Rješavanje magičnih kvadrata u programskom jeziku C

Lorger, Lea

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

Rješavanje magičnih kvadrata u programskom jeziku C

Završni rad

Lea Lorger

Osijek, 2016.

SADRŽAJ

1. UVOD

Magični kvadrat je kvadratna tablica popunjena prirodnim brojevima od 1 do n^2, te su zbrojevi stupaca, redaka i dvije glavne dijagonale jednaki. Ovakav kvadrat prikazuje određene pravilnosti s obzirom na brojeve koji su u njoj raspoređeni. Kvadratna matrica M veličine M×M može biti ispunjena svim brojevima 1,2,3,...,M^2 . Broj M naziva se red magičnog kvadrata, a zbroj elemenata po recima (stupcima ili po dijagonalama) naziva se magična suma ili magična konstanta, te se označava sa S(M). Postoje magični kvadrati svih dimenzija osim dimenzije 2x2. Trivjalni magični kvadrat dimenzije 1 sastoji se samo od jednog kvadrata.

U daljnim stranicama ovog završnog rada pobliže su objašnjeni magični kvadrati, poznati još pod nazivom čarobne četvorine; programsko okruženje u kojem je program izrađen, te objašnjenje koda. Zadatak rada je napisati program koji provjerava odgovaraju li zadane matrice pravilima magičnih kvadrata.

1.1. Zadatak završnog rada

Zadatak završnog rada je rješavanje problema magičnih kvadrata u programskom jeziku C, odnosno pisanje programa koji za zadanu veličinu matrice M ispisuje popunjenu matricu, takvu da ona bude magična.

2. PROGRAMSKO OKRUŽENJE C

2.1. Uvod u programske jezike

Programski jezici su programi ili upute za računalo zapisani upotrebom određene sintakse i pravila koja vrijede za svaki programski jezik, koji se potom prevodi u strojni jezik. Možemo ih usporediti s bilo kojim govornim jezikom jer ima određene znakove i simbole koji imaju neko značenje. Osnovna podjela programskih jezika je na niže: strojni jezici i simbolički jezici; te na više programske jezike.

Strojni jezik (*engl. machine language, machine code*) je jedini oblik programskog jezika kojeg računalo razumije, što znači da se svi ostali oblici programa pisanih u nekim drugim jezicima moraju prevesti u taj oblik. Nastao je u ranim 50-im godinama prošlog stoljeća. Strojni jezik je u binarnom obliku, što znači da se koriste samo 2 elementa, a to su nule i jedinice. Prevođenje s višeg programskog jezika na strojni jezik provodi se preko programa prevoditelja, odnosno kompajlera ili se naredbe u višem jeziku izravno prevode preko takozvanog p_koda u strojni jezik. Izvršnim programom nazivamo program napisan u strojnom jeziku. Strojni jezik je određen arhitekturom računala te ga definira proizvođač sklopovlja.

Simbolički jezici su jezici koji su razvijeni zbog nepreglednog i nepraktičnog pisanja instrukcija u binarnom kodu. Nastali su jer je ljudima jednostavnije pratiti i pamtiti riječi, a nizove nula i jedinica zamjenila su slova i riječi. Pisanje programa započinje programerovim unosom mnemoričkih oznaka u tekstualnu datoteku pomoću uređivača teksta te se nakon toga poziva asembler - program koji prevodi mnemoričke instrukcije u binarne instrukcije strojnog jezika koje računalo razumije. Taj program napisan u asemblerskom jeziku nazivamo izvorni program. Dakle, pisanje programa je proces koji obuhvaća pisanje izvornog programa, te njegovo prevođenje u binarne instrukcije koje računalo razumije. Postoje simbolički jezici niže i više razine.

Asemblerski jezik, odnosno Asembler, je jezik niže razine prilagođen radu računala. Nastao je polovicom 50-ih godina prošlog stoljeća. Svaka instrukcija u asembleru predstavlja jednu instrukciju strojnog jezika. Iako je Asembler jezik niže razine, mnogo je napredniji u odnosu na strojni jezik. Sam način programiranja nije bitno različit u odnosu na strojni jezik, ali je svaki binarni kod zamijenjen slovom i oznakom tako da je ovaj programski jezik mnogo razumljiviji i jednostavniji ljudima. U njemu se pojedine naredbe označavaju skraćenicama koje podsjećaju na svoju namjenu (LD ili MOV - učitavanje, MUL – množenje, ADD – zbrajanje itd.).

Programski jezici više razine nastali su oko 1960. godine. U odnosu na programske jezike niske razine, mogu biti apstraktniji, lakši za uporabu ili lakše prenosivi po platformama. Naredbe su kratke riječi, najčešće vezane uz englesko govorno područje. One se brže pamte i bliže su ljudskom načinu razmišljanja. Korištenjem ovih programskih jezika brže izrađujemo kraće i razumljivije programe. Najpopularniji viši programski jezici su BASIC, FORTRAN, Pascal, C itd. Ti jezici su jezici opće namjene.

Viši jezici pak mogu biti sekvencijalni, proceduralni (Pascal, C), funkcijski (Lisp, Erlang, ML), te objektno orijentirani (C++, Java).

2.2. Programski jezik C

C je jedan od najpopularnijih programskih jezika današnjice. Gotovo da nema nijednog aplikacijskog programa koji nije napisan u C-u, a to se odnosi i na većinu operacijskih sustava. Razlog za ovakvu rasprostanjenost jezika leži u tome da, pored toga što se C smatra višim programskih jezikom (uključuje sve elemente viših jezika), dovoljno je niskog nivoa da posjeduje mogućnost rezerviranu samo za assembler. Naime C omogućava direktan pristup memoriji i manipuliranje bitovima. Zbog ovakvih svojstava C je i zaslužio glas jezika pogodnog za sistematsko programiranje i razvoj operacijskih sustava.

Pored toga C omogućava pisanje racionalnog koda koji je vrlo brz uz odgovarajuće optimizacije kao i programa prenosivih između različitih računala, pa se jezik intenzivno primjenjuje i za pisanje aplikacijskih programa. Jasno je da će programi pisani na asembleru biti brži, ali je programiranje u C-u mnogo lakše, brže i pouzdanije.

C snažno podržava različitost različitih tipova podataka: pored osnovnih tipova (znakova, cjelobrojnih i realnih vrijjednosti u više veličina) mogu se definirati izvedeni tipovi kreirani pokazivačima, poljima, strukturama i unijama.

Podržano je mnoštvo operatora uključujući i operatore na nivou bita. Izrazi se mogu pisati u skraćenom obliku, te se i sami mogu koristiti kao arguenti funkcija.

C sadrži i pretprocesor koji prije kompiliranja programa vrši transformaciju teksta programa, uključivanje izvornih datoteka i uvjetno kompiliranje.

C ne sadržava ulazno-izlazne naredbe (izlaz na ekran, ulaz sa tipkovnice, rad s datotekama) kao i rad sa stringovima, te dinamičko alociranje memorije. Sve te operacije u obliku funkcija sadrži standardna biblioteka koja dolazi uz svaki C kompilator.
Tijekom svog razvoja C je ograničio mogućnosti pojavljivanja pogrešaka uvođenjem strože kontrole tipova i drugih osobina preuzetih od C++ jezika.

2.2.1. Povijest programskog jezika C

Programski jezik C svoje korijene vuče iz BCPL-a (Based Combined Programming Language) autora Martina Richardsa (MIT Boston, 1967.) koji se u to doba koristio za sistematsko programiranje. Ovaj je jezik poslužio Kenu Thompsonu iz Bell Laboratoriesa za razvoj novog jezika pod imenom B. B je razvijen za pisanje prve verzije operacijskog sustava UNIX na DEC-ovom računalu PDP-11. Uvođenjem novih koncepata i neovisnosti jezika od računala Dennis Ritchie je 1972. godine na temelju BCPL-a i B-a definirao novi jezik i nazvao ga C. Godinu dana kasnije Ken Thompson i Dennis Ritchie (Bell Laboratories) napisali su u C-u UNIX operacijski sustav i predstavili ga javnosti 1974. godine.

C se dalje razvijao da bi pojavom knjige "The Programming Language" (autora Dennisa Ritchiea i Briana W.Kernighana) 1978. godine jezik bio po prvi puta standardiziran. Zbog događaja koji su slijedili – pojave različitih verzija kompilatora koje su prijetilie narušavanju jedne od osnovnih zamisli autora – potpune kompatibilnosti na nivou izvornog koda, ANSI (American National Standards Institute) je započeo 1983. godine rad na standardizaciji jezika. Svrha je bila predložiti nedvosmislenu i od računala neovisnu definiciju C jezika. Krajem 1988. godine rad je završen i objavljen je ANSI standard C jezika. ANSI komitet je tijekom vremena unaprijeđivao prvobitnu definiciju standarda. Danas je C visoko standardiziran jezik, čije gotovo sve implementacije podržavaju ANSI standard uz dodatne biblioteke funkcija karakteristične za pojedinu implementaciju.

2.2.2. Sintaksa programskog jezika C

C je programski jezik slobodnog formata, koji ne propisuje bilo kakva pravila koja bi se odnosila na stil pisanja. Naredbe mogu početi u bilo kojem stupcu reda, a između njih se mogu umetniti i prazni redovi ako ih programmer želi koristiti za odvajanje pojedinih cjelina programa. U istom redu moguće je napisati nekoliko naredbi odvojenih točka-zarezom, npr.

```
int i,n; printf("Unesite broj elemenata:"); scanf ("%d", &n);
```

Iako C omogućuje programerima veliku slobodu strukturiranja naredbi programa, ipak je prvenstveni cilj napisati čitav program. Posebno se to odnosi na ljude koji sudjeluju u timskom radu i čije programe pregledavaju drugi ljudi.

Sve ključne riječi i naredbe u C-u se pišu obavezno malim slovom. Velika i mala slova u imenima se razlikuju, ali se mogu ravnopravno koristiti (npr. imena varijable Sum, SUM ili sum). Uobičajena je praksa da se imena (varijabli, funkcija) pišu malim slovom, a simboličke konstante velikim.

Svi C programi moraju sadržavati sljedeće linije:

```
main ()
{
}
```

Sekcija programa koja počinje s main() i otvorenom vitičastom zagradom, a završava sa zatvorenom vitičastom zagradom zove se main, odnosno glavna funkcija. U svakom C programu mora postojati jedna funkcija koja se zove main, a predstavlja mjesto od kojeg počinje izvođenje programa. Naredba iza prve otvorene vitičaste zagrade u main funkciji je prva naredba koja se izvodi.

Dio programa između dvaju vitičastih zagrada naziva se blok. U primjeru program se sastoji od dvije funkcije ili dva bloka. Jedan blok se može nalaziti unutar drugoga.

Svaka izvršna naredba u C-u mora završavati točka zarezom (;). Komentari u C-u pišu se unutar /* i */ znakova i ne moraju završavati s točka-zarezom jer se ne radi o izvršnim naredbama, već naredbama koje definiraju funkciju, njen početak i kraj.

Čitljivost programa povećavaju komentari. Mogu se protezati preko nekoliko linija, zauzeti cijelu liniju ili se nalaziti iza izvršne naredbe u istom redu. Preporučuje se svrsishodno korištenje komentara, tj. ne treba pretjerati u njihovom korištenju. Dobra praksa je komentirati ukratko svaku funkciju pomoću uvodnog komentara, te kritične naredbe unutar funkcija, a ne svaki redu u programu. Npr. treba izbjegavati komentiranje naredbi sličnih sljedećoj:

```
printf (“Unesi n: “);    / *Ispisuje na zaslonu: Unesi n:   */
```

Također dobra preporuka je komentirati program za vrijeme njegovog pisanja, a ne poslije. Jedan komentar se ne može koristiti unutar drugog (ne dozvoljava se ugnježđivanje komentara, npr.

```
/* definicija sume /* funkcije za izračunavanje sume parnih brojeva */*/
```

Prva tri reda programa su komentari s nazivom programa i kratkim objašnjenjem njegove svrhe. C prema ANSI standardu sadrži 32 ključne riječi:

Tablica 1. Ključne riječi ANSI standard

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Izvor: *http://www.progtutorial.com/c-programming/images/keywords.jpg*

Gornje riječi i sintaksna pravila čine osnovnu definiciju jezika. Međutim, s njom bi se malo toga moglo napraviti jer ne sadrži npr. nijednu ulazno-izlaznu naredbu. Zbog toga uz svaki C kompilator obavezno dolazi standardna biblioteka funkcija koja sadrži funkcije za izvršavanje ulazno-izlaznih operacija, složenih matematičkih operacija, manipuliranje stringovima, slobodnom memorijom i dr.

U C-u sve varijable moraju biti deklarirane prije početka njihova korištenja. Deklaracije varijabli predstavljaju opis varijabli koji se nalazi najčešće na početku funkcije ili bloka naredbi:

sastoji se od tipa podatka koji može sadržavati i liste jednoznačnih imena odvojenih zarezom koja završavaju točka-zarezom.

Za vrijeme izvođenja programa varijabla može poprimiti različite vrijednosti odgovarajućeg tipa. C poznaje samo 4 osnovna tipa: cjelobrojni (int), znakovni (char) i dva realna (float i double).

Konstante za razliku od varijabli nikad ne mijenjaju svoju vrijednost. C dozvoljava nekoliko vrsta konstanti: brojeve (cjelobrojne i realne), znakove i nizove znakove ili stringove (riječi i fraze).

2.2.3. Primjer programa u C-u

Na sljedećim stranicama prikazani su primjeri kratkih programa pisanih u C programskom jeziku radi lakšeg razumijevanja sintakse, te logike programa.

Trivijalni i najpoznatiji primjer programa u C-u je ispis “Hello world”.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello World");

}
```

Kao sljedeći primjer prikazan je program koji zbraja određeni broj elemenata polja koji se unosi pomoću tipkovnice, te ispisuje elemente i njihovu sumu.

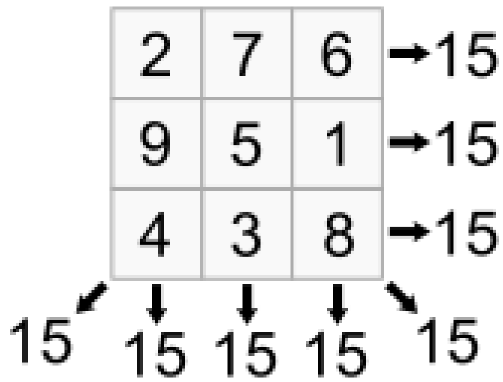
```
#include<stdio.h>
#include<stdlib.h>

#define MAXN 10
float suma(int n);
float data[MAXN]={234.56, -82.76, 324.89, 9323.7, 3.56, -2345.67, 45.97,
12341.4, -298.32, 70.5 };
main()
{
    int i,n;
    float sum;

printf(„Unesite broj elemenata: ");
scanf(„%d, &n");
if(n>MAXN||n<1)
{
printf(„\n\n Pogreska:nedozvoljen broj elemenata\n");
exit(1);
}
else
Sum=suma(n);
for(i=0;i<n;i++)
printf(„\n%9.2f",data[i]);
printf("\n-------\n");
printf(„%9.2f\n", sum);
return 0;
}
Float suma(int n)
{
int i;
float s=0;
for(i=0;i<n;i++)
s+=data[i];
return(s);
}
```

3. MAGIČNI KVADRAT

Magični kvadrat je kvadrat u čija su polja upisani brojevi tako da su zbrojevi po retcima, stupcima i dijagonalama međusobno jednaki (Slika 3.1). Zbroj stupaca, redova i dijagonala naziva se *magični zbroj* ili *magična konstanta*. Magični kvadrati postoje za svako M > 2. Postoje magični kvadrati svih dimenzija osim dimenzije 2×2. Magični kvadrat dimenzije 1 sastoji se samo od jednog kvadrata.

Matematička definicija čarobne četvorine reda n je kvadratna tablica n × n ispunjena cijelim brojevima 1, 2, 3..., n2, tako da je zbroj brojeva svakog retka, svakog stupca i obiju dijagonala jednak. Odgovarajuća formula glasi ovako:

$$S_n = \frac{n(1+n^2)}{2}$$

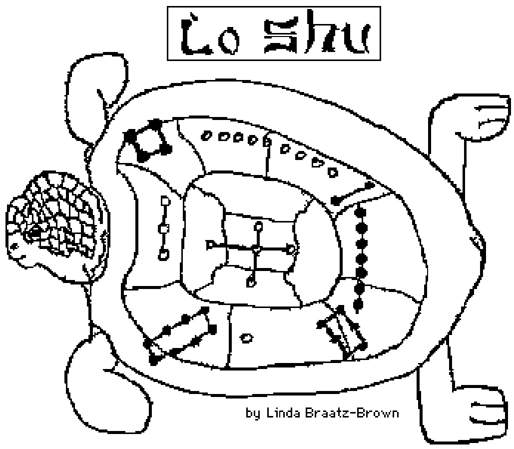


Slika 3.1. Magični kvadrat 3.reda (*http://bit.ly/1Ymd69o*)

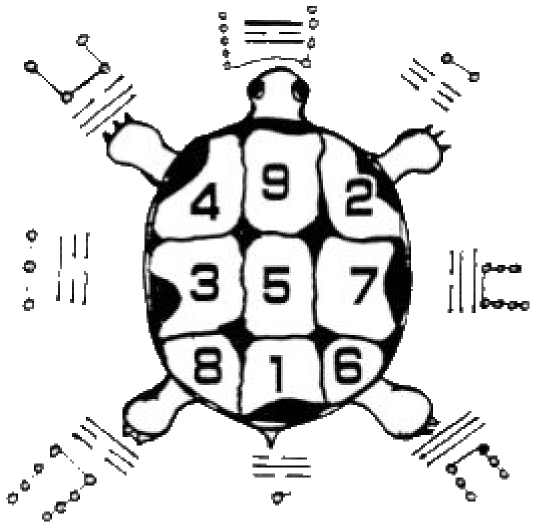
3.1. Uvod u magične kvadrate

Povijest magičnih kvadrata poznata je od davnina. U početku su im davana magična ili religijska svojstva, no danas su zanimljivi zbog svojih rješenja. Vjeruje se da su prvi magični kvadrati bili poznati u drevnoj Kini oko 2000 godina prije Krista.

Kako nema konkretnih pisanih dokaza kada i kako su otkriveni, njihova povijest vezana je uz jednu kinesku legendu. Legenda govori o ljudima pored rijeke koji su pokušali prinijeti žrtvu bogu rijeke Lo gdje se uvijek pojavljivala kornjača. Kornjača je prolazila pored darova svaki put kad bi ljudi prinosili žrtvu, te se to ponavljalo sve dok ljudi nisu primjetili čudan obrazac na leđima kornjače, odnosno magični kvadrat koji je poznat kao Lo Shu. Kornjača je na oklopu imala ispisane neke znakove. Ti znakovi su kineskom majstoru dali ključ za razumijevanje ritmova prirode. Uz pomoć modernih brojeva dobije se magični kvadrat dimenzije 3, čiji je zbroj 15 što u biti označava broj poklona koje treba žrtvovati da bi bog bio zadovoljan. Kinezi su tom kvadratu pripisivali magična svojstva, pa su oko vrata nosili magične kvadrate da ne bi oboljeli ili prosily. Slika oklopa kornjače prikazana je na slikama 3.2,3.3.



Slika 3.2. Lo Shu kvadrat na oklopu kornjače
(*http://mathforum.org/alejandre/magic.square/loshu.drawing.html*)



Slika 3.3. Brojevi ispisani na oklopu kornjače (Lo Shu) (*http://eljunktudatosan.hu/wp-content/uploads/2014/03/Lo-Shu.gif*)

Iz Kine su se magični kvadrati raširili do Indije gdje je moguće naći opise magičnih kvadrata 4×4 u prvim stoljećima poslije Krista. Kasnije se magični kvadrati šire diljem arapskog svijeta. Za njega je (gotovo sigurno) znao i Konfucije u 5. stoljeću prije nove ere, te Fibonacci koji je s divljenjem opisivao kvadrat tablicu razmjera 3×3 ispunjenu brojevima 1, 2, 3..., 9 u svom djelu Liber Abaci. Magični kvadrat spominje se u grčkim zapisima iz 1300. godine prije Krista, a u svojim radovima iz 130. godine poslije Krista opisuje ga i Theon iz Smyrne. U 9. stoljeću arapski astronomi koristili su ih za zapisivanje horoskopa. U 11. st. spominje ga arapski pjesnik, filozof i astronom Abraham ben Ezra. Njemački fizičar i teolog iz 16. stoljeća Heinrich Cornelius Agrippa načinio je 9 magičnih kvadrata od 3. do 9. reda i svakom je pridružio po jedan od 7 tada poznatih planeta Sunčevog sustava (Slika 3.4).

Saturn=15		
4	9	2
3	5	7
8	1	6

Jupiter=34			
4	14	15	1
9	7	6	12
5	11	10	8
16	2	3	13

Mars=65				
11	24	7	20	3
4	12	25	8	16
17	5	13	21	9
10	18	1	14	22
23	6	19	2	15

Sol=111					
6	32	3	34	35	1
7	11	27	28	8	30
19	14	16	15	23	24
18	20	22	21	17	13
25	29	10	9	26	12
36	5	33	4	2	31

Venus=175						
22	47	16	41	10	35	4
5	23	48	17	42	11	29
30	6	24	49	18	36	12
13	31	7	25	43	19	37
38	14	32	1	26	44	20
21	39	8	33	2	27	45
46	15	40	9	34	3	28

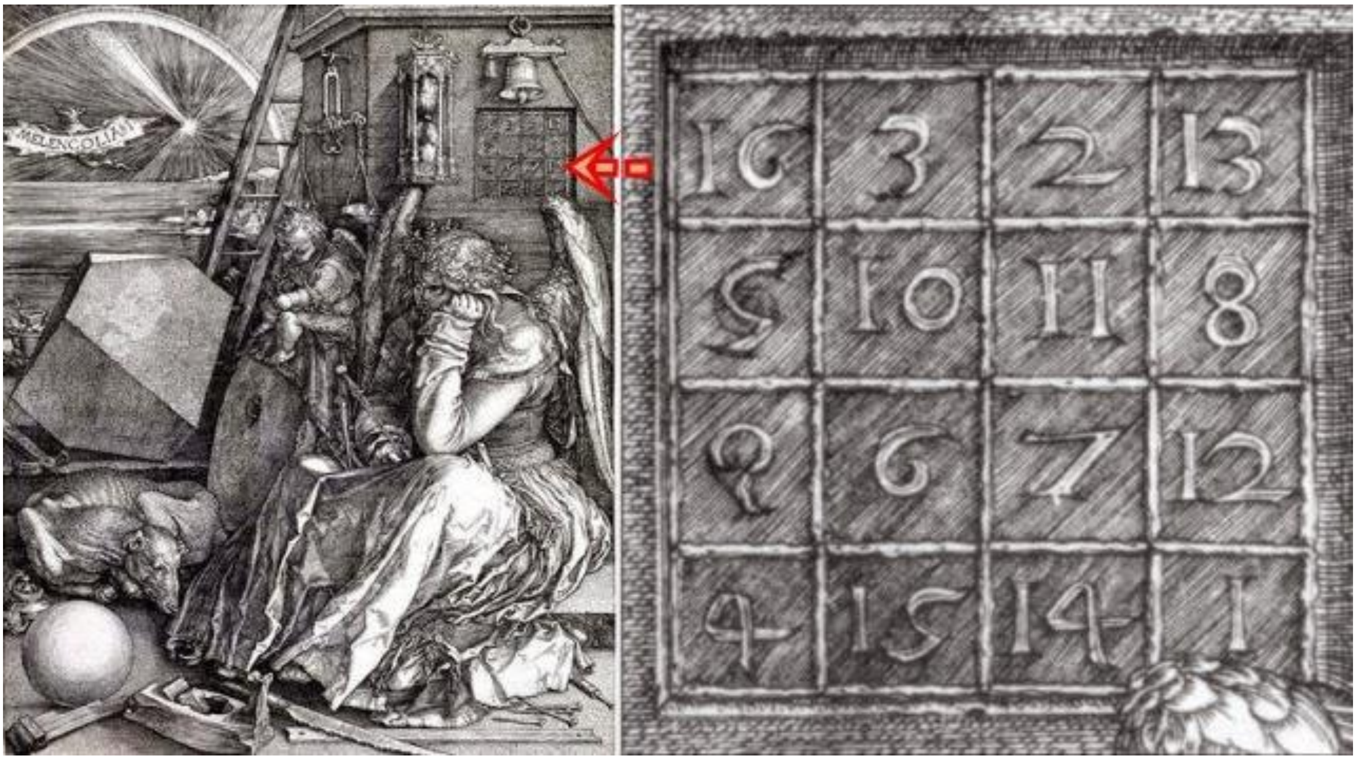
Mercury=260							
8	58	59	5	4	62	63	1
49	15	14	52	53	11	10	56
41	23	22	44	45	19	18	48
32	34	35	29	28	38	39	25
40	26	27	37	36	30	31	33
17	47	46	20	21	43	42	24
9	55	54	12	13	51	50	16
64	2	3	61	60	6	7	57

Luna=369								
37	78	29	70	21	62	13	54	5
6	38	79	30	71	22	63	14	46
47	7	39	80	31	72	23	55	15
16	48	8	40	81	32	64	24	56
57	17	49	9	41	73	33	65	25
26	58	18	50	1	42	74	34	66
67	27	59	10	51	2	43	75	35
36	68	19	60	11	52	3	44	76
77	28	69	20	61	12	53	4	45

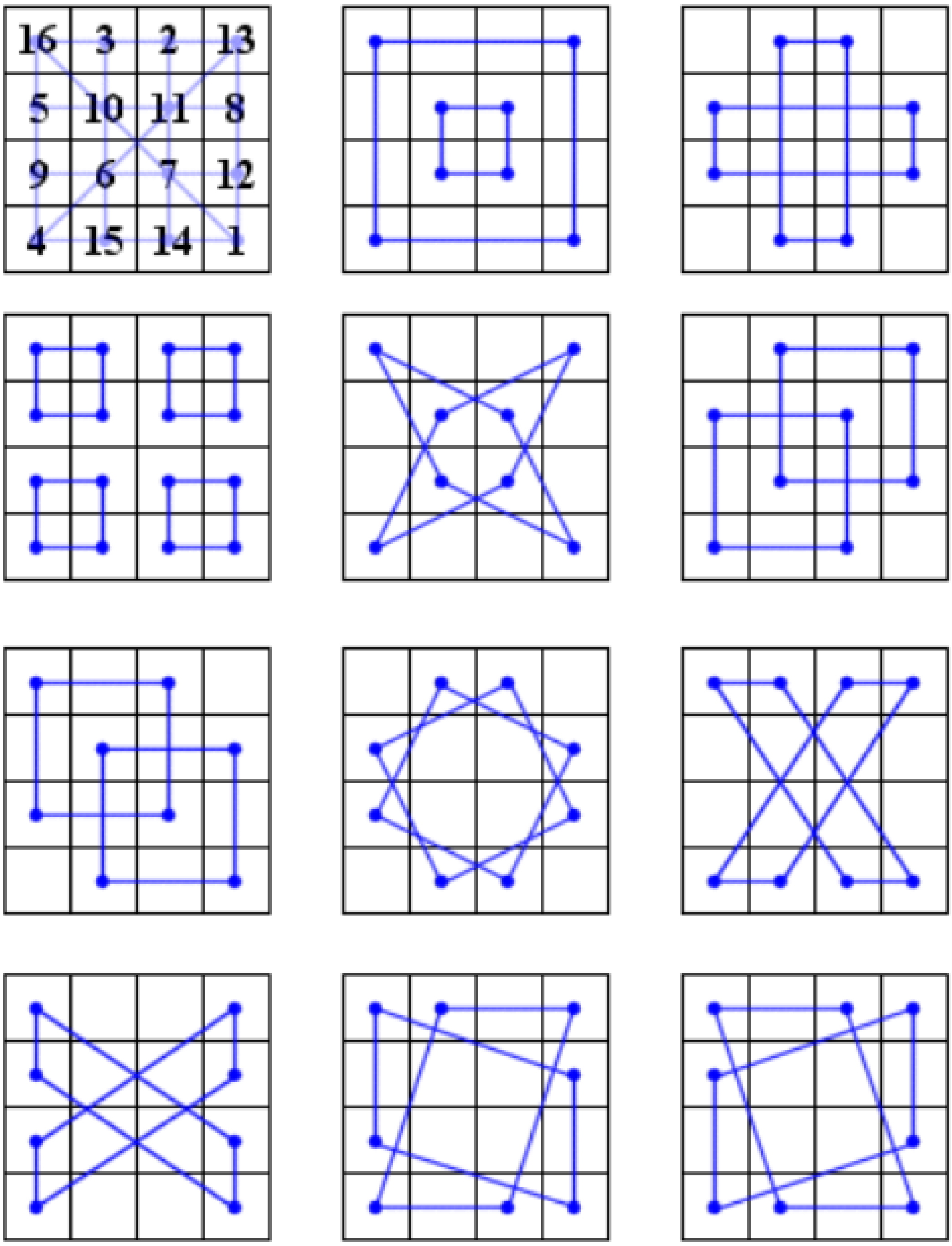
Slika 3.4. Magični kvadrati Corneliusa Agrippa pridodjeljeni planetama (*http://bit.ly/2cHkiwC*)

S vremenom magični kvadrati dolaze i do Europe. Najpoznatiji magični kvadrat četvrtog reda, moguće je vidjeti na djelu Melancolia I, gravuri njemačkog umjetnika Albrechta Dürera

(Slika 3.5). Djelo je nastalo 1514. godine što je vidljivo u srednjim kvadratićima zadnjeg reda. Magični zbroj je 34 i taj rezultat se pored zbroja u redovima, stupcima i dijagonalama može pronaći i na drugim mjestima u kvadratu. Vidljivo je da četiri kvadratića u sredini daju zbroj 34, četiri kvadratića u kutovima daju zbroj 34, prva dva kvadratića u prvom redu i zadnja dva u četvrtom redu daju zbroj 34, te prva dva kvadratića u prvom stupcu i zadnja dva u četvrtom stupcu daju zbroj 34. Može se primijetiti i da je zbroj obje dijagonale zajedno, jednak zbroju svih ostalih kvadratića. Također je zbroj svih kvadratića u dvije dijagonale jednak zbroju kvadrata ostalih brojeva, a isto vrijedi i za kubove. Isto tako zbroj kvadriranih brojeva u dva gornja reda je jednak zbroju kvadrata za donja dva reda. Isto vrijedi i za stupce 1 i 2 nasuprot stupaca 3 i 4. Svi zbrojevi, odnosno rješenja koja daju zbroj 34 prikazana su na slici 3.6. Rezbarije Albrechta Dürera su zagonetke, višeslojna djela umjetnosti koja se mogu interpretirati na različite načine u isto vrijeme. Neka shvaćanja vezana su uz povijesne činjenice, neka uz društvene okolnosti onoga vremena, a neka otkrivaju specifično briljantan um umjetnika onoga vremena.

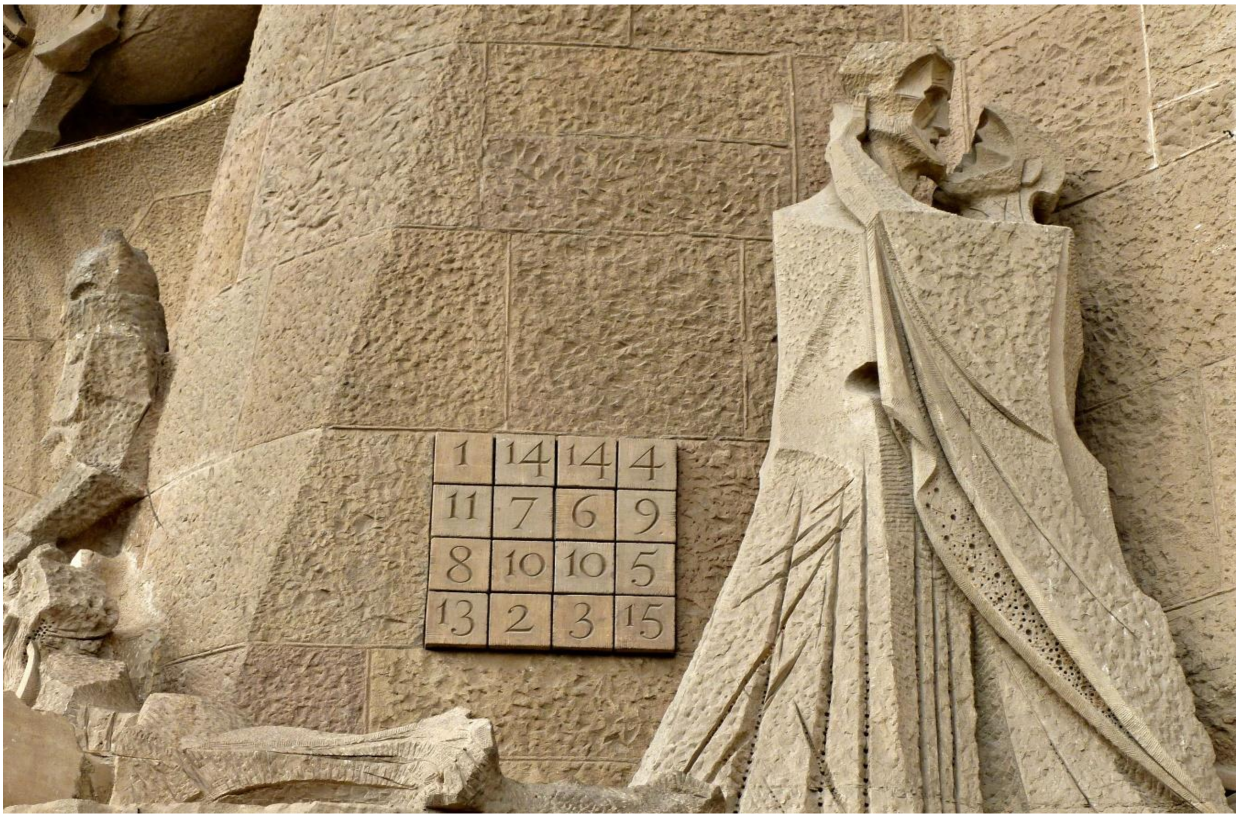


Slika 3.5. Najpoznatiji magični kvadrat (Melancolia I, Albrecht Dürer) (*http://bit.ly/1U4jxJ2*)



Slika 3.6. Sva rješenja Dürerovog ‘supermagičnog’ kvadrata (zbroj 34) (https://s-media-cache-ak0.pinimg.com/236x/f5/ae/73/f5ae73c73b41779a4c41b251bba0d9d3.jpg)

Manje poznat primjer magičnog kvadrata u umjetnosti može se vidjeti na katedrali La Sagrada Família u Barceloni (Slika 3.7). Ovaj kvadrat oblikom liči na Dürerov, no magični zbroj je 33, a taj broj označava starost Isusa kada je umro na križu.



Slika 3.7. Magični kvadrat na katedrali La Sagrada Família, Barcelona (*http://bit.ly/25Y2qQR*)

Prvi detaljniji rad na temu magičnih kvadrata objavio je francuski matematičar F. Bernard de Bessy 1693. godine. On je pokazao da postoji 880 magičnih kvadrata 4. Reda, te ih je sve ispisao. Zanimljivo je da je najveći "ručno" izračunat magični kvadrat dimenzije 1111 × 1111. Izračunao ga je Norbert Behnke (Njemačka). Magičnim kvadratima bavili su se iz hobija i mnogi uglednici. Jedan od njih je i Benjamin Franklin (američki državnik, filozof, izumitelj, fizičar, ekonomist i pisac). On se 1736. godine "zabavljao" konstrukcijom magičnih kvadrata, te je u svojim radovima objavio dva magična kvadrata 8. reda (Slika 3.8).

52	61	4	13	20	29	36	45
14	3	62	51	46	35	30	19
53	60	5	12	21	28	37	44
11	6	59	54	43	38	27	22
55	58	7	10	23	26	39	42
9	8	57	56	41	40	25	24
50	63	2	15	18	31	34	47
16	1	64	49	48	33	32	17

Slika 3.8. Magični kvadrat Benjamina Franklina
(*http://mathforum.org/alejandre/magic.square/franklin2.gif*)

Nakon pojave računala magični kvadrati više nisu "neriješena misterija", iako se još uvijek ne može točno odrediti koliko ih je za svaki n. Tako je uz pomoć računala 1973. godine dokazano da postoji 275305224 magičnih kvadrata 5. reda. Broj kvadrata dimenzije 6×6 još nije točno određen, ali se procjenjuje da ih postoji $(1.7745 \pm 0.0016) \times 1019$.

3.2. Vrste magičnih kvadrata

Pri sastavljanju kvadrata koriste se dva niza brojeva: prvi niz je 1, 2, 3. . . , n, a drugi je 0, n, 2n, n(n−1). Prilikom ove metode treba razlikovati slijedeća četiri slučaja. Ako je n neparan i nije djeljiv s 3, ako je n neparan i djeljiv je s 3, zatim n je djeljiv s 4, te je n paran i nije djeljiv s 4. Za magični kvadrat dimenzije *n* kaže se da je pravi, ako su u njemu svi brojevi od 1 do n^2. Prikazani su primjeri magičnih kvadrata trećeg reda (Slika 3.9).

2	7	6
9	5	1
4	3	8

2	9	4
7	5	3
6	1	8

8	3	4
1	5	9
6	7	2

6	7	2
1	5	9
8	3	4

Slika 3.9. Primjer magičnih kvadrata reda 3

(*http://www.math.uniri.hr/~ajurasic/magicni%20kvadrati.pdf*)

Svi ovi magični kvadrati zapravo su isti jer se iz jednoga od njih drugi dobiju zrcaljenjima i zakretanjima. Zbroj svakog retka, svakog stupca i obiju dijagonala jednak je magičnoj sumi 15. Dakle, magična suma reda 3 je M(3) = 15. Poistovjećujemo magične kvadrate koji se jedan iz drugog mogu konstruirati zrcaljenjem ili rotacijom (Slika 3.10).

2	7	6
9	5	1
4	3	8

~

2	9	4
7	5	3
6	1	8

Slika 3.10. Magični kvadrati dobiveni rotacijom

(*http://www.mathos.unios.hr/~middlemath/ppt/magicni-web.ppt*)

Nije poznata opća ovisnost broja različitih magičnih kvadrata o redu kvadrata. Reda 1 i 3 su jedinstveni i to je odavno poznato; reda 4 ih ima 880 (de Bessy, 1693.), reda 5 ih ima 275 305 224 (Schroeppel, 1973.); reda 6 ih ima reda veličine 10^{19} (Pinn &Wieczerkowski,1998., Monte Carlo simulacije i metode statističke mehanike).

Postoje posebne vrste magičnih kvadrata. Ako jedan ili oba zbroja dijagonala nisu jednaki zbrojevima redaka i stupaca kvadrat je polumagičan. Povezani kvadrati su oni čiji su zbrojevi centralno simetričnih polja jednaki $n^2 + 1$ (npr. *Lo Shu*). Zatim panmagični (pandijagonalni, vražji, Nasik): sve (uključivo i prelomljene) dijagonale imaju magičnu sumu – ne postoje reda 3 ni reda $4k + 2$, te Polu-Nasik: nasuprotne kratke dijagonale imaju magičnu sumu ($14 + 4 + 11 + 5 = 34$, $12 + 6 + 13 + 3 = 34$). Panmagične kvadrate se može rasporediti u beskonačnu mrežu (ravnina) i svaki podkvadrat veličine osnovnog bit će panmagičan. Od 880 magičnih kvadrata reda 4, njih 448 su obični, tj. zadovoljavaju samo temeljna svojstva magičnosti, njih 48 su panmagični, a njih 384 su polu-Nasik (uključivo povezanih).

Najsavršeniji magični kvadrat je panmagični i svaki 2×2-podkvadrat ima isti zbroj ($2n^2 - 2$) – svi panmagični reda 4 su takvi, dok su kod antimagičnog kvadrata svi retci, stupci i dijagonale različitih suma i sume čine niz uzastopnih prirodnih brojeva. Temeljeni su na oduzimanju, množenju ili dijeljenju.

4. RJEŠAVANJE DIPLOMSKOG ZADATKA

4.1. Algoritmi za rješavanje magičnih kvadrata

Danas su razvijeni brojni algoritmi za rješavanje magičnih kvadrata. Različito je rješavanje za kvadrate parnih dimenzija i one neparnih dimenzija. Rješavanje je prikazano u sljedećim poglavljima (4.1. i 4.2.).

4.1.1. Kvadrati neparnih dimenzija

Za konstruiranje magičnih kvadrata neparnog reda, jedna od najpoznatijih metoda je Loub´ere-ova metoda. Počinjemo s okvirom praznog kvadrata dimenzije $n \times n$ kojeg ćemo ispuniti brojevima od 1 do n^2 po sljedećim pravilima:

1.) Prvo se napiše broj 1 u sredini prvog reda.

2.) Svaki sljedeći broj stavljamo na mjesto dijagonalno gore-desno. Tako će se brojevi upisivati dijagonalno prema gore, desno od svog prethodnika.

3.) Ako smo došli do desnog ruba, sljedeći broj ćemo upisati na mjesto u krajnju lijevu kolonu i jedan red iznad

4.) Ako smo došli do gornjeg ruba, sljedeći broj ćemo smjestiti na mjesto u najdonji red i jedan stupac desno

5.) Ako je mjesto u kvadratu zauzeto, npr. ako je mjesto gdje želimo staviti broj k+1 zauzeto, onda ćemo broj k+1 smjestiti direktno ispod broja k u kvadratu

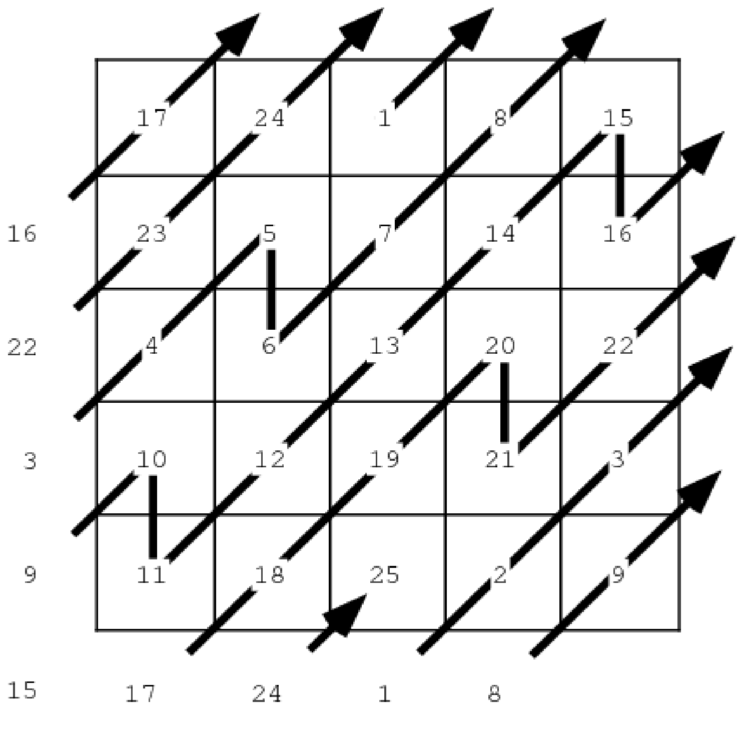
Ova metoda uspijeva uvijek kada se jedinica upiše u sredinu najgornjeg reda. Ispod je magični kvadrat 5-te dimenzije generiran na taj način. Moguće je također koristiti druge nizove koji se razlikuju od normalnog $1 \rightarrow n^2$, samo ako je razlika između brojeva ista.

		1		
	5			
4	6			
				3
			2	

Slika 4.1. Konstrukcija magičnog kvadrata

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Slika 4.2. Popunjeni magični kvadrat



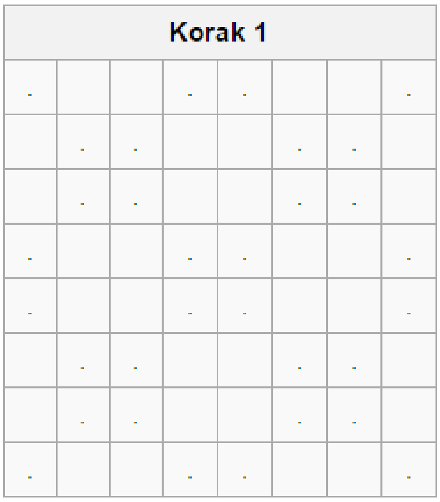
Slika 4.3. Popunjavanje neparne matrice (http://mathworld.wolfram.com/images/eps-gif/MagicSquareSiamese_1000.gif)

4.1.2. Kvadrati duple parne dimenzije

Pod parnom dimenzijom misli se na kvadrate koji su djeljivi sa 4. Ovdje spadaju magični kvadrati dimenzija 4, 8, 12, 16 itd. Tehnika za ovaj algoritam je vrlo zabavna i nije teška. Može se podijeliti u tri koraka koja su opisana tri koraka jednog kvadrata 8x8.

Prvi korak: Ucrtavaju se točkice po obrascima u kvadrat. Točkice trebaju napraviti mini kvadrate i ličiti na šahovnicu. Najlakši način može biti upisujući od malih kvadrata u sredini koji se lako pronađe na mjestu gdje se susreću dijagonale.

Drugi korak: mali kvadrati se sada upisuju unutar velikog kvadrata, tako da cijelo vrijeme upisujemo kut nasuprot kuta, ne bi li se dobio obrazac. U sljedećim koracima se upisuju brojevi od 1 do n^2.



Slika 4.4. Rješavanje magičnih kvadrata (*https://hr.wikipedia.org/wiki/Magični_kvadrat*)

Treći korak: Upisuju se prvi brojevi u kvadrat. Brojevi se počinju upisivati jedan za drugim započevši s prvim redom, s lijeva na desno a zatim se prelazi u red ispod. Brojevi se upisuju samo na mjesta gdje su upisane točkice u koraku 2.

Korak 2							
1			4	5			8
	10	11			14	15	
	18	19			22	23	
25			28	29			32
33			36	37			40
	42	43			46	47	
	50	51			54	55	
57			60	61			64

Slika 4.5. Rješavanje magičnih kvadrata (*https://hr.wikipedia.org/wiki/Magični_kvadrat*)

Četvrti korak: Na kraju se uspisuju ostali brojevi. Oni se upisuju na isti način kao u drugom koraku, no obrnutim redom. Započinju se upisivati u zadnjem redu, s desna na lijevo, a zatim se prelazi u red iznad. Kao i u drugom koraku opet se upisuju brojevi od 1 do 64, ali u prazna mjesta.

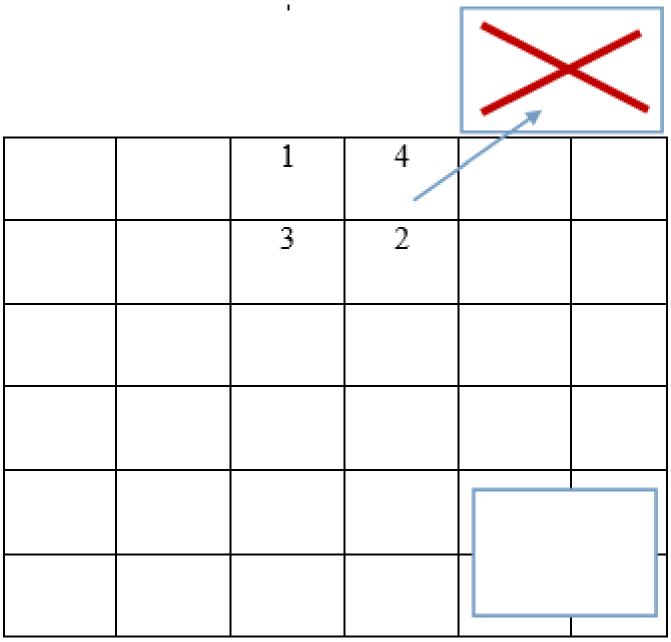
Korak 3							
1	63	62	4	5	59	58	8
56	10	11	53	52	14	15	49
48	18	19	45	44	22	23	41
25	39	38	28	29	35	34	32
33	31	30	36	37	27	26	40
24	42	43	21	20	46	47	17
16	50	51	13	12	54	55	9
57	7	6	60	61	3	2	64

Slika 4.6. Rješavanje magičnih kvadrata (*https://hr.wikipedia.org/wiki/Magični_kvadrat*)

4.1.3. Kvadrati dimenzije nedjeljive s 4

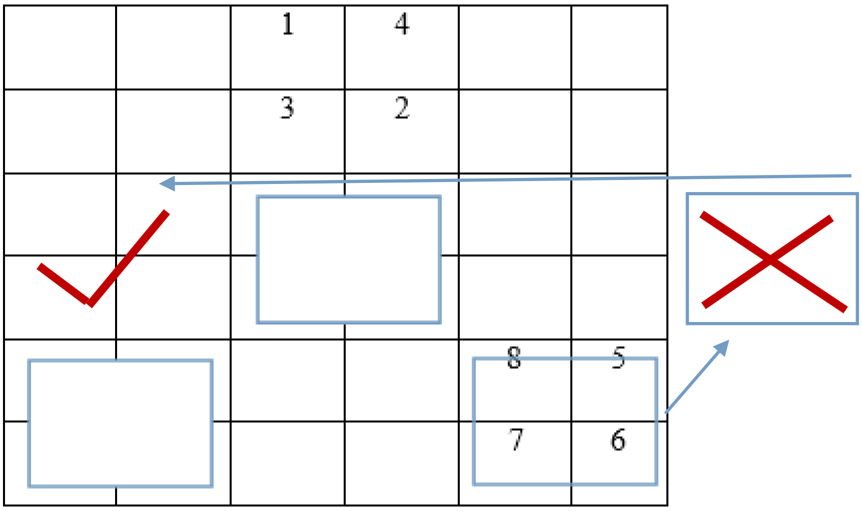
Ova vrsta kvadrata je parna, no nisu djeljivi s 4. Kako bi se lakše riješila njihova problematika, postoje sljedeća pravila za njihovo rješavanje:

Prvi korak: Cijeli kvadrat popunjavamo tako da upisujemo po četiri broja (Slika 4.7). Upisujemo ih po dijagonali, odnosno kao da ispisujemo slovo ‘X’. Nakon popunjavanja pomičemo se gore-desno, te ako prelazimo granicu kvadrata potrebno je spustiti kvadrat po stupcu ili retku prema dolje (desno ili lijevo) u prvi slobodni kvadratić.



Slika 4.7. Prvi korak popunjavanja tablice

Drugi korak: Kvadrate prikazane na slici 4.8 popunjavamo drugačije od ostalih. Oni se popunjavaju u obliku slova ‘U’.



Slika 4.8. Drugi korak popunjavanja matrice

Treći korak: Popunjavamo gornji desno kvadrat, no ako prelazi granicu kvadrata premještamo ga po stupcu ili redu kao na slici 4.9.

		1	4		
		3	2		
9	12				
11	10				
				8	5
				7	6

Slika 4.9. Treći korak popunjavanja tablice

Četvrti korak: Ako je kvadrat gore-desno popunjen, popunjavaju se četiri mjesta ispod prijašnje popunjenog kvadrata. Primjer je prikazan na slici 4.10.

		1	4		
		3			
9	12				
11	10				
16	13			8	5
15	14			7	6

Slika 4.10. Četvrti korak popunjavanja magičnog kvadrata

Peti korak: Nastavimo popunjavati matricu po pravilima dok ju ne ispišemo, pritom je potrebno pripaziti na upis brojeva (‘X’ ili ‘U’).

29	32	1	4	21	24
30	31	3	2	23	22
9	12	20	17	25	28
11	10	19	18	27	26
16	13	33	36	8	5
15	14	35	34	7	6

Slika 4.11. Peti korak popunjavanja magičnog kvadrata

4.3. Programski dio

Ideja zadatka je na temelju algoritama rješavanja magičnih kvadrata napisati program koji će za određenu veličinu matrice koju korisnik upisuje, ispisati kvadrat koji ima magična svojstva, odnosno kojemu su zbroj redaka, stupaca i dijagonala jednaki . Na samom početku programa od korisnika se traži da upiše elemente matrice koji ga zanimaju. Kako se matrica sastoji od stupaca i redaka, vrlo je važan redosljed pri kojem upisujemo članove, Matrica se popunjava po redovima, počevši od prvoga ka zadnjem. Nakon upisa elemenata matrice, ona se ispisuje na zaslon, te se na temelju popunjenih polja unutar matrice provjerava odgovaraju li sume redaka, stupaca i dijagonala pravilu magičnog kvadrata, odnosno jesu li sume navedenih iste. Ako jesu, program ispisuje da je zadana matrica čarobni kvadrat, a ako sume ne odgovaraju, program ispisuje da zadana matrica nije magični kvadrat.

4.4. Analiza programskog djela

Zadatak rada je na temelju unešene veličine matrice, ispisati magični kvadrat. Rješavanje toga zadatka i pisanje programskog koda zahtjevao je učenje rješavanja magičnih kvadrata. Postoji nekoliko načina rješavanja spomenutog problema, a svaki ovisi o broju koji će na početku programa biti upisan od strane korisnika. Dakle, po pravilima magičnog kvadrata postoje tri vrste racunanja:

0) stranice su jednake duljine (minimum 3x3)

1) stranice imaju neparan broj polja (3x3, 5x5, 7x7,...)

2) stranice imaju paran broj polja, ali broj polja nije djeljiv sa 4 (6x6, 10x10, 14x14,...)

3) stranice imaju paran broj polja i broj polja je djeljiv sa 4 (4x4, 8x8, 12x12,...).

Svaki od ovih slučaja potrebno je uzeti u obzir prilikom rješavanja programskog zadatka.

Prije pisanja glavnog djela koda, program započinje zaglavljem u kojem su definirane biblioteke (stdio.h, stdlib.h) koje služe za korištenje ugrađenih funkcija. Biblioteka <stdlib.h> korištena je zbog funkcije atoi() koja je potrebna da prima string i konvertira ga u integer. U ovom djelu definiran je i maksimalan broj za koji će se program izvršavati, a to je 99. Zatim, definirane su varijable koje će se koristiti kasnije u programu.

U int main() funkciji deklarirane su četiri varijable. Mod, flag, matrix[MAX][MAX] koji predstavlja matricu, to jest dvodimenzionalni niz. Size sadrzi veličinu dimenzije, te su oboje int tipa. Nakon prvog unosa broja, od korisnika se očekuje da pokuša ponovno s nekim drugim brojem ili da izađe iz programa. Pomoću flaga određuje se ponavlja li se program ili ne, dok mod[20] prima korisnikov unos željenoga broja. Flag[1] prima jedan znak i ako je taj znak 'n'(no) prekida s izvršavanjem programa. U suprotnom daje korisniku još jednu mogućnost za stvaranje novog kvadrata.

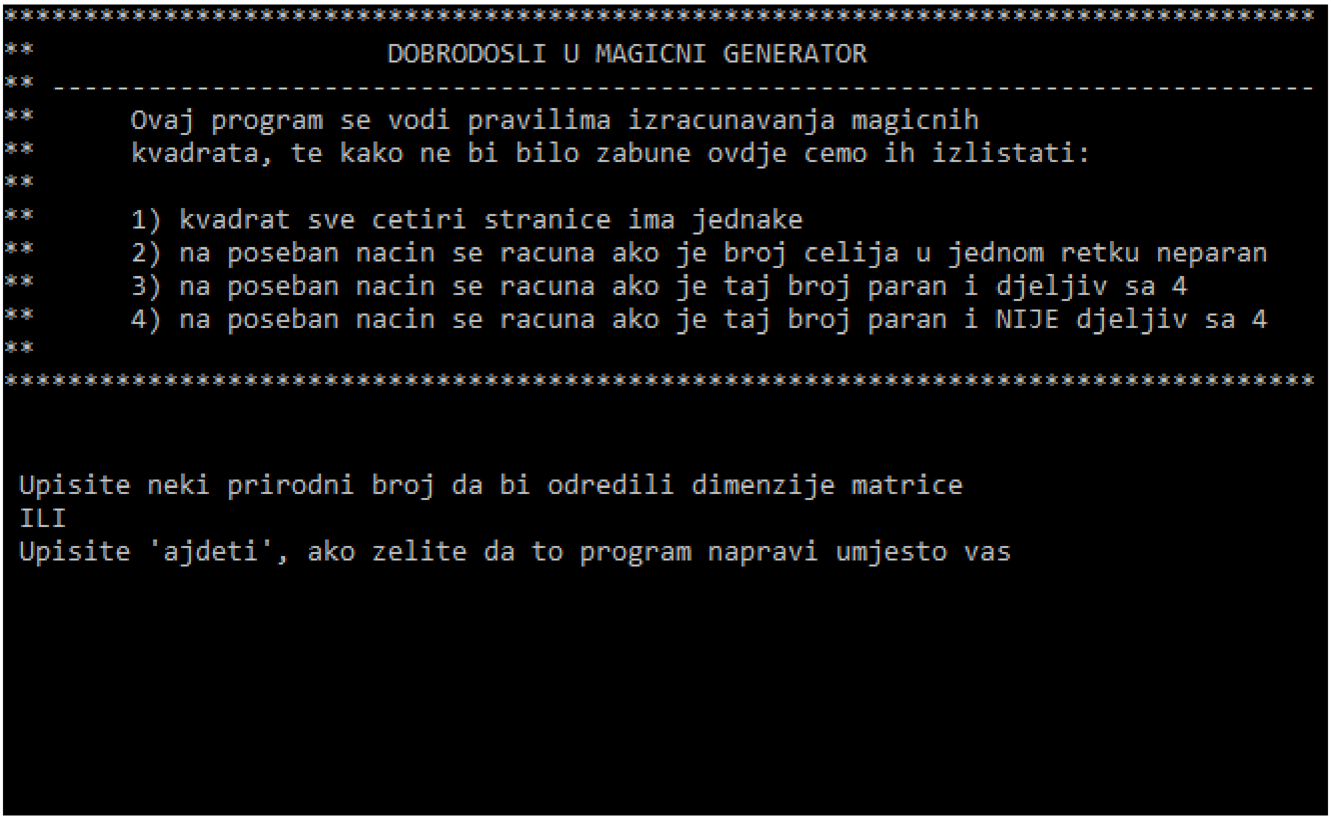
Odmah nakon deklaracije varijabli pokreće se funkcija naslovna( ). Ona se većinom sastoji od printf funkcija koje služe za prikazivanje teksta i vrijednosti na monitor. Naslovna() je void funkcija što znači da ne vrača nikakvu vrijednost već je koristimo samo za ispit na ekran. U toj funkciji zapisan je sav tekst koji vidimo pri otvaranju programa.

Ako korisnik izabere izraz '*ajdeti*' jer želi da program umjesto njega izabere broj, navedena radnja obavlja se preko funkcije rand(), koja je takoder iz <stdlib.h> biblioteke. Broj koji može biti nasumično izabran je unutar intervala <3,15>. Nisu se koristili veći brojevi isključivo zbog 'nepreglednosti' ispisa velikih matrica. Također je postavljen uvjet da magični kvadrat ne postoji za brojeve manje od 3. Ako korisnik nije upisao '*ajdeti*' onda se njegov unos preko funkcije atoi() pretvara u integer, te se provjerava je li manji od 3. Upiše li se neki drugi string, biti će preveden kao nula i opet će ispasti manji od 3. Isto vrijedi za broj 2 ili negativne brojeve. Ako dođe do takvoga neispravnoga unosa broja, funkcija exit() prekida rad aplikacije. Daljnji dijelovi programa odnose se na zapis suma unutar matrice, te računanje magičnosti za svaki slučaj pojedinačno (od četiri već navedena na početku poglavlja) što je prethodno objašnjeno u algoritmima rješavanja čarobnih četvorina.

4.5. Rad programskog djela

Program je napisan u programskom jeziku C, te je izrađen kao *Win32 console application*.

Pri pokretanju programa otvara se prozor kao na slici 4.12. Na samom početku objašnjeni su načini računanja 'magičnosti' kvadrata, te se od korisnika traži da upiše veličinu matrice za koju je potrebno ispisati magični kvadrat. Kako se radi o kvadratu, potrebno je upisati samo jedan broj. Upisani broj mora pripadati skupu prirodnih brojeva 1, 2, 3..., n^2. Ako korisnik nije siguran za koji broj želi ispis čarobne četvorine, omogućen mu je upis riječi *'ajdeti'*, nakon čega program nasumično izabire neki broj, te ispisuje matricu.

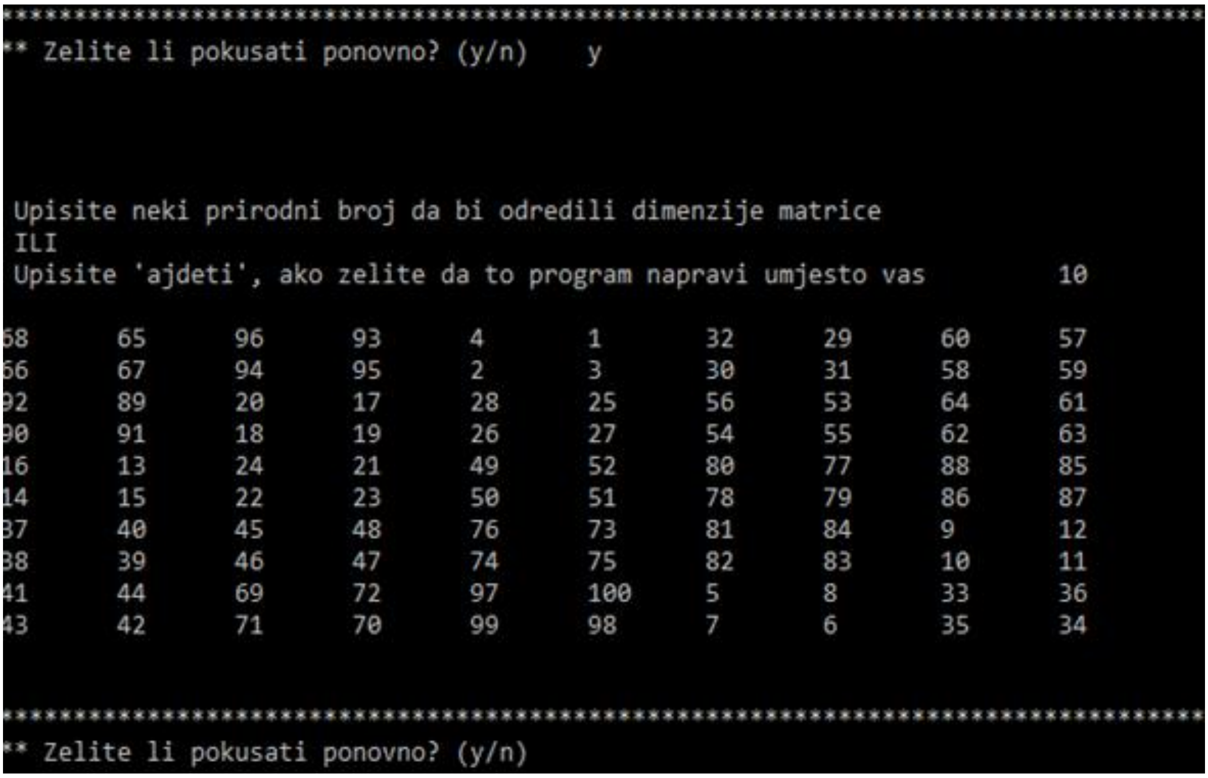


Slika 4.12. Unos veličine matrice

Za daljni rad programa potrebno je upisati broj ili izraz *'ajdeti'*. Ako je korisnik upisao broj, ispisuje se izračunata matrica te veličine. Za prvi primjer, upisan je broj 3 (Slika 4.13.), te za drugi primjer upisan je broj 10 (Slika 4.14.). No, ako je korisnik upisao *'ajdeti'*, tada se nasumičnim odabirom ispisuje matrica koja odgovara tom broju.

```
*******************************************************************************
**                     DOBRODOSLI U MAGICNI GENERATOR
** -----------------------------------------------------------------------------
**      Ovaj program se vodi pravilima izracunavanja magicnih
**      kvadrata, te kako ne bi bilo zabune ovdje cemo ih izlistati:
**
**      1) kvadrat sve cetiri stranice ima jednake
**      2) na poseban nacin se racuna ako je broj celija u jednom retku neparan
**      3) na poseban nacin se racuna ako je taj broj paran i djeljiv sa 4
**      4) na poseban nacin se racuna ako je taj broj paran i NIJE djeljiv sa 4
**
*******************************************************************************


 Upisite neki prirodni broj da bi odredili dimenzije matrice
 ILI
 Upisite 'ajdeti', ako zelite da to program napravi umjesto vas         3

8       1       6
3       5       7
4       9       2


*******************************************************************************
** Zelite li pokusati ponovno? (y/n)
```

Slika 4.13. Uneseni broj je 3/ Ispis magične matrice veličine 3



```
*******************************************************************************
** Zelite li pokusati ponovno? (y/n)    y



 Upisite neki prirodni broj da bi odredili dimenzije matrice
 ILI
 Upisite 'ajdeti', ako zelite da to program napravi umjesto vas         10

68      65      96      93      4       1       32      29      60      57
66      67      94      95      2       3       30      31      58      59
92      89      20      17      28      25      56      53      64      61
90      91      18      19      26      27      54      55      62      63
16      13      24      21      49      52      80      77      88      85
14      15      22      23      50      51      78      79      86      87
37      40      45      48      76      73      81      84      9       12
38      39      46      47      74      75      82      83      10      11
41      44      69      72      97      100     5       8       33      36
43      42      71      70      99      98      7       6       35      34


*******************************************************************************
** Zelite li pokusati ponovno? (y/n)
```

Slika 4.14. Uneseni broj je 10/ Ispis magične matrice veličine 10

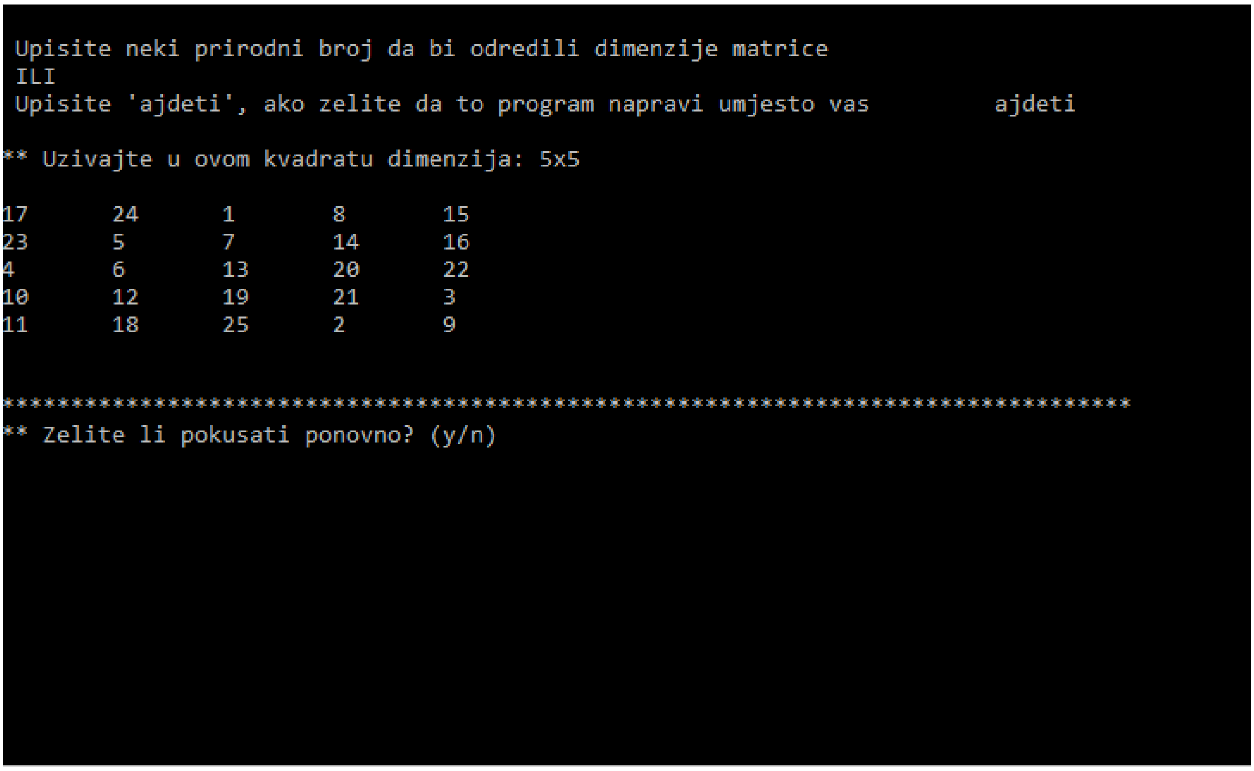
Najveći broj koji korisnik može upisati je 99, te izlistana matrica izgleda kao na slici 4.15. Primjeri ispisa matrica za upis izraza 'ajdeti' prikazani su na slikama 4.16. i 4.17. Na slici 4.16. nasumično je određen broj 14, dok je na slici 4.17 nasumično određen broj 5.

```
Upisite neki prirodni broj da bi odredili dimenzije matrice
ILI
Upisite 'ajdeti', ako zelite da to program napravi umjesto vas        99

4952    5053    5154    5255    5356    5457    5558    5659    5760    5861    5962    6063    6164    6265    6366
6467    6568    6669    6770    6871    6972    7073    7174    7275    7376    7477    7578    7679    7780    7881
7982    8083    8184    8285    8386    8487    8588    8689    8790    8891    8992    9093    9194    9295    9396
9497    9598    9699    9800    1       102     203     304     405     506     607     708     809     910     1011
1112    1213    1314    1415    1516    1617    1718    1819    1920    2021    2122    2223    2324    2425    2526
2627    2728    2829    2930    3031    3132    3233    3334    3435    3536    3637    3738    3839    3940    4041
4142    4243    4344    4445    4546    4647    4748    4849    4950
5052    5153    5254    5355    5456    5557    5658    5759    5860    5961    6062    6163    6264    6365    6466
6567    6668    6769    6870    6971    7072    7173    7274    7375    7476    7577    7678    7779    7880    7981
8082    8183    8284    8385    8486    8587    8688    8789    8890    8991    9092    9193    9294    9395    9496
9597    9698    9799    99      101     202     303     404     505     606     707     808     909     1010    1111
1212    1313    1414    1515    1616    1717    1818    1919    2020    2121    2222    2323    2424    2525    2626
2727    2828    2929    3030    3131    3232    3333    3434    3535    3636    3737    3838    3939    4040    4141
4242    4343    4444    4545    4646    4747    4848    4949    4951
5152    5253    5354    5455    5556    5657    5758    5859    5960    6061    6162    6263    6364    6465    6566
6667    6768    6869    6970    7071    7172    7273    7374    7475    7576    7677    7778    7879    7980    8081
8182    8283    8384    8485    8586    8687    8788    8889    8990    9091    9192    9293    9394    9495    9596
9697    9798    98      100     201     302     403     504     605     706     807     908     1009    1110    1211
1312    1413    1514    1615    1716    1817    1918    2019    2120    2221    2322    2423    2524    2625    2726
2827    2928    3029    3130    3231    3332    3433    3534    3635    3736    3837    3938    4039    4140    4241
4342    4443    4544    4645    4746    4847    4948    5049    5051
5252    5353    5454    5555    5656    5757    5858    5959    6060    6161    6262    6363    6464    6565    6666
6767    6868    6969    7070    7171    7272    7373    7474    7575    7676    7777    7878    7979    8080    8181
8282    8383    8484    8585    8686    8787    8888    8989    9090    9191    9292    9393    9494    9595    9696
9797    97      198     200     301     402     503     604     705     806     907     1008    1109    1210    1311
```

Slika 4.15. Uneseni broj je 99/ Ispis magične matrice veličine 99 (*matrica nije u potpunosti na slici)

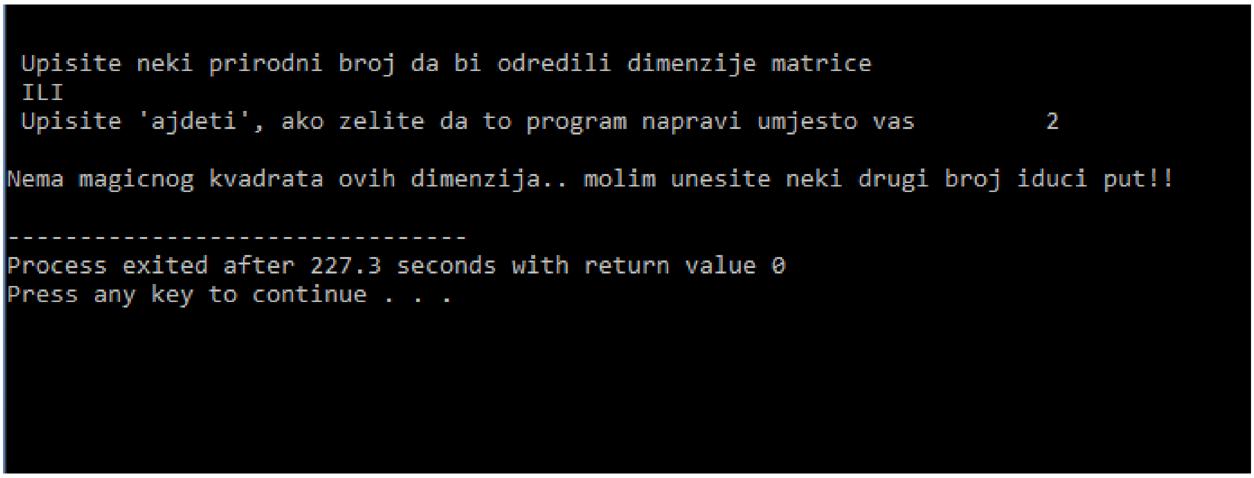
```
******************************************************************************


 Upisite neki prirodni broj da bi odredili dimenzije matrice
 ILI
 Upisite 'ajdeti', ako zelite da to program napravi umjesto vas        ajdeti

** Uzivajte u ovom kvadratu dimenzija: 14x14

120     117     156     153     192     189     4       1       40      37      76      73      112     109
118     119     154     155     190     191     2       3       38      39      74      75      110     111
152     149     188     185     28      25      36      33      72      69      108     105     116     113
150     151     186     187     26      27      34      35      70      71      106     107     114     115
184     181     24      21      32      29      68      65      104     101     140     137     148     145
182     183     22      23      30      31      66      67      102     103     138     139     146     147
20      17      56      53      64      61      97      100     136     133     144     141     180     177
18      19      54      55      62      63      98      99      134     135     142     143     178     179
49      52      57      60      93      96      132     129     165     168     173     176     13      16
50      51      58      59      94      95      130     131     166     167     174     175     14      15
81      84      89      92      125     128     161     164     169     172     9       12      45      48
83      82      91      90      127     126     163     162     171     170     11      10      47      46
85      88      121     124     157     160     193     196     5       8       41      44      77      80
87      86      123     122     159     158     195     194     7       6       43      42      79      78


******************************************************************************
** Zelite li pokusati ponovno? (y/n)
```

Slika 4.16. Unesen je izraz ‘*ajdeti*’ (nasumična veličina matrice je 14)



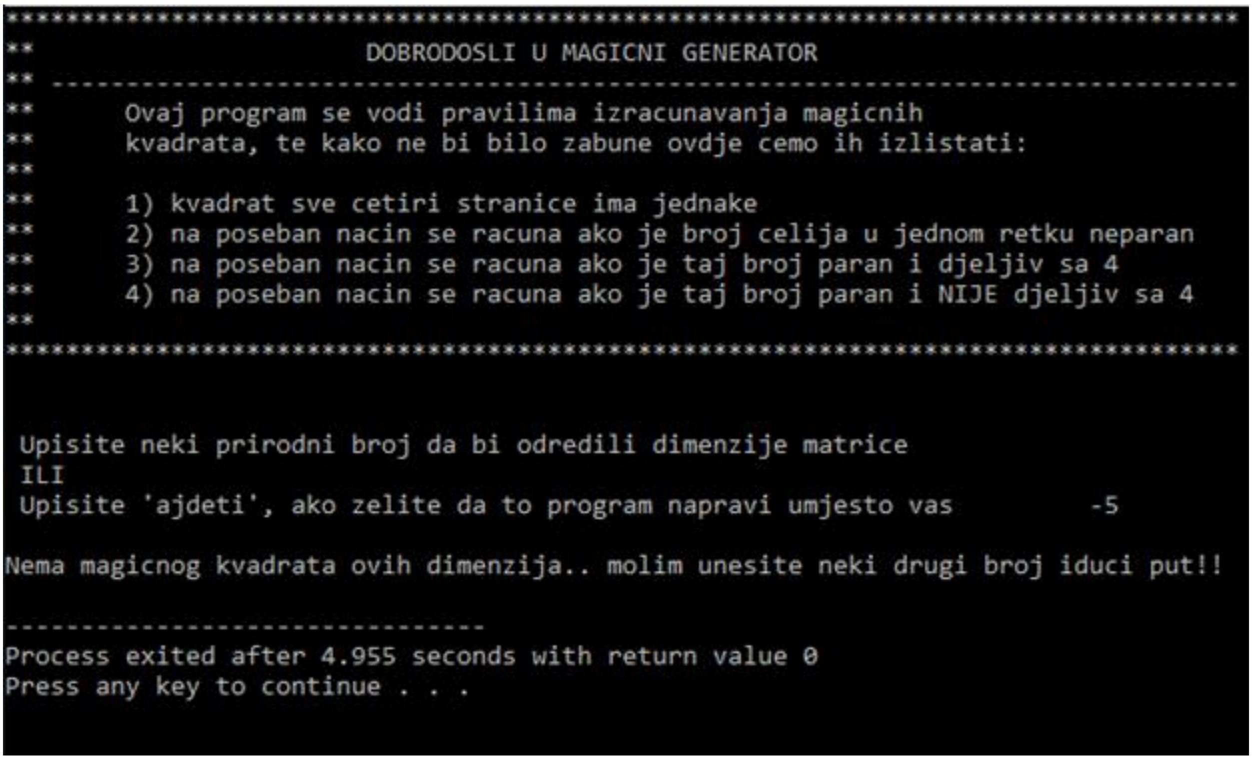
Slika 4.17. Unesen je izraz ‘*ajdeti*’ (nasumična veličina matrice je 5)

Prilikom unosa veličine matrice moguće je da korisnik unese neispravan broj, odnosno broj za koji nije moguće izračunati magični kvadrat. Kako je već navedeno, brojevi za koje program računa magični kvadrat su svi prirodni osim broja 2. Prilikom unosa neispravnog broja, program vraća povratnu informaciju: “*Nema magičnog kvadrata ovih dimenzija…molim unesite neki drugi broj iduci put!*” što je vidljivo na slici 4.18., te slici 4.19.



```
 Upisite neki prirodni broj da bi odredili dimenzije matrice
 ILI
 Upisite 'ajdeti', ako zelite da to program napravi umjesto vas        2

Nema magicnog kvadrata ovih dimenzija.. molim unesite neki drugi broj iduci put!!

--------------------------------
Process exited after 227.3 seconds with return value 0
Press any key to continue . . .
```

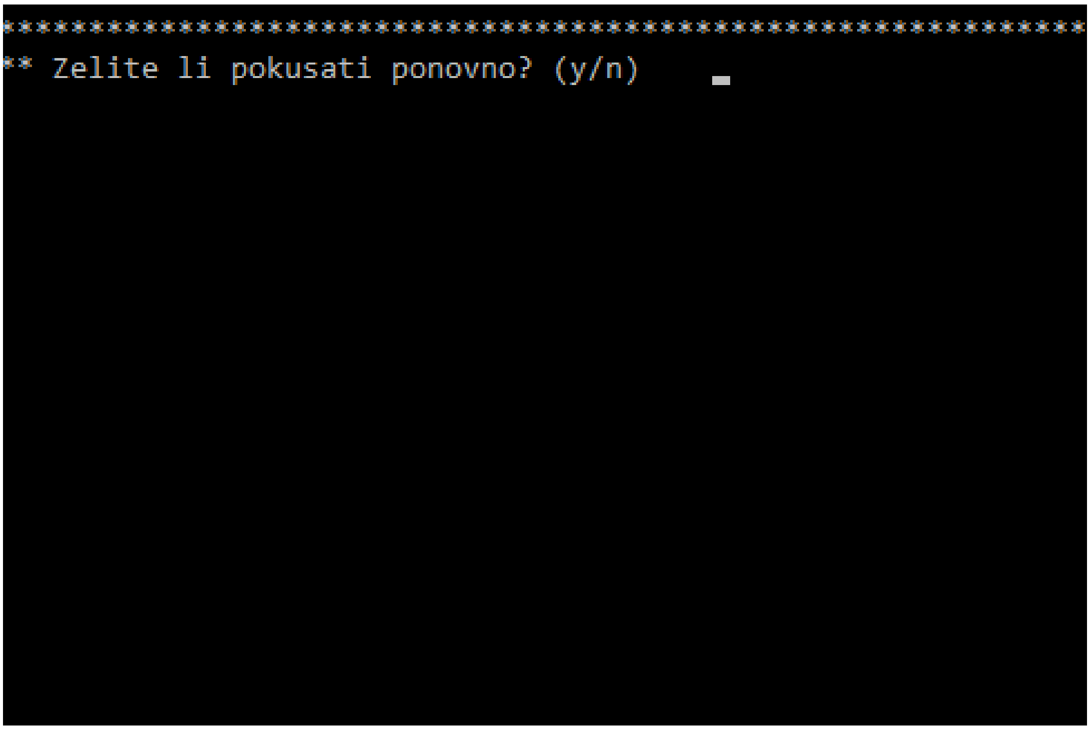
Slika 4.18. Unos broja za koji ne postoji magični kvadrat (npr. broj 2)



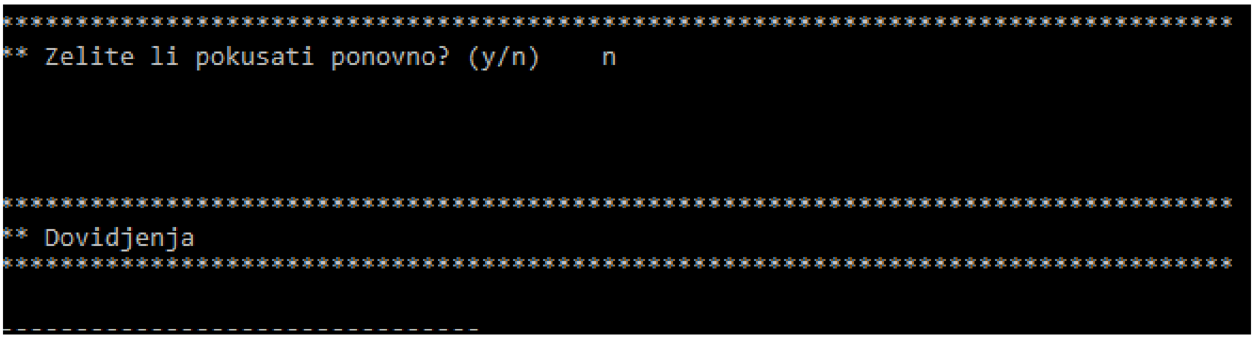
```
*******************************************************************************
**                      DOBRODOSLI U MAGICNI GENERATOR
** ---------------------------------------------------------------------------
**      Ovaj program se vodi pravilima izracunavanja magicnih
**      kvadrata, te kako ne bi bilo zabune ovdje cemo ih izlistati:
**
**      1) kvadrat sve cetiri stranice ima jednake
**      2) na poseban nacin se racuna ako je broj celija u jednom retku neparan
**      3) na poseban nacin se racuna ako je taj broj paran i djeljiv sa 4
**      4) na poseban nacin se racuna ako je taj broj paran i NIJE djeljiv sa 4
**
*******************************************************************************


 Upisite neki prirodni broj da bi odredili dimenzije matrice
 ILI
 Upisite 'ajdeti', ako zelite da to program napravi umjesto vas        -5

Nema magicnog kvadrata ovih dimenzija.. molim unesite neki drugi broj iduci put!!

--------------------------------
Process exited after 4.955 seconds with return value 0
Press any key to continue . . .
```

Slika 4.19. Unos broja za koji ne postoji magični kvadrat (npr. broj -5)

Nakon svakog unosa, korisniku je ponuđeno da pokuša ponovno upisati broj. Potrebno je samo upisati ‘y’ za potvrdu (yes) ili ‘n’ za negativan odgovor (no). Primjer je prikazan na slici 4.20. Ako korisnik odabere ‘n’ otvara mu se prozor kao na slici 4.21.



Slika 4.20. Ponovni unos broja



Slika 4.21. Negativan odgovor na upit o ponovnom upisu broja

5. ZAKLJUČAK

Zadatak seminarskog rada bilo je rješavanje magičnih kvadrata u programskom okruženju C. Magični kvadrati zagonetka su koja je oduševljavala ljude kroz mnoga razdoblja, a mogli su se naći u različitim kulturama i fascinirali su ljude tijekom različitih vremenskih perioda.

Prema legendi, prvi magični kvadrat je otkriven u Kini oko 2800.g.pr.Kr.. Unatoč činjenici da su magične N-kocke izučavane od davnina, one su još uvijek predmet istraživanja i proučavanja. Čarobne četvorine pojavljuju se u djelima brojnih umjetnika (najpoznatiji je Albrecht Dürer), matematičara (Norbert Behnke, F.Bernard de Bessy, H.C.Agrippa), uglednika (Benjamin Franklin) itd.

U radu je prikazan rad samog programa, te sva teorijska podloga potrebna za njegovu izradu. Prikazan je povijesni pregled programskih jezika,od samog početaka pa sve do modernih vremena. Spomenuta je podjela programskih jezika, te su navedeni prvi jezici kao što je strojni jezik, simbolički jezik, pa sve do viših programskih jezika. Poseban naglasak stavljen je na programski jezik C u kojemu je napravljena programska izvedba same igre. Dotaknuta je i sintaksa samog jezika, a navedeni su i primjeri jednostavnijih programa.

Nakon opisa programskoga jezika C, pojašnjena je sama definicija magičnih kvadrata. Kako oni imaju jako dugu i zanimljivu povijest, te su poznati od davnina, postoje i brojni algoritmi za njihovo rješavanje. Sama izrada progama vezana je uz rješavanje matrica, te program za zadani red veličine matrice, ispisuje popunjenu matricu takvu da ima svojstvo magičnosti.

Program izrađen u ovome radu namjenjen je rješavanju matematičkog problema, te razvoju intelektualnih sposobnosti pojedinca. Igra može biti zanimljiva i služiti u provjeravanju određenih kvadrata prilikom rješavanja zadataka vezanih uz čarobne četvorine u srednjim i osnovnim školama. Zadaci rješavanja magičnih kvadrata namijenjena je svim uzrastima, te može dobro doći bilo kome za kvalitetno provođenje vremena.

LITERATURA

[1] J. Šribar, B. Motik, Demistificirani C++, Element, Zagreb, 2010. , treće dopunjeno izdanje

[2] Brian W.Kernighan, Dennis M.Ritchie, Programski jezik C, prijevod drugog izdanja, 1989.

[3] Tihomir Čukman, Vlatko Bolt, C/C++ kroz primjere, 1994.

[4] W. S. Andrews, Magic Squares and Cubes , Open Court Publishing Company, 1917.

[5] Programski jezik C – PMF (http://web.math.pmf.unizg.hr/~singer/Prog_Add/c.pdf)

[6] Strojni jezik (https://hr.wikipedia.org/wiki/Strojni_jezik)

[7] PROGRAMSKI JEZICI I OSNOVE PROGRAMIRANJA (http://bit.ly/290ijit)

[8] Magični kvadrat (https://hr.wikipedia.org/wiki/Magi%C4%8Dni_kvadrat)

[9] Čarobne četvorine (iliti magični kvadrati), Darko Veljan (http://bit.ly/291oLui)

[10] Magični kvadrati – Antonija Horvatek (http://bit.ly/291pbRx)

[11] Cijeli članak – Element (http://bit.ly/29jxlQZ)

[12] Magični kvadrat ppt. (http://www.mathos.unios.hr/~middlemath/ppt/magicni-web.ppt)

[13] C (programski jezik) (https://sh.wikipedia.org/wiki/C_(programski_jezik))

[14] MAGIČNE KOCKE I 3-ADSKA ZETA FUNKCIJA (diplomski rad) (*http://digre.pmf.unizg.hr/4382/1/diplomski.pdf*)

SAŽETAK

Zadatak ovog seminarskog rada bilo je rješavanje problema magičnih kvadrata u programskom jeziku C. Za izradu programa važno je dobro poznavanje spomenutog programskog jezika, te način rješavanja čarobnih četvorina. Na početku seminarskog rada opisan je uvod u programske jezike, te karakteristike samog C jezika, od njegovog početka i razvoja, sve do njegove sintakse i upotrebe. Kako bi jezik bio što jasnije pojašnjen, priloženi su i primjeri jednostavnijih programa. Nakon toga, opisan je sam problem magičnih kvadrata, njihova definicija, rješavanje i sama povijest njihovog pojavljivanja u prošlosti starih civilizacija. Naknadno, objašnjeni su algoritmi rješavanja magičnih kvadrata u dva slučaja, kada su oni parni, te kada su neparni. Naposljetku, na osnovi stečenog znanja, uspješno je izrađen program koji provjerava za zadane kvadrate jesu li magični.

Ključne riječi: programski jezik C, magični kvadrati, maigična suma, Lo Shu, Melancolia I, algoritmi, program

ABSTRACT

The aim of this seminar work was solving magic squares in the programming language C.To create a program, it is important to have a good knowledge of the said programming language as well as the way of dealing with magical foursquare.At the beginning of the seminar paper describes the introduction to programming languages, as well as characteristics of the C language, from its inception and development, until his syntax and usage.To have the language clearly explained, attached are examples of simple programs.After that, I described the problem of magic squares, their definition, solving the very history of their occurrence in the history of ancient civilizations. Subsequently, explained are the algorithms of solving magic squares in the two cases, when they are steam, and when they are odd. Finally, based on acquired knowledge, I have successfully developed a program that checks the score squares whether magic.

Keywords: programming language C, magic squares, magic sum, Lo Shu, Melancolia I, algorithms, program

ŽIVOTOPIS

Lea Lorger, rođena je 22. siječnja 1994. godine u Slavonskom Brodu. Osnovnu školu je završila u Županji, te u istom gradu upisuje opći smjer Gimnazije Županja. Maturirala je 2012. godine, nakon čega je upisala Preddiplomski sveučilišni studij Računarstva na Elektrotehničkom fakultetu u Osijeku. Završila je drugu godinu preddiplomskog studija Računarstva na Fakultetu elekrtrotehnike, računarstva i informacijskih tehnologija. Od posebnih vještina ističe znanje engleskog jezika govornim i pismenim izražavanjem, poznavanje rada na računalu od čega ističe korištenje Office-a, rad u programskim jezicima C, C++, C#, osnove HTML-a, CSS-a i JavaScript-a te poznaje rad s bazama podataka u MySQL-u.

PRILOZI

Na CD-u priloženom uz Završni rad nalaze se:

Dokumenti:

MagicniKvadrati.doc

MagicniKvadrati.pdf

MagicniKvadratiKod.txt