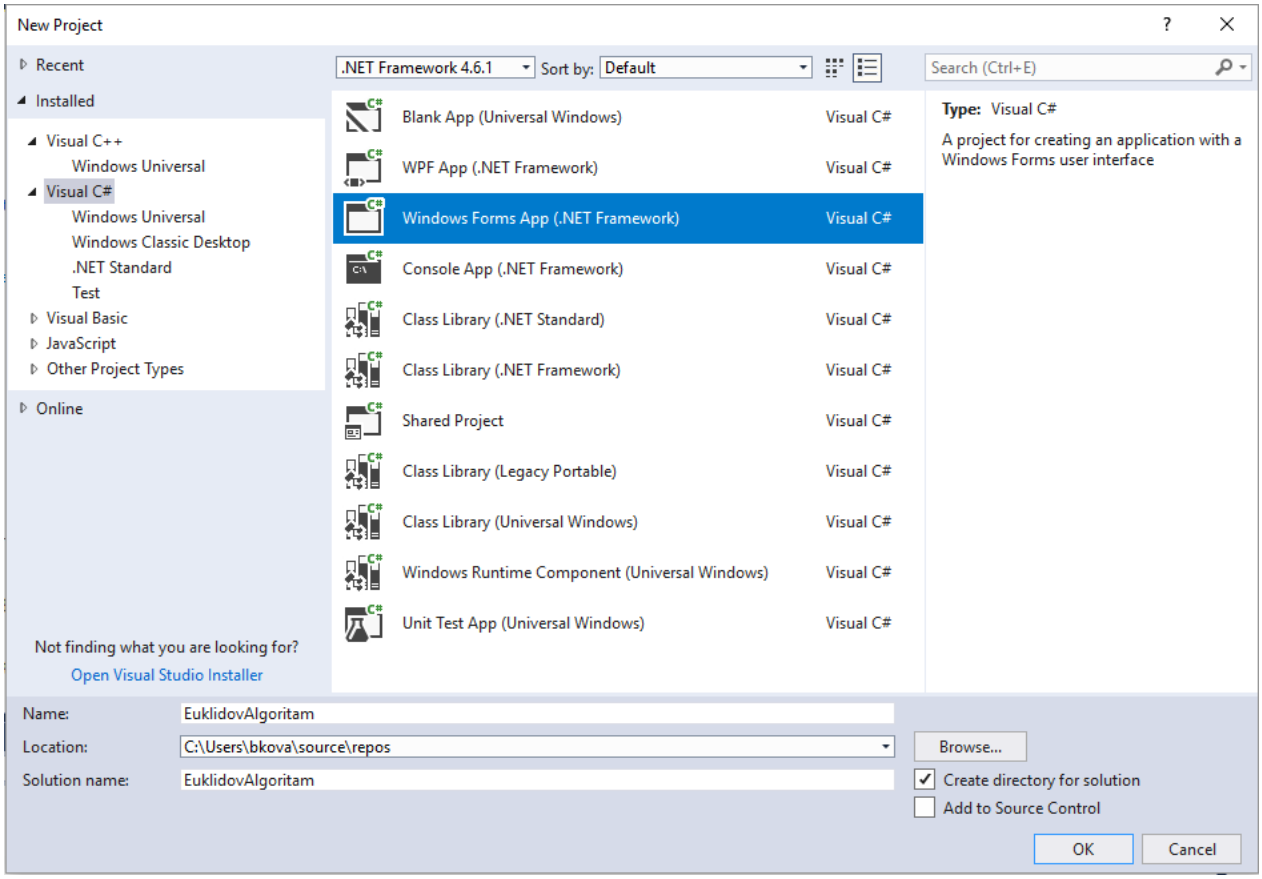
Microsoft Visual Studio 2017 je integrirano razvojno okruženje od Microsofta. Koristi sa za razvoj računalnih programa za Microsoft Windows. Također se može i koristiti za razvoj web stranica, web aplikacija, web usluga i mobilnih aplikacija. Visual studio koristi Microsoftove platforme za razvoj softvera kao što su Windows API, Windows Forme, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Može proizvesti i izvorni i upravljački kod.

Visual Studio podržava 36 različitih programskih jezika i omogućuje uređivaču koda i debuggeru da podrži (u različitim stupnjevima) gotovo bilo koji programski jezik, pod uvjetom da postoji usluga specifična za taj jezik. Ugrađeni jezici uključuju C, C++, VB.NET, C#, F# i TypeScript. Također može podržavati i Pyton, Ruby, Node.js, M, XML/XSLT, HTML/XHTML, JavaScript i CSS koji se mogu instalirati dodatno.

Microsoft pruža besplatnu verziju Visual Studio programskog paketa nazvanu Community edition. Ovaj završni rad napisan je u Microsoft Visual Studio 2017 Community edition. [3]

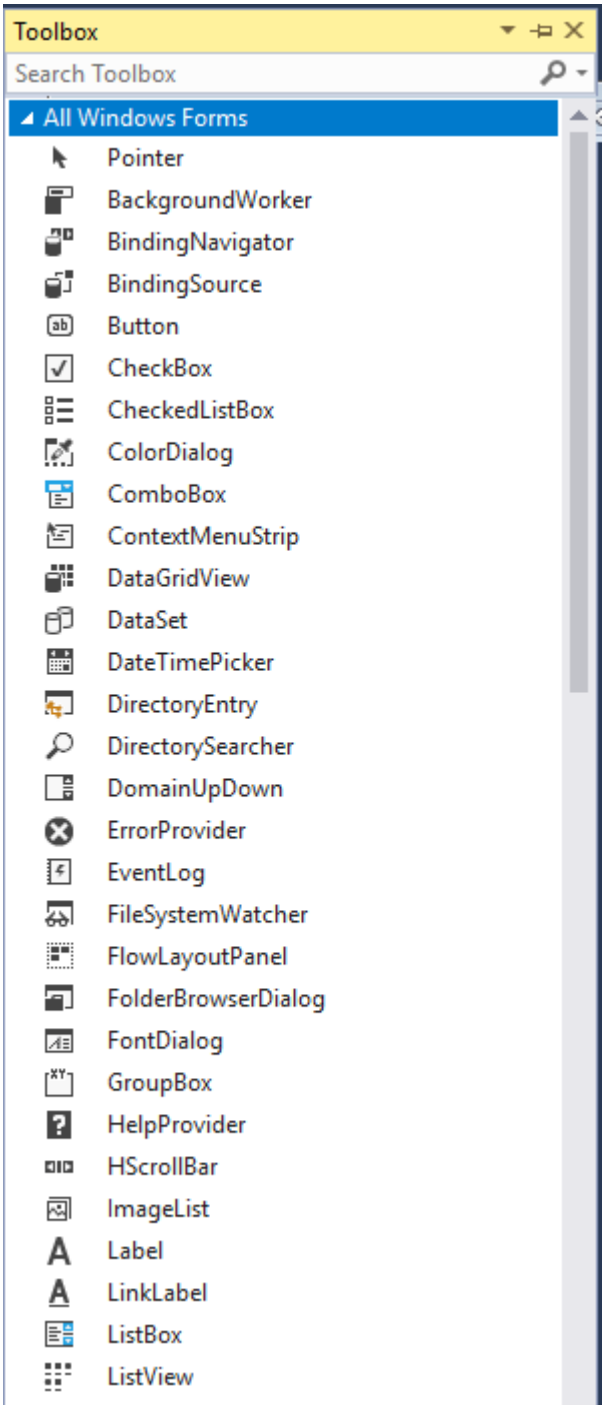
4.2. Windows Forme

Za izradu ovog završnog rada korištena je Windows Forme platforma za razvoj softvera i programski jezik C#. Windows Forme koriste se za razvoj grafičkog sučelja. Slika 4.1. prikazuje početni zaslon novog projekta u Visual Studio-u gdje su odabrane Windows Forme kao platforma za razvoj programa.

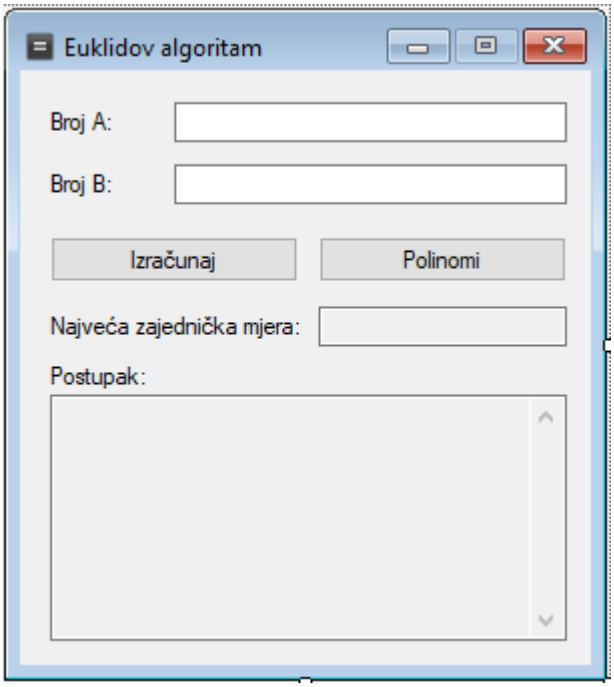


Sl. 4.1. *Početni zaslon novog projekta i odabir C# programskog jezika i Windows Formi*

Forma predstavlja prozor programa koji korisnik vidi te u njemu odabire željene funkcije programa. Izgled programa se kreira „povlačenjem“ željene funkcije za prikaz podataka iz alatne trake, kao npr. tekstualni okvir (u daljnjem tekstu: okvir), oznaka, dugme, potvrdni okvir itd. Na slici 4.2. prikazan je izgled alatne trake.

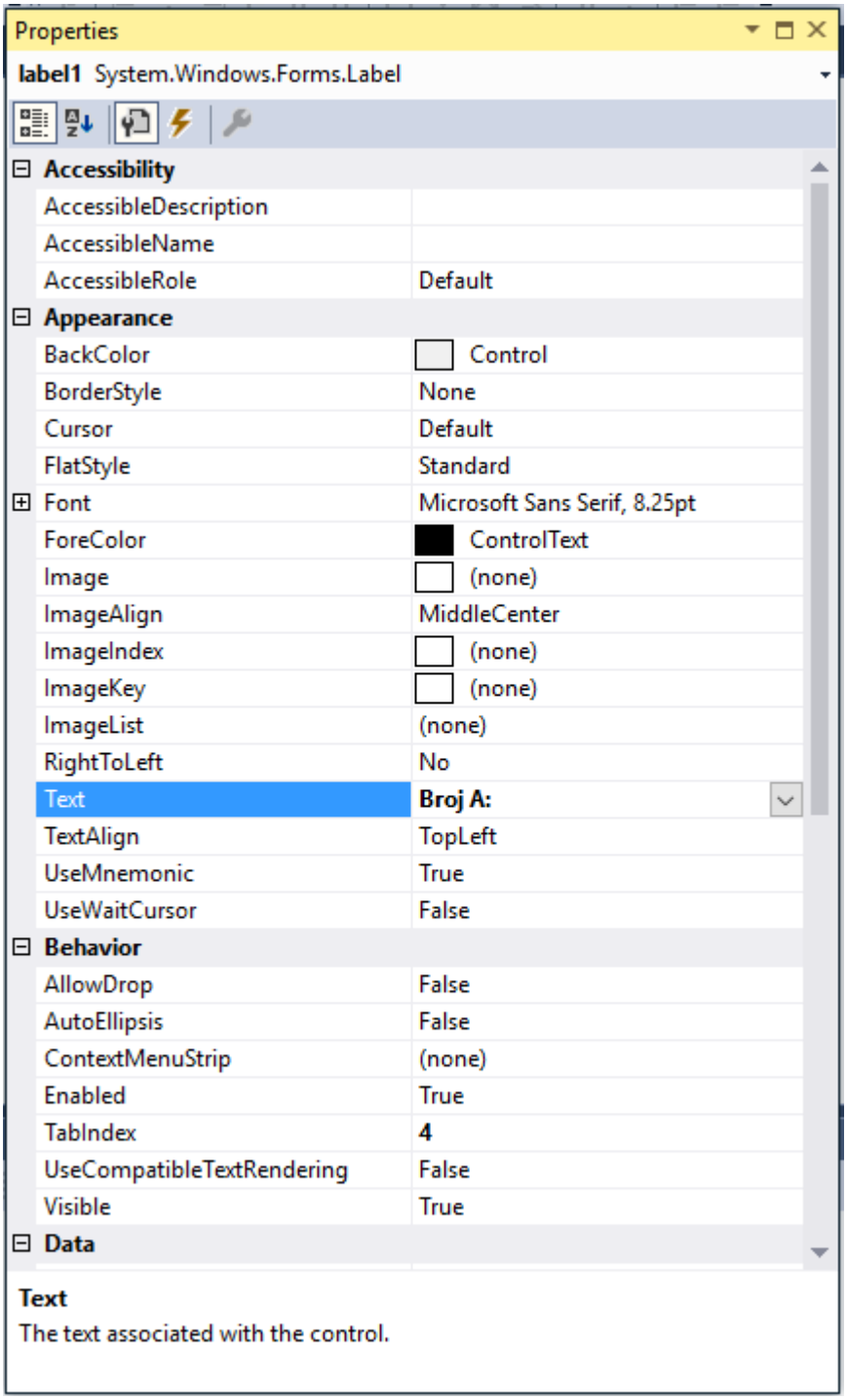


Sl. 4.2. *Alatna traka*



Sl. 4.3. *Prikaz forme*

Na slici 4.3. prikazana je forma u kojoj su već smješteni svi potrebni elementi iz alatne trake. Svaki od tih elemenata može se upravljati pomoću miša ili odgovarajućim izborom u svojstvima. Tako se može kontrolirati lokacija, font, veličina i mnoga druga svojstva svakog od elemenata u formi. Slika 4.4. prikazuje svojstvo oznake „Broj A“ prikazanog na slici 4.3.



Sl. 4.4. *Prikaz prozora svojstava*

Svi ti elementi u formi povezani su programskim kodom koji određuje kako će se oni međusobno ponašati. Na primjer, pritiskom na dugme „Izračunaj“ program će izračunati najveću zajedničku mjeru za brojeve „Broj A“ i „Broj B“ koji korisnik također unosi.

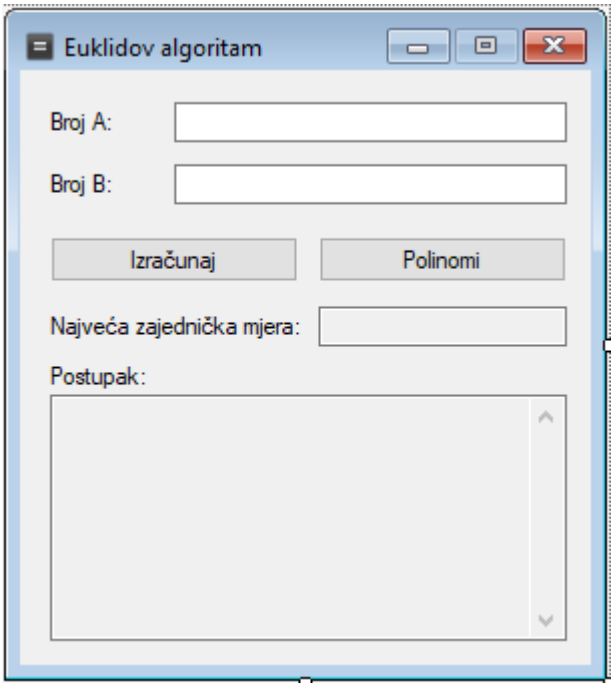
4.3. Microsoft Visual C#

C# (izgovara se C sharp) je multi-paradigmski programski jezik razvijen od Microsofta unutar .NET inicijative. Cilj mu je da bude jednostavan, moderan, opće namjeran i objektno orijentiran. Sličan je programskim jezicima C, C++ i Java po sintaksi. Microsoft Visual C# je implementacija C# jezika unutar Microsoft Visual Studio-a koji je usmjeren na .NET Framework. Visual C# podržava dizajn podataka, dizajn klasa i dizajn formi.

5. IZRADA PROGRAMA

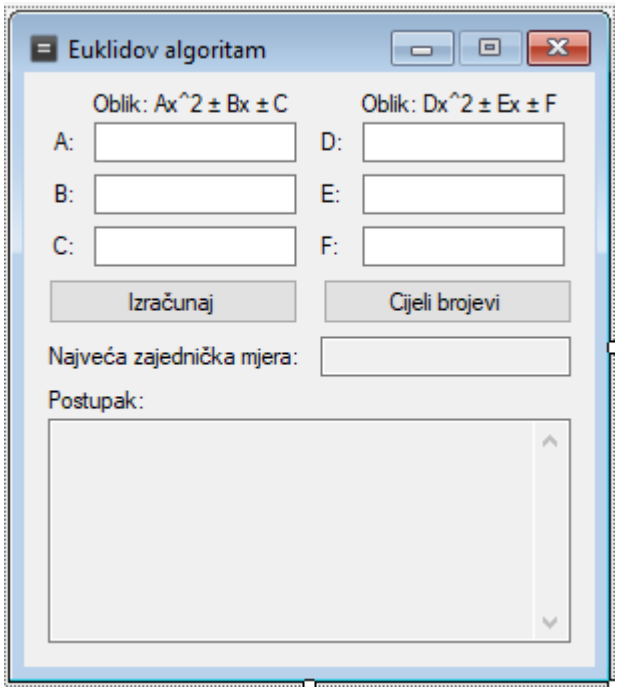
Ideja je bila napraviti program koji će izračunati najveću zajedničku mjeru dvaju cijelih brojeva i dvaju polinoma drugog stupnja koristeći Euklidov algoritam.

U Microsoft Visual Studio 2017 otvoren je novi projekt za izradu desktop aplikacije uporabom Visual C# programskog jezika i Windows formi. Program se sastoji od dvije forme. Prva forma prikazuje okruženje za računanje najveće zajedničke mjere za dva cijela broja koristeći Euklidov algoritam, kao što je prikazano na slici 5.1.



Sl. 5.1. *Prikaz prve forme za računanje cijelim brojevima pri pokretanju programa*

Klikom na dugme „Polinomi“ otvara se druga forma koja prikazuje okruženje za računanje najveće zajedničke mjere za dva polinoma drugog stupnja koristeći Euklidov algoritam, kao što je prikazano na slici. 5.2. Klikom na dugme „Cijeli brojevi“ ponovno se otvara prva forma, tako da korisnik može cijelo vrijeme mijenjati između računanja najveće zajedničke mjere za cijele brojeve ili polinome.



Sl. 5.2. *Prikaz druge forme za računanje polinomima*

5.1. Program za računanje cijelim brojevima

U prvoj formi za dizajn korištena su 3 elementa: oznaka, dugme i okvir. Oznaka služi za označavanje pojedinog okvir. Tako da prvi okvir je označen kao „Broj A:“, drugi kao „Broj B:“, postupak rješavanja Euklidovog algoritma je označen sa „Postupak:“ i na kraju okvir za krajnji rezultat je označen sa „Najveća zajednička mjera:“. Također postoje i dva dugmeta, „Izračunaj“ i „Cijeli brojevi“. Već je opisano u prijašnjem odlomku što radi dugme „Cijeli brojevi“, a dugme „Izračunaj“ je glavni dio programa jer se pritiskom na to dugme zapravo odvijaju sve operacije u programu, što će biti opisano kasnije.

5.1.1. Unos cijelih brojeva

Otvaranjem programa korisnik zatiče prazna polja kao na slici 5.1. te unosi željena dva cijela broja u prva dva okvira. Ni jedan od brojeva ne smije biti nula jer to je uvjet za računanje najveće zajedničke mjere. Također i postoji ograničenje da polja ne smiju biti prazna i da se smije unijeti samo jedan minus u svaki okvir i to samo na početku, te da ne smije biti unesen samo minus u okvir.

```
private void button1_Click(object sender, EventArgs e)
{
	if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text))
	{
		MessageBox.Show("Polja ne smiju biti prazna!");
	}
	else if (textBox1.Text == "-" || textBox2.Text == "-")
	{
		MessageBox.Show("Polja ne smiju sadržavati samo minus!");
	}
	else if (int.Parse(textBox1.Text) == 0 || int.Parse(textBox2.Text) == 0)
	{
		MessageBox.Show("Broj ne smije biti 0!");
	}
	else
	{
		textBox4.Clear();
		Int32 NZD = NajmanjiZajednickiDjelitelj(Convert.ToInt32(textBox1.Text),
	Convert.ToInt32(textBox2.Text));
		this.textBox3.Text = NZD.ToString();
	}
}
```

Program sve te uvjete provjerava pritiskom na dugme „Izračunaj“ i ako su svi uvjeti zadovoljeni program se nastavlja te ispisuje postupak u textbox4 te krajnji rezultat odnosno najveću zajedničku mjeru u textbox3.

5.1.2. Računanje najveće zajedničke mjere

Pritiskom na dugme „Izračunaj“ poziva se funkcija koja izračunava najveću zajedničku mjeru dva unesena cijela broja.

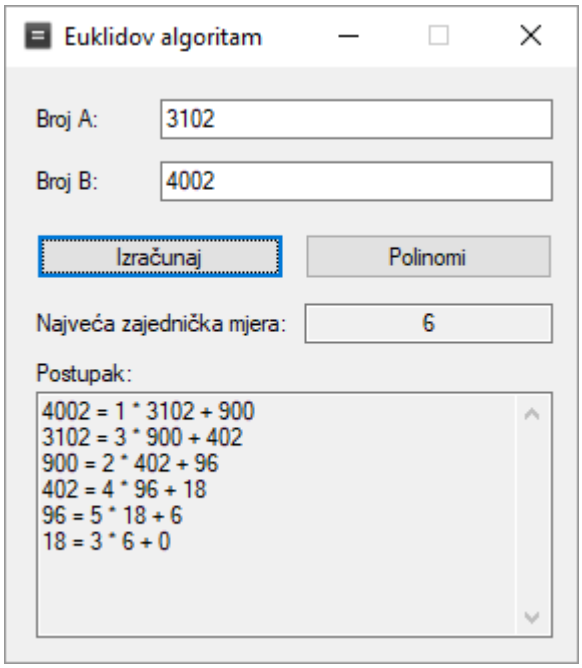
```
public int NajmanjiZajednickiDjelitelj(int a, int b)
{
	int ostatak;
	if (a < 0)
		a = a * -1;
	if (b < 0)
		b = b * -1;
	while (a != 0 && b != 0)
	{
		if (a > b)
		{
			int c;
			c = a / b;
			ostatak = a % b;
			textBox4.Text += a + " = " + c + " * " + b + " + " + ostatak +
		Environment.NewLine;
			a = b;
			b = ostatak;
		}
		else
		{
			int c;
			c = b / a;
			ostatak = b % a;
			textBox4.Text += b + " = " + c + " * " + a + " + " + ostatak +
		Environment.NewLine;
			b = a;
			a = ostatak;
		}
	}
	if (a == 0)
	{
		return b;
	}
	else
	{
		return a;
	}

}
```

Ako su uneseni brojevi *a* ili *b* negativni, funkcija ih pretvara u pozitivne.

```
if (a < 0)
        a = a * -1;
if (b < 0)
        b = b * -1;
```

Postupak se ispisuje u textbox4 nakon svakog koraka, sve dok ostatak ne bude nula.

```
textBox4.Text += a + " = " + c + " * " + b + " + " + ostatak + Environment.NewLine;
```



Sl. 5.3. *Primjer izvođenja programa za cijele brojeve*

5.2. Program za računanje polinomima

Pritiskom na dugme „Polinomi“ otvara se druga forma (Sl. 5.2.) koja služi za računanje najveće zajedničke mjere dva polinoma drugog stupnja pomoću Euklidovog algoritma.

```
private void button2_Click(object sender, EventArgs e)
{
    this.Hide();
    Form2 f2 = new Form2();
    f2.ShowDialog();
    this.Close();
}
```

5.2.1. Unos koeficijenata

Otvaranjem forme za računanje polinomima se vidi da ima 6 praznih polja za unos koeficijenata polinoma. Uvjeti za unos tih koeficijenata su slični kao i kod prve forme, znači da u polju smije biti samo jedan minus, i to na prvom mjestu. Polja se ne smiju ostaviti prazna. Jedina razlika je što je dozvoljeno da u svakom polju bude samo nula.

Za rad i operacije sa polinomima korišten je Extreme.Mathematics.Curves library.[4]

```
using Extreme.Mathematics.Curves;
```

5.2.2. Računanje najveće zajedničke mjere

Pritiskom na dugme „Izračunaj“ kreće izvod programa. Program prvo stvara dva polinoma od koeficijenta koji je korisnik unio u polja.

```
Polynomial polinom1 = new Polynomial(new Double[] { Convert.ToInt32(textBox3.Text),
Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox1.Text) });
```

```
Polynomial polinom2 = new Polynomial(new Double[] { Convert.ToInt32(textBox6.Text),
Convert.ToInt32(textBox5.Text), Convert.ToInt32(textBox4.Text) });
```

Može se dogoditi da jedan ili oba dva polinoma nisu drugog stupnja. Zato postoje uvjeti koji se na to odnose.

```
int degree = polinom1.Degree;//stupanj prvog polinoma
int degreee = polinom2.Degree;//stupanj drugog polinoma
```

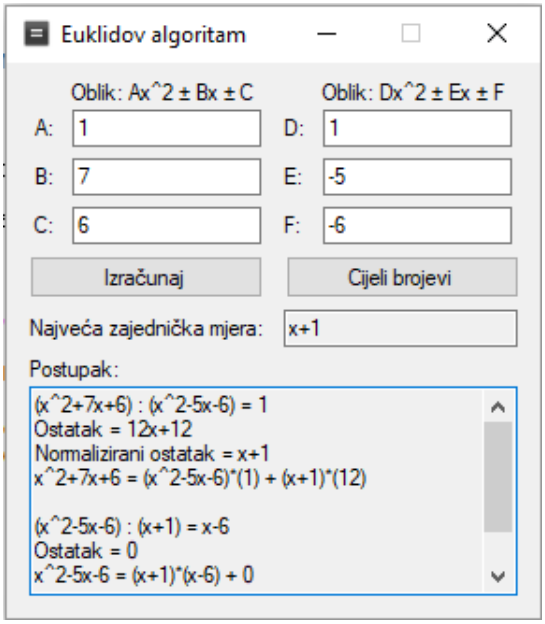
Slučaj kada su oba polinoma drugog stupnja

U ovom slučaju program dijeli prvi polinom sa drugim i izbacuje ostatak njihovog dijeljenja.

```
Polynomial ostatak1;
Polynomial rezultat1 = Polynomial.Divide(polinom1, polinom2, out ostatak1);
```

Zatim se provjerava stupanj tog ostatka dijeljenja, te ako je stupanj tog ostatka jednak 1 onda je drugi polinom bio drugog stupnja. Tada program provjerava treba li se taj ostatak normalizirati. Normalizira ga tako što traži najveću zajedničku mjeru oba koeficijenta ostataka, te nakon toga dijeli svaki koeficijent sa tom mjerom. Iz toga nastaje normalizirani ostatak. Za normaliziranje korištena je funkcija za traženje najveće zajedničke mjere iz prve forme. Nakon toga slijedi dijeljenje drugog polinoma tim ostatkom, normaliziranim ili ne (ovisno o potrebi). U sljedećem kodu prikazan je slučaj kada se ostatak ne treba normalizirati. Prvo se ispisuje u textbox8 postupak dijeljenja prva dva polinoma. Zatim slijedi dijeljenje drugog polinoma ostatkom prvog dijeljenja. Nakon toga se provjerava da li je prvi broj drugog ostatka jednak nuli. Ako je slijedi provjera da li je taj ostatak prvog stupnja ili ne. Ako je onda se ispisuje taj ostatak u textbox8 te postupak dijeljenja i na kraju krajnji rezultat, odnosno najveća zajednička mjera koja je u tom slučaju 1. Ako ostatak nije prvog stupnja to znači da ostatak uistinu je nula i to znači da je najveća zajednička mjera prvi ostatak koji je nastao dijeljenjem dva unesena polinoma. Ako prvi broj drugog ostatka nije nula onda je najveća zajednička mjera 1.

```
textBox8.Text += polinom1.ToString() + " = " + "(" + polinom2.ToString() + ")*(" +
rezultat1.ToString() + ") + (" + ostatak1.ToString() + ")" + Environment.NewLine +
Environment.NewLine;
Polynomial ostatak2;
Polynomial rezultat2 = Polynomial.Divide(polinom2, ostatak1, out ostatak2);
textBox8.Text += "(" + polinom2.ToString() + ") : (" + ostatak1.ToString() + ") = " +
rezultat2.ToString() + Environment.NewLine;
int degree2 = ostatak2.Degree;
if (ostatak2[0] == 0)
{
	if (degree2 == 1)
	{
		textBox8.Text += "Ostatak = " + ostatak2.ToString() + Environment.NewLine;
		textBox8.Text += polinom2.ToString() + " = " + "(" + ostatak1.ToString() +
		")*(" + rezultat2.ToString() + ") + (" + ostatak2.ToString() + ")" +
		Environment.NewLine;
		textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
		this.textBox7.Text = "1";

	}
	else
	{
		textBox8.Text += "Ostatak = 0" + Environment.NewLine;
		textBox8.Text += polinom2.ToString() + " = " + "(" + ostatak1.ToString() +
		")*(" + rezultat2.ToString() + ") + 0" + Environment.NewLine;
		textBox8.Text += "Najveća zajednička mjera: " + ostatak1.ToString() +
		Environment.NewLine;
		this.textBox7.Text = ostatak1.ToString();
	}
}
else
{
	textBox8.Text += "Ostatak = " + ostatak2.ToString() + Environment.NewLine;
	textBox8.Text += polinom2.ToString() + " = " + "(" + ostatak1.ToString() + ")*(" +
	rezultat2.ToString() + ") + (" + ostatak2.ToString() + ")" + Environment.NewLine;
	textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
	this.textBox7.Text = "1";
}
```

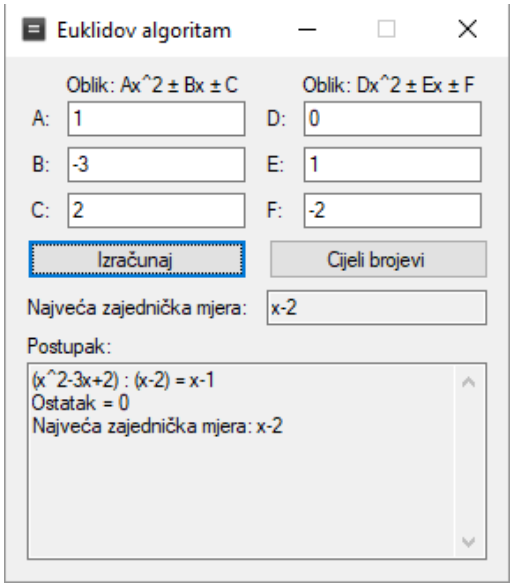


Sl. 5.4. *Primjer izvođenja programa za polinome drugog stupnja*

Slučaj kada je drugi polinom prvog stupnja

Ako je drugi polinom prvog stupnja onda program provjerava da li je ostatak jednak nuli ili ne. Ako je onda se poziva funkcija „NajmanjiZajednickiDjelitelj“ koja služi na normalizaciju ako je potrebno. Najveća zajednička mjera je u tom slučaju taj normalizirani drugi polinom. Ako ostatak nije nula onda je najveća zajednička mjera jednaka 1.

```
//slučaj kada je drugi polinom prvog stupnja
else
{
	textBox8.Text += "(" + polinom1.ToString() + ") : (" + polinom2.ToString() + ") =
	" + rezultat1.ToString() + Environment.NewLine;
	//slučaj kada je ostatak 0
	if (ostatak1[0] == 0)
	{
		textBox8.Text += "Ostatak = 0" + Environment.NewLine;
		double NZD = NajmanjiZajednickiDjelitelj(polinom2[0], polinom2[1]);
		double nnorm1 = polinom2[1] / NZD;
		double nnorm2 = polinom2[0] / NZD;
		Polynomial nzm = new Polynomial(new Double[] { Convert.ToInt32(nnorm2),
		Convert.ToInt32(nnorm1) });
		textBox8.Text += "Najveća zajednička mjera: " + nzm.ToString() +
		Environment.NewLine;
		this.textBox7.Text = nzm.ToString();
	}
	//slučaj kada ostatak nije 0
	else
	{
		textBox8.Text += "Ostatak = " + ostatak1.ToString() + Environment.NewLine;
		textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
		this.textBox7.Text = "1";
	}
}
```

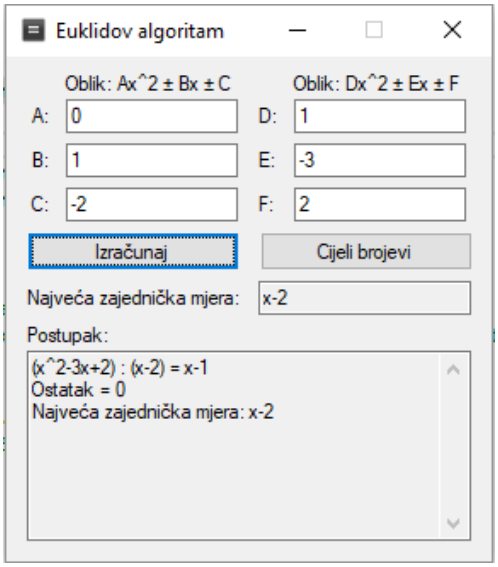


Sl. 5.5. *Primjer izvođenja programa kada je drugi polinom prvog stupnja.*

Slučaj kada je prvi polinom prvog stupnja

Ako je prvi polinom prvog stupnja postupak izvođenja je sličan slučaju kada je drugi polinom drugog stupnja. U kodu se samo izmjene pozicije prvog i drugog polinoma tako da je drugi polinom tada prvi, a prvi drugi.

```
//slučaj kada je prvi polinom prvog stupnja
else if (degree == 1)
{
	Polynomial ostatak1;
	Polynomial rezultat1 = Polynomial.Divide(polinom2, polinom1, out ostatak1);
	textBox8.Text += "(" + polinom2.ToString() + ") : (" + polinom1.ToString() + ") =
	" + rezultat1.ToString() + Environment.NewLine;
	//slučaj kada je ostatak jednak 0
	if (ostatak1[0] == 0)
	{
		textBox8.Text += "Ostatak = 0" + Environment.NewLine;
		double NZD = NajmanjiZajednickiDjelitelj(polinom1[0], polinom1[1]);
		double nnorm1 = polinom1[1] / NZD;
		double nnorm2 = polinom1[0] / NZD;
		Polynomial nzm = new Polynomial(new Double[] { Convert.ToInt32(nnorm2),
		Convert.ToInt32(nnorm1) });
		textBox8.Text += "Najveća zajednička mjera: " + nzm.ToString() +
		Environment.NewLine;
		this.textBox7.Text = nzm.ToString();
	}
	//slučaj kada ostatak nije 0
	else
	{
		textBox8.Text += "Ostatak = " + ostatak1.ToString() + Environment.NewLine;
		textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
		this.textBox7.Text = "1";
	}
}
```

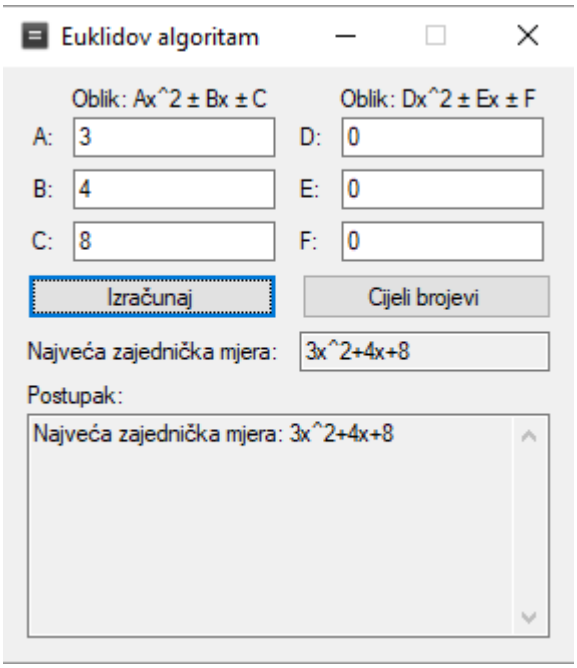


Sl. 5.6. *Primjer izvođenja programa kada je prvi polinom prvog stupnja.*

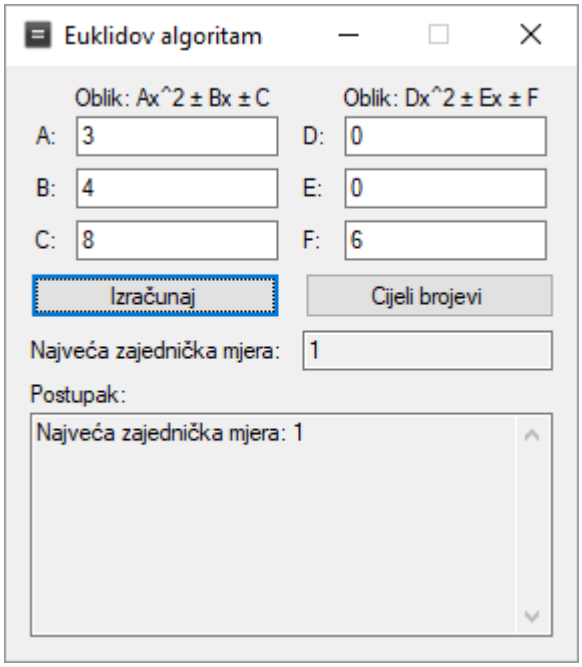
Slučaj kada je drugi polinom nultog stupnja

Kada je drugi polinom nultog stupnja tada program provjerava da li su svi parametri drugog polinoma jednaki nuli. Ako jesu onda je najveća zajednička mjera jednaka prvom polinomu. U suprotnom slučaju najveća zajednička mjera je jedan.

```
//slučaj kada je drugi polinom nultog stupnja
else
{
	//slučaj kada su svi parametri drugog polinoma jednaki 0
	if (polinom2[0] == 0 && polinom2[1] == 0 && polinom2[2] == 0)
	{
		textBox8.Text += "Najveća zajednička mjera: " + polinom1.ToString() +
		Environment.NewLine;
		this.textBox7.Text = polinom1.ToString();
	}
	//slučaj kada su prva 2 parametra drugog polinoma jednaka 0
	else
	{
		textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
		this.textBox7.Text = "1";
	}
}
```



Sl. 5.7. *Primjer kada su svi parametri drugog polinoma jednaki nuli.*

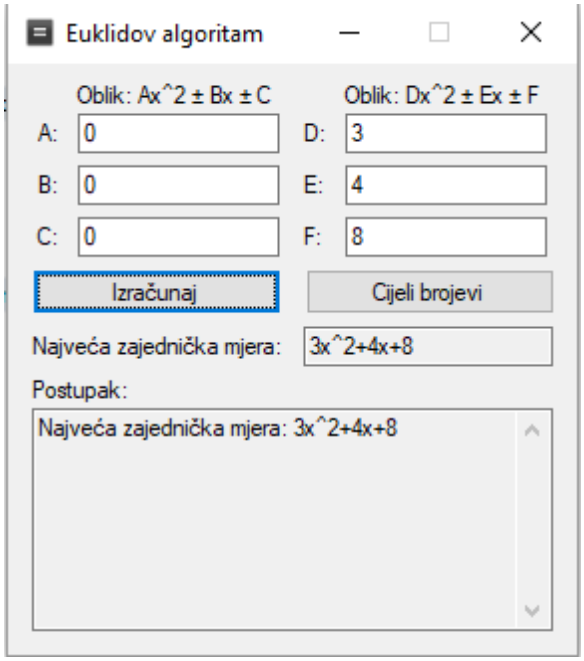


Sl. 5.8. *Primjer kada je drugi polinom nultog stupnja.*

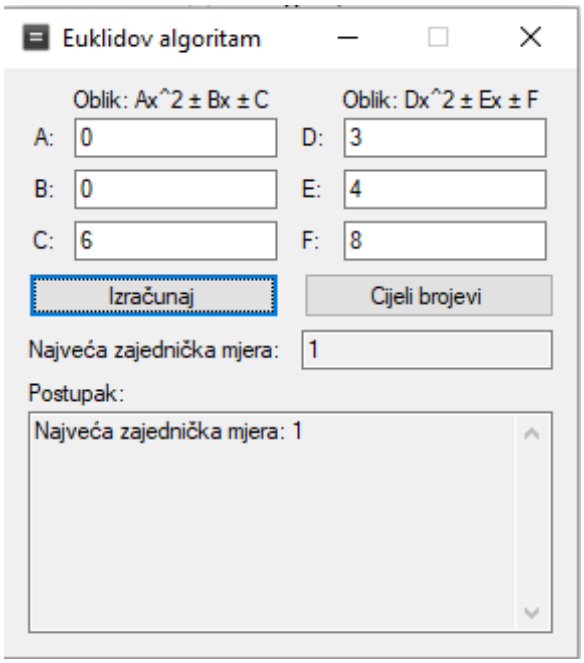
Slučaj kada je prvi polinom nultog stupnja

Ako je prvi polinom nultog stupnja, proces je sličan kao i kod slučaja kada je drugi polinom nultog stupnja. U kodu se samo zamijeni raspored polinoma, umjesto „polinom2“ koristi se „polinom1“.

```
//slučaj kada je prvi polinom nultog stupnja
else
{
	//slučaj kada su svi parametri prvog polinoma jednaki 0
	if (polinom1[0] == 0 && polinom1[1] == 0 && polinom1[2] == 0)
	{
		textBox8.Text += "Najveća zajednička mjera: " + polinom2.ToString() +
		Environment.NewLine;
		this.textBox7.Text = polinom2.ToString();
	}
	//slučaj kada su prva 2 parametra prvog polinoma jednaka 0
	else
	{
		textBox8.Text += "Najveća zajednička mjera: 1" + Environment.NewLine;
		this.textBox7.Text = "1";
	}
}
```



Sl. 5.9. *Primjer kada su svi parametri prvog polinoma jednaki nuli.*



Sl. 5.10. *Primjer kada je prvi polinom nultog stupnja.*

6. ZAKLJUČAK

U ovom radu je napravljen program koji računa najveću zajedničku mjeru dva cijela broja ili dva polinoma.

Rad je zahtijevao detaljno proučavanje Euklidovog algoritma i njegovo matematičko rješenje. Euklidov algoritam je najbolje i najbrže rješenje za određivanje najveće zajedničke mjere velikih brojeva i polinoma. Cilj ovog rada je bilo usavršavanje znanja iz matematike i programiranja. Izrada ovog programa zahtijevala je primjenu prijašnje stečenih znanja i proširivanja tog znanja radi izrade funkcionalnog i detaljnog programa.

Programski dio ovog rada napravljen je u programskom okruženju Microsoft Visual Studio 2017 koji je jedan od najboljih okruženja za izradu programa na Microsoft Windows operativnom sustavu. Korišten je programski jezik C# koji ima mogućnost korištenja Windows formi. Njihovim korištenjem omogućena je relativno jednostavna izrada programa i njegovog grafičkog sučelja.

Zadatak rada je bio napraviti program u programskom jeziku C koji će izračunati najveću zajedničku mjeru dva broja ili polinoma koristeći Euklidov algoritam. Program se sastoji od dva dijela, prvi računa najveću zajedničku mjeru za cijele brojeve, a drugi za polinome. Oba dva dijela aplikacije su testirana s obzirom na sve moguće probleme koji su se mogli pojaviti pri izradi programa. Zaključno tome može se zaključiti kako program radi ispravno.

Postoje mogućnosti poboljšanja izrađenog programa. Moguće je u prvom dijelu dodati više parametara za računanje najveće zajedničke mjere. U drugom dijelu se može proširiti stupnjevi zadanih polinoma.

LITERATURA

[1] Euclidean algorithm, https://en.wikipedia.org/wiki/Euclidean_algorithm#Extended_Euclidean_algorithm

(Pristup 15. rujan 2017.)

[2] Exteded Euclidean algorithm, https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm

(Pristup 15. rujan 2017.)

[3] Visual Studio, www.visualstudio.com

(Pristup 15. rujan 2017.)

[4] Extreme.Mathematics.Curves library, http://www.extremeoptimization.com/

(Pristup 15. rujan 2017.)

SAŽETAK

Euklidov algoritam je najbolje i najbrže rješenje za određivanje najveće zajedničke mjere velikih brojeva i polinoma. U radu se prikazuje teorijska matematička podloga Euklidovog algoritma i njegova primjena. Napravljen je program u programskom jeziku C# u razvojnom okruženju Microsoft Visual Studio 2017 koji računa najveću zajedničku mjeru za dva cijela broja ili za 2 polinoma koristeći Euklidov algoritam, te prikazuje postupak računanja.

Pri izradi programa za računanje najveće zajedničke mjere bilo je potrebno obratiti pozornost na sve različite kombinacije početnih parametara koje korisnik može unijeti. U dijelu programa koji računa cijelim brojevima bilo je potrebno omogućiti točno računanje za slučaj kada su uneseni brojevi negativni, te slučaj kada je prvi broj manji od drugog. Kod računanja polinoma trebalo je obratiti pozornost na stupnjeve prvog i drugog polinoma te napraviti program tako da se može točno izračunati najveća zajednička mjera za svaku od kombinacija.

Ključne riječi:

Euklidov algoritam, Visual C#

ABSTRACT

Title:

Euclid's algorithm for greatest common divisor in programming language C

Euclid's algorithm is the best and fastest solution for determining the greatest common divisor of large numbers and polynomials. This thesis interprets theoretical mathematics background of Euclid's algorithm and it's uses. A C# program was developed in Microsoft Visual Studio 2017 that computes the greatest common divisor of two whole numbers or two polynomials using Euclid's algorithm, and displays the computation process.

When designing a program for computing the greatest common divisor, it was necessary to pay attention to all the different combinations of initial parameters that the user can enter. In the part of the program that counts whole numbers, it was necessary to enable accurate calculation in case that the input numbers are negative, and the case when the first number is smaller than the other. When calculating polynomials, attention should be paid to the degrees of the first and second polynomials and to create a program so that the greatest common divisor for the each of the combinations can be accurately calculated.

Key words:

Euclid's algorithm, Visual C#

Borna Kovač

Sjenjak 46

31000 Osijek

Mobitel: 091/595-6559

E-mail: b2kovac@gmail.com

ŽIVOTOPIS

Borna Kovač rođen je 28. kolovoza 1993. godine u Osijeku.

Student je 3. godine sveučilišnog preddiplomskog studija elektrotehnike, smjer komunikacije i informatika, na Sveučilištu Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija.

2012. godine je završio srednju Elektrotehničku i prometnu školu u Osijeku i stekao zvanje elektrotehničar.

Aktivno se bavi odbojkom već 12 godina. Prvo je igrao za MOK „Mursa“, a trenutno igra za odbojkaški klub MOK „Željezničar“.

Tijekom studija je sudjelovao na sveučilišnim natjecanjima iz odbojke. Dvije godine zaredom (2014. i 2015.) je sa fakultetom osvojio prvo mjesto. 2015. i 2016. je predstavljao fakultet na Elektrijadi igrajući odbojku. 2016. je predstavljao sveučilište igrajući odbojku na državnom prvenstvu.

U Osijeku, 18.09.2017. Borna Kovač

Potpis:

_____