

Univerzitet u Beogradu
Elektrotehnički fakultet
Bulevar kralja Aleksandra 73

REALIZACIJA TJURINGOVE MAŠINE

semestralni rad iz diskretne matematike

Student:
Nemanja Karapetrović, ****/****

Mentor:
dr Branko J. Malešević

SADRŽAJ

| | |
|--|----|
| Uvod..... | 3 |
| O Alanu Turing-u..... | 4 |
| Osnovni podaci..... | 4 |
| Biografija..... | 5 |
| Kriptoanaliza..... | 5 |
| Računari i Tjuringov test..... | 5 |
| Tužba za homoseksualna dela i Tjuringova smrt..... | 5 |
| Počasna obeležja..... | 5 |
| Enigma..... | 5 |
| Seksualna orijentacija..... | 6 |
| PROGRAMSKI KOD..... | 9 |
| Realizacija Turing-ove mašine u programskom jeziku C..... | 9 |
| Test primeri | 29 |
| Test primer 1 – prvi komplement zadatog broja..... | 29 |
| Test primer 2 – šifrovanje za jedno mesto ulevo, azbuka {1,b}..... | 31 |
| Test primer 3 – šifrovanje za jedno mesto ulevo, azbuka {0,1,b}..... | 33 |
| Test primer 4 – dodavanje broja jedan u binarnom sistemu..... | 35 |
| Test primer 5 – oduzimanje broja jedan u binarnom sistemu..... | 37 |
| PRILOG..... | 39 |
| Specijalizovani programi..... | 39 |
| Program 1 – nula funkcija..... | 39 |
| Program 2 – funkcija sledbenik..... | 45 |
| Program 3 - funkcija projekcije na prvu koordinatu..... | 50 |
| Program 4 – deljivost zadatog broja sa 4..... | 56 |
| Program 5 - deljivost zadatog broja sa 8..... | 63 |
| Literatura..... | 72 |

UVOD

Projekat je realizovan u programskom jeziku C. Korišćeno je okruženje iz programskog paketa Microsoft Visual Studio 2015. Za pokretanje programa neophodno je da računar na kom se pokreće ima instaliran navedeni programski paket, kao i da ulazne datoteke budu ispravno zadate¹.

Glavni program poziva funkcije, *Turing* učitavanje_trake(Turing*lst)* koja čita datoteku **ulaz.txt** u kojoj se nalazi zapis koji simulira traku Turing-ove mašine. U slučaju nekorektnog sadržaja u datoteci ispisuje se poruka o grešci i prekida se program. Ako je datoteka korektna funkcija formira jednostruko ulančanu listu u koju smešta simulacije vrednosti polja trake Turing-ove mašine. Zatim se poziva, *Program*ucitavanje_programa(Program*lstP)*, koji cita datoteku **program.txt** prekida ga u slučaju nekorektnog zapisa u istoj uz ispisivanje poruke o primećenoj nepravilnosti. U slučaju korektnog unosa, formira se dvostruko ulančana lista sa podacima potrebnim za obradu. Dalje, se poziva funkcija *Turing*obrada(Turing*lst,Program*lstP)* koja simulira kretanje glave Turing-ove mašine. Po njenom završetku poziva se *void ispis_trake(Turing*lst)*, koja ispisuje sadržaj obradjene trake koja se učitala na početku programa. Na kraju se pozivaju funkcije *Turing*obrisi1(Turing*lst)* i *Program*obrisi2(Program*lst)* koje oslobađaju korišćeni memorijski prostor i time ga stavljaju na raspolaganje nekoj budućoj obradi podataka na datom računaru.

¹ Program je otporan na bilo koju vrstu greške osim na nestanak struje.

O ALANU TJURINGU

Osnovni podaci

Alan Matison Tjuring (engl. Alan Mathison Turing; London, 23. jun 1912 — Češir, 7. jun 1954) je bio engleski matematičar, logičar i kriptograf. Smatra se ocem modernog računarstva - napravio je koncept algoritama koji se danas koristi u svetu,^[3] i računanja pomoću Tjuringove mašine, formulišući danas široko prihvaćenu Tjuringovu verziju Čerč-Tjuringove teze, naime, da svaki praktični računarski model ima ili ekvivalentne mogućnosti Tjuringovoj mašini, ili njegove mogućnosti predstavljaju podskup mogućnosti Tjuringove mašine.

Svojim Tjuringovim testom, dao je značajan i provokativan doprinos debati koja se ticala veštačke inteligencije: da li će ikad biti moguće reći da je mašina svesna i da može da misli. Kasnije je radio u Nacionalnoj fizičkoj laboratoriji, a 1947. godine je prešao u Mančesterski univerzitet da radi, uglavnom na softveru, na mančesterskom Marku I, za koji se tada smatralo da je jedan od prvih pravih računara.



Slika 1 -
Alan Matison Turing(1912-1954)

Tokom Drugog svetskog rata, Tjuring je radio u Blečli parku, britanskom kriptanalitičkom centru, i bio je jedno vreme šef Hut-a 8, odeljenja zaduženog za nemačku mornaricu. Tjuring je razvio više tehnika za razbijanje šifara, uključujući metod bombe, elektromehaničku mašinu, koja je mogla da otkrije postavke Enigme. Godine 1952. Tjuring je osuđen za delo „velike nepristojnosti“ (engl. Act of gross indecency) pošto je priznao da je bio u vezi sa muškarcem u Mančesteru. Bio je poslat na hormonsku terapiju. Tjuring je umro 1954. godine pošto je pojeo jabuku napunjenu cijanidom. Njegova smrt se smatra samoubistvom.

Biografija

Alanov otac, Džulijus Tjuring (Julius Mathison Turing) je radio u Indiji, ali kada je supruga ostala u drugom stanju, 1911. godine, mladi bračni par odlučuje da je bolje za dete da odrasta u Engleskoj, pa su se vratili u London, gde se Alan i rodio 23. juna 1912. godine. Kao veoma mlad je pokazivao mnogo više nego što se očekuje od deteta njegovih godina. Priča se da je naučio da čita za tri nedelje, a i kao jako mlad je pokazivao interesovanje za brojeve.

Sa šest godina su ga upisali u dnevnu školu St Michael. Godine 1926., a sa 14 godina je upisan u Šerborn školu u Dorsetu. Tjuringova naklonjenost matematici i hemiji mu nije donela poštovanje profesora u Šerbornu, pošto se kod njih učenje zasnivalo na opštem obrazovanju. Vreme provedeno u Šebroonu i u pratećem internatu u velikoj meri će oblikovati njegova buduća

lična i profesionalna interesovanja i domete. U školi se negovalo klasično obrazovanje, te ne čudi što je zbog svojih istupa od redovnog nastavnog plana u matematičke vode jednom izazvao bes razrednog starešine koji mu je podviknuo: „Ova soba smrdi na matematiku! Izađi i donesi sprej za dezinfikovanje!“ Ipak, većina nastavnika i drugi učenici su mu se divili zbog talenta za prirodne nauke i hrabрили su ga da nastavi u tom smeru.

Veći deo 1937. i 1938. godine je proveo na Princeton Univerzitetu, učeći kod Alonza Čerča (Alonzo Church). Godine 1938. dobio je diplomu Prinstona. Vratio se 1939. godine u Kembridž, gde je kod Ludviga Vitgenštajna (Ludwig Wittgenstein) pohađao časove o fundamentalnoj matematici.

Kriptoanaliza

U toku Drugog svetskog rata, Tjuring je bio najzaslužniji za razbijanje nemačkih šifara, radeći u Blečli parku. Radio je na razbijanju Enigme i Lorenca C3 40/42, i jedno vreme je bio šef odeljenja Hut 8.

Od 1938. godine Tjuring je radio u Britanskoj organizaciji za razbijanje šifara (Government Code and Cypher School). Radio je na problemu nemačke mašine, Enigma, i sarađivao sa Dili Noks (Dilli Knox), koja je takođe radila u GCCS-u. 4. septembra 1939. godine, dan pošto je Engleska objavila rat Nemačkoj, Tjuring je prešao u Blečli park, štab GCCS-a za vreme Drugog svetskog rata.

Računari i Tjuringov test

Od 1945. do 1947. godine je u Nacionalnoj Fizičkoj Laboratoriji, gde je radio na dizajniranju ACE-a (Automatic Computing Engine). 19. februara 1946. godine prezentirao je dizajn prvog računara u Britaniji. Iako ga je dizajnirao, ACE je izvršio prvi program tek 10. maja 1950. godine, i to u Tjuringovom odsustvu, jer je on tada bio u Kembridžu. Godine 1949. je postao direktor računarske laboratorije, Mančesterskog Univerziteta, i radio je na softveru jednog od prvih pravih računara, Mančesterskom Marku I. Radio je i na problemu veštačke inteligencije, i predstavio je eksperiment poznat kao Tjuringov test.

Godine 1948. je pisao šahovski program za računar koji još uvek nije postojao, tako da je 1952. godine sam simulirao program, koji je jednom pobedio i jednom izgubio meč.

Tužba za homoseksualna dela i Tjuringova smrt

Tjuring je bio homoseksualac, kada je to bilo nezakonito u Engleskoj, i predstavljano kao mentalna bolest. Osuđen je jer je bio u vezi sa Arnoldom Marejom, i bio je podvrgnut terapiji primanja estrogen injekcija, što je trajalo godinu dana. To je, u stvari, značilo i prestanak njegovog rada. Godine 1954. je umro od trovanja cijanidom, jer je pojeo jabuku u koju je bio ubrizgan cijanid. Njegova smrt je proglašena samoubistvom.

Osuda za "teško nedolično ponašanje" je povučena i njega je, 24. decembra 2013. godine, posthumno pomilovala Kraljica Elizabeta II. Alan Tjuring je tek četvrta osoba od 1945. godine, koja je u Ujedinjenom Kraljevstvu dobila kraljevsko pomilovanje.

Počasna obeležja

Od 1966. godine Asocijacija za računarstvo (Association for Computing Machinery) dodeljuje Tjuringovu nagradu, za računarska dostignuća. Smatra se da je ta nagrada u svetu računara jednaka sa Nobelovom nagradom. U Mančesteru, u gradu u kojem je radio do kraja svog života, se održavaju razne počasti u ime Tjuringa. Jedna od ulica u Mančesteru je 1994. godine nazvana po Tjuringu (Alan Turing Way).

Statua Tjuringu je otkrivena u Mančesteru 23. juna 2001. godine u Sakvil parku, između Mančesterskog univerziteta i ulice kanal (Canal Street). Kad je trebalo da bude Tjuringov 86-i rođendan, 23. juna 1998. godine, Endru Hodžis, njegov biograf, je otkrio zahvalnicu na kapiji

njegovog doma, u Londonu, gde je sada Kolonejd Hotel. To je uradio na 50. godišnjici njegove smrti, 7. juna 2004. godine.

Enigma

Prvi septembar 1939. godine. Nemačka i Sovjetski Savez kreću na Poljsku. Uporedo sa oružanim sukobima na zemlji i nebu, van očiju javnosti vodi se jednako važan, tihi rat. Godinama unazad, poljska obaveštajna služba zajedno sa francuskim i britanskim obaveštajcima zna za Enigmu, mašine za šifrovanje poruka koja će zadavati muke saveznicima. Engleska vlada odmah po izbijanju rata okuplja tim obaveštajaca i kriptanalitičara u Vladinoj školi za kodove i šifre u Blečli parku u okolini Londona. Tajni projekat Ultra predvode Alan Tjuring i Dilvin Noks. Za samo nekoliko nedelja, Tjuring, uz pomoć kolege matematičara Gordona Velčmana, uspeva da konstruiše elektromehanički kriptanalitički uređaj, takozvanu Bombu za razbijanje šifri Enigme.

Fizičko srce Enigme bili su rotori koji su se okretali različitim brzinama. Svaki od rotora je mogao da se nađe u 26 različitih položaja. U slučaju najjednostavnije Enigme, postojala su tri rotora, te je stanja u kojima su rotori mogli da se nađu zajedno bilo $26^3 = 17576$. Pomorske Enigme imale su i do osam rotora. Za tumačenje poruke poslate jednom Enigmom, operater sa druge strane je morao da ima istovetnu mašinu i da zna početnu postavku mašine, takozvani key setting. Nemci su početno stanje menjali mesečno. Situaciju je komplikovalo što se sa svakim pritiskom dugmeta tastature kôd menjao, tako da neko slovo – na primer a – u prvom pritisku daje jedno slovo, a u drugom neko drugo slovo: dakle, dva slova a u jednoj reči neće biti zamenjena istim slovom. Promena koda nakon pritiska slova se ostvarivala time što se menjala pozicija rotora. Vojne Enigme su imale i prekidačku tabluna kojoj su se slova uparivala, tako da su slova u parovima menjala mesto prilikom kodiranja – na primer, a bi zamenilo mesto sa m – što je dodatno komplikovalo situaciju.

Enigma je imala slabe tačke. Jedna je bila to što dato slovo nikad nije bilo kodirano istim tim slovom – na primer, ako bismo pritisli a na tastaturi, nikada ne bismo kao njegovu šifrovanu zamenu dobili a. Drugi izvor ranjivosti bilo je to što su se neke fraze ponavljale – najčešće one koje su se ticale meteoroloških okolnosti. Konkretnije, Nemci su svako jutro slali meteorološki izveštaj.

Tjuring je svoj dizajn Bombe bazirao na logičkoj kontradikciji – traženo stanje, odnosno postavka za kodiranje nije moglo da rezultira time da se jedno slovo menja tim istim slovom. Dakle, Bomba je tragala za tačnim postavkama odnosno kodom, i odbacivala one postavke koje su davale isto slovo za dato slovo. Većina mogućih postavki je bila kontradiktorna, te su ih odbacivali, pa je ostajalo samo nekoliko koje je imalo smisla detaljno proveriti.

1941. godine, Tjuring je uspeo, zahvaljujući sredstvima za koja je apelovao direktno Čerčilu, da napravi Bombu za komplikovanu pomorsku Enigmu.

Nešto kasnije, 1942, Tjuring razvija tehniku duhovito nazvanu turingizmus, koja će biti osnova za konstruisanje Kolosa, prvog digitalnog programibilnog računara, namenjenog razbijanju nacističkih Lorencovih šifri. Bomba i Kolos su nemačkim snagama zadale udarac koji je jezičak pobeđe nepovratno pomerio na stranu saveznika.

Seksualna orijentacija

Tokom školovanja u Šerbornu 1928. godine Tjuring se sprijateljio s Kristoferom Morkomom, starijim učenikom, takođe veoma nadarenim za prirodne nauke. Ova ljubav, podgrejavana dugim, i na svoj način romantičnim razgovorima o vrednosti broja π i teoriji relativnosti, bila je, po svemu sudeći platonska, jer Morkom najverovatnije nije bio homoseksualac. Morkom umire 1930. godine od tuberkuloze.

U knjizi Čovek koji je suviše znao, autor Dejvid Livit piše:

Nije iznenađujuće što je gubitak idealnog sadruga uticao da se u Tjuringovoj mašti začne ideal romantične ljubavi, a to se zbilo pre nego što je ta idealna veza uopšte mogla da se rasprši ili preraste u zreo odnos. U romanu Edgara M. Forstera, *Moris*(1914), ljubav naslovnog lika prema Klajvu Daramu prvo se razvila u čvrst i, po svoj prilici, postojan odnos (međutim, što je značajno, na Klajvovo insistiranje među njima nije bilo seksualnih odnosa), potom se pretvorila u prezir kada je Klajv odlučio da se oženi. Nasuprot tome, Tjuring nikada nije imao priliku da produbi svoju privrženost prema Krisu Morkomu ma kakav bio ishod.

U eseju *Priroda duše* iz 1931. godine, posvećenom majci Kristofera Morkoma, navodi: „Kada telo umre, mehanizam tela koji drži dušu prestaje da funkcioniše, te duša pre ili kasnije, možda i u trenutku smrti, nalazi novo telo“. Ta misao, o novom telu kao domu besmrtno duše, biće nadalje njegova vodilja u svim profesionalnim i intelektualnim stremljenjima – centar njegove vizije biće mašine kao entiteti sa svešču, ne slični, već jednaki ljudima.

Godine 1941. Tjuring će zaprositi Džuan Klark koja je bila deo kriptografske ekipe u Blečli Klarku. Džuan će pristati, ali Tjuring će joj ubrzo priznati svoju homoseksualnu orijentaciju i reći da ipak ne može da dopusti da se venčaju. Klarkova, po sopstvenim rečima, nije bila iznenađena tim priznanjem i ostaće njegov blizak prijatelj sve do Tjuringove smrti.

PROGRAMSKI KOD

Realizacija Turing-ove mašine u programskom jeziku C

U nastavku će biti dat sadržaj dateoteke programskog jezika C gde je napisan programski kod za računarsku realizaciju Turing-ove mašine kao i slike izlaznih datoteka koje se dobiju kao rezultat programske obrade ulaznih datoteka(program formira novu datoteku) za test primere sa časova.

Kako bi program ispravno funkcionisao neophodno je da se na istoj lokaciji gde se podrazumevano nalazi fajl Universal_Turing_machine.c nalaze i datoteke ulaz.txt(koja simulira sadržaj trake Turingove mašine), program.txt(koja sadrži zapisan program koji se izvršava po formatu i oznakama sa časa, na primer: $f(q_0,1)=(q_1,1,+1)$).

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

```
#include<time.h>
```

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

```
#include<ctype.h>
```

```
#include<math.h>
```

```
typedef struct turing {  
    char pod;  
    struct turing *pret;  
    struct turing *sled;  
}Turing;
```

```
typedef struct program {  
    int q; // 1, 2, 3, ... ,n  
    char uslov; // {1, b} V {0, 1, b}  
    char sl_stanje_q_plus_ili_minus; // +, -, 1, 2, 3, ... ,n  
    int sl_stanje_broj;  
    char zamena; // {1, b} V {0, 1, b}  
    int pomeraj_glave; // +1 V -1  
    struct program *pret;  
    struct program *sled;  
}Program;
```

```
void podaci() {  
    time_t vreme;  
    time(&vreme);
```

```

puts("-----");
puts("-----");
printf("Univerzitet u Beogradu\n");
printf("Elektrotehnicki fakultet Beograd\n");
printf("Bulevar kralja Aleksandra 73\n");
puts(" ");
printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
printf("Razlog: predispitne obaveze\n");
printf("Format: semestralni rad iz diskretne matematike\n");
printf("Tema: Racunarska realizacija Turing-ove masine\n");
printf("Test primer [universal]: UNIVERZALNI PROGRAM\n");
printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
printf("Student: Nemanja Karapetrovic [broj indeksa: ****/****]\n");
printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz.txt i da se nalazi gde i
izvorni fajl Universal_Turing_machine.c .\n");
printf("Vreme: %s", ctime(&vreme));
puts("-----");
puts("-----");
puts("-----");
puts("");
}

```

```

Turing* učitavanje_trake(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspeo!\n");
        exit(1);
    }
    while (fscanf(p, "%c", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
    }
}

```

```

        novi->pod = pod1;
        novi->sled = NULL;
        if (lst == NULL) {
            lst = novi;
        }
        else {
            posl->sled = novi;
            novi->pret = posl;
        }
        posl = novi;
    }
    fclose(p);
    return lst;
}

```

```

int da_li_je_broj_iza_kog_sledi_zarez(FILE*p) { //Uraditi realokaciju.
    char cifra;
    int broj[1000];
    int i = 0;
    int kraj;
    int q1;
    int stepen;
    int koeficijent;
    fscanf(p, "%c", &cifra);
    broj[i++] = (int)cifra - 48; // Da bi se od ASCII koda dobio konkretan broj.
    if (isdigit(cifra)) {
        while (isdigit(cifra)) {
            fscanf(p, "%c", &cifra);
            if (!isdigit(cifra)) break;
            broj[i++] = (int)cifra - 48;

        } //Kad iscrpi cifre proverava da li je zatvorena zagrada sledeca.
        kraj = i;
        if (cifra == ',') { //Ako jeste broj iza koga sledi zarez onda ga izracunja.
            for (i = kraj - 1, q1 = 0, stepen = 0; i >= 0; i--) {
                koeficijent = pow(10, stepen);
                q1 += broj[i] * koeficijent;
                stepen++;
            }
            return q1;
        }
    }
}

```

```

        }
        else {
            return -1;
        }
    }
    else {
        return -1;
    }
}

int da_li_je_broj_ili_plus_ili_minus(FILE*p) {
    char cifra;
    fscanf(p, "%c", &cifra);
    if (isdigit(cifra) || cifra == '+' || cifra == '-') {
        if (cifra == '+') return 2;
        if (cifra == '-') return 3;
        return cifra;
    }
    else {
        return 0;
    }
}

```

```

Program* učitavanje_programa(Program* lstP) {
    FILE*p;
    Program*novi, *posl, *tek, *pom;
    int q1; // 1, 2, 3, ... ,n
    char uslov1; // {1, b} V {0, 1, b}
    char sl_stanje_q_plus_ili_minus1; // +, -, 1, 2, 3, ... ,n
    int sl_stanje_broj1;
    char zamena1; // {1, b} V {0, 1, b}
    int pomeraj_glave1; // +1 V -1

    char procitano;
    int procitani_broj;
    int sl_stanje_broj1_procitano;
    int sortiran = 0;

    p = fopen("program.txt", "r");
    if (p == NULL) {

```

```

    printf("Neuspesno otvaranje datoteke!!!\n");
    exit(1);
}

while (fscanf(p, "%c", &procitano) != EOF) {
    //U ovom trenutku procitano bi trebalo da bude f.
    if (procitano == 'f') {
        fscanf(p, "%c", &procitano);
        // Sad bi procitano trebalo da bude (.
        if (procitano == '(') {
            fscanf(p, "%c", &procitano);
            //Sad bi procitano trebalo da bude q.
            if (procitano == 'q') {
                //Sad bi procitano trebao da bude broj.
                q1 = da_li_je_broj_iza_kog_sledi_zarez(p); // Ako jeste
                ujedno ga cita i smesta u q1, ako nije ispisuje poruku o gresci i prekida program.
                if (q1 >= 0) {
                    fscanf(p, "%c", &procitano);
                    //Sad bi procitano trebalo da bude iz {1, b} V {0, 1,
                    b}.
                    if (procitano == '0' || procitano == '1' || procitano ==
                    'b') {
                        uslov1 = procitano;
                        // Sad bi procitano trebalo da bude ).
                        fscanf(p, "%c", &procitano);
                        if (procitano == ')') {
                            fscanf(p, "%c", &procitano);
                            //Sad bi procitano trebalo da bude =.
                            if (procitano == '=') {
                                fscanf(p, "%c", &procitano);
                                //Sad bi procitano trebalo da
                                bude (.
                                if (procitano == '(') {
                                    fscanf(p, "%c",
                                    &procitano);
                                    if (procitano == 'q') {
                                        //Sad bi
                                        procitano trebao da bude broj ili plus ili minus.
                                        if
                                        (sl_stanje_broj1_procitano = da_li_je_broj_ili_plus_ili_minus(p)) { //Svakako se ulazi jer, if
                                        gleda razliku od 0.

```

```

                                                                    if
(sl_stanje_broj1_procitano == 2) { //Plus. Stanje q+.

    sl_stanje_q_plus_ili_minus1 = '+';

    sl_stanje_broj1 = -2; //Ovo je inikator da ne gledamo to polje.

                                                                    }
                                                                    if
(sl_stanje_broj1_procitano == 3) { //Minus. Stanje q-.

    sl_stanje_q_plus_ili_minus1 = '-';

    sl_stanje_broj1 = -2; //Ovo je inikator da ne gledamo to polje.

                                                                    }
                                                                    if
(isdigit(sl_stanje_broj1_procitano)) { //Broj. Stanje qn. Ovde smo sigurni da je cifra u pitanju.

    sl_stanje_broj1 = sl_stanje_broj1_procitano;

    sl_stanje_q_plus_ili_minus1 = '*'; // Zvezdica nam je indikator da ne gledam to polje.

                                                                    }//
Ovde se zavrсило citanje i smestanje u lokalne promenljive svih parametara do parametra zamene.

    fscanf(p, "%c", &procitano);

                                                                    //Sada
bi procitano trebalo da bude zarez.

                                                                    if
(procitano == ',') {

    fscanf(p, "%c", &procitano);

    //Sada bi procitano trebalo da bude iz {1, b} V {0, 1, b}.

    if (procitano == '0' || procitano == '1' || procitano == 'b') {

        zamenal = procitano;

```

```

fscanf(p, "%c", &procitano);

//Sada bi procitano trebalo da bude zarez.

if (procitano == ',') {

    fscanf(p, "%c", &procitano);

    //Sada bi procitano trebalo da bude znak pomeraja tj. + ili -.

    if (procitano == '+' || procitano == '-') {

        if (procitano == '+') {

            pomeraj_glave1 = 1;

        }

        else {

            pomeraj_glave1 = -1;

        }

        fscanf(p, "%c", &procitano);

        //Sada bi procitano trebalo da bude isključivo 1.

        if (procitano == '1') {

            fscanf(p, "%c", &procitano);

            //Sada bi procitano trebalo da bude ).

            if (procitano == ')') {

                fscanf(p, "%c", &procitano);

                //Sada bi procitano trebalo da bude \n (znak za novi red).

```

/*FORMIRANJE DVOSTRUKO ULANCANE LISTE*/

if (procitano == '\n') { // Ako je i ovaj uslov ispunjen znaci
da smo uspesno procitali jedan red datoteke. Sada formiramo dvostruko ulancanu listu.

```
    novi = malloc(sizeof(Program));  
  
    if (novi == NULL) {  
  
        printf("Alokacija      memorije      nije  
uspela!!!\n");  
  
        system("pause");  
  
        exit(1);  
  
    }  
  
    novi->q = q1;  
  
    novi->uslov = uslov1;  
  
    novi->sl_stanje_broj = sl_stanje_broj1;  
  
    novi->sl_stanje_q_plus_ili_minus      =  
sl_stanje_q_plus_ili_minus1;  
  
    novi->zamena = zamena1;  
  
    novi->pomeraj_glave = pomeraj_glave1;  
  
    novi->pret = NULL;  
  
    novi->sled = NULL;  
  
    if (lstP == NULL) {
```



```

        lstP = novi;

    }

else { //Sortiranje

    tek = lstP;

    pom = lstP->sled;

    sortiran = 0;

    while (tek != NULL && sortiran == 0) {

        if (tek == lstP && pom == NULL

&& !(novi->q > tek->q)) {

            if (novi->q <= tek->q) {

                tek->pret = novi;

                novi->sled = tek;

                lstP = novi;

                sortiran = 1;

            }

        }

        while (tek != NULL && tek->q <=

novi->q) { //Dodavanje bilo gde.

            tek = tek->sled;

```

```

    }

    if (tek != NULL && sortiran == 0) {

        pom = tek;

        tek = tek->pret;

        tek->sled = novi;

        novi->pret = tek;

        novi->sled = pom;

        pom->pret = novi;

        sortiran = 1;

    }

    if (tek != NULL && sortiran ==
0)tek = tek->sled;

}

if (sortiran == 0) {

    posl->sled = novi;

    novi->pret = posl;

}

}

for (tek = lstP; tek != NULL; tek = tek->sled) {

    //posl = novi;

    posl = tek;

```

```

        }

    }

    else {

        printf("Neispravna datoteka!!!\n");

        system("pause");

        exit(1);

    }

}

else {

    printf("Neispravna datoteka!!!\n");

    system("pause");

    exit(1);

}

}

else {

    printf("Neispravna datoteka!!!\n");

    system("pause");

    exit(1);

}

```

```

    }

    else {

        printf("Neispravna datoteka!!!\n");

        system("pause");

        exit(1);

    }

}

else {

    printf("Neispravna datoteka!!!\n");

    system("pause");

    exit(1);

}

}

else {

    printf("Neispravna datoteka!!!\n");

    system("pause");

    exit(1);

}

```

```

}
else {

```

```

printf("Neispravna datoteka!!!\n");

system("pause");

exit(1);
}
}

else {

printf("Neispravna datoteka!!!\n");

system("pause");

exit(1);
}

}

else {

printf("Neispravna datoteka!!!\n");

system("pause");

exit(1);
}

}

else {

printf("Neispravna
datoteka!!!\n");

system("pause");
exit(1);
}

}

else {

printf("Neispravna
datoteka!!!\n");

```



```
}
```

```
Turing*obrada(Turing*lst, Program*lstP) {
```

```
    Turing *glava;
```

```
    Program *p;
```

```
    int trenutno_stanje_q = 0;
```

```
    glava = lst;
```

```
    p = lstP;
```

```
    //Petlja pozicionira glavu na krajnji levi element.
```

```
    while (glava->pod == ' ') {
```

```
        glava = glava->sled;
```

```
    }
```

```
    while (p != NULL) {
```

```
        if (p->q == trenutno_stanje_q) {
```

```
            if (p->uslov == glava->pod) {
```

```
                //Sada treba da menjamo.
```

```
                glava->pod = p->zamena;
```

```
                //Sada treba odraditi pomeraj glave.
```

```
                if (p->pomeraj_glave == 1) { //Programski namešteno 1 i -1,  
nezavisno od upisa. 1 jeste +1.
```

```
                    glava = glava->sled;
```

```
                }
```

```
            else { // Ako nije 1 onda je sigurno -1.
```

```
                glava = glava->pret;
```

```
            }
```

```
            //Sada bi trebalo videti u koje stanje se ide nakon nastalih zamena.
```

```
            if (p->sl_stanje_q_plus_ili_minus != '*') { // Zvezdica je indikator  
da je stanje broj.
```

```
                if (p->sl_stanje_q_plus_ili_minus == '+') {
```

```
                    printf("Program je uspesno završen!\n");
```

```
                    return lst;
```

```
                }
```

```
                if (p->sl_stanje_q_plus_ili_minus == '-') {
```

```
                    printf("Doslo je do greske u radu programa!\n");
```

```
                    exit(1);
```

```

        }

        }
        else { // Ako nije * onda je sledece stanje broj.
            trenutno_stanje_q = p->sl_stanje_broj;

        }

    }

    else {
        p = p->sled;
    }
}
else {
    p = p->sled;
}
}
return lst;
}

```

```

Turing* obrisi1(Turing *lst) {
    Turing *stari, *tek;
    tek = lst;
    while (tek != NULL) {
        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lst = NULL;
    return lst;
}

```

```

Program* obrisi2(Program *lstP) {
    Program *stari, *tek;
    tek = lstP;
    while (tek != NULL) {

```



```

        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lstP = NULL;
    return lstP;
}

void ispis_trake(Turing*lst) {
    FILE *p;
    Turing *tek;
    char izbor = 'a', izbor1;

    puts(" ");
    printf("Da li zelite da se kreira posebna datoteka u kojoj ce se nalaziti rezultat obrade
ulazne datoteke? \n");
    printf("Molim Vas unesite izbor: 1-DA 0-NE\n");
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    // Zastita od nekorektnog unosa.
    scanf("%c", &izbor);
    if (izbor != '\n') {
        while (izbor != '\n') {
            scanf("%c", &izbor);
            izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo unese.
        }
    }
    while (izbor1 != '1' && izbor1 != '0') {
        printf("Nekorektan unos!!!\n");
        printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");
        while (izbor != '\n')scanf("%c", &izbor);
        printf("Vas izbor: ");
        scanf("%c", &izbor);
        izbor1 = izbor;
        scanf("%c", &izbor);
        while (izbor != '\n') {
            scanf("%c", &izbor);
            izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.
        }
    }
}

```

```

        }
    }
    puts(" ");

    printf("Rezultat (ispisuje se cela traka): ");
    puts("");
    puts("");

    if (izbor1 == '1') {
        p = fopen("izlaz.txt", "w");
        if (p == NULL) {
            printf("Neuspesno kreiranje datoteke!!!\n");
            exit(1);
        }

        for (tek = lst; tek != NULL; tek = tek->sled) {

            fprintf(p, "%c", tek->pod);
            printf("%c", tek->pod);

        }

        fclose(p);

    }
    else {
        for (tek = lst; tek != NULL; tek = tek->sled) {

            printf("%c", tek->pod);

        }
    }
}

```

```

void ispisi_program(Program*lstP) {
    Program*tek;
    puts(" ");
    printf("Program:\n");
}

```

```

puts(" ");
for (tek = lstP; tek != NULL; tek = tek->sled) {
    if (tek->sl_stanje_q_plus_ili_minus != '*') {
        if (tek->pomeraj_glave == 1) {

            printf("f(q%d,%c)=(q%c,%c,+%d)\n", tek->q, tek->uslov, tek-
>sl_stanje_q_plus_ili_minus, tek->zamena, tek->pomeraj_glave);
        }
        else { //Ako nije +1 onda je -1.

            printf("f(q%d,%c)=(q%c,%c,%c)\n", tek->q, tek->uslov, tek-
>sl_stanje_q_plus_ili_minus, tek->zamena, (char)tek->pomeraj_glave);
        }
    }
    else { // Ako je sledece stanje qn a, ne q+ ili q-.
        if (tek->pomeraj_glave == 1) {
            if (tek->sl_stanje_broj == 98 || tek->sl_stanje_broj == 49) tek-
>sl_stanje_broj -= 48;
            printf("f(q%d,%c)=(q%d,%c,+%d)\n", tek->q, tek->uslov, tek-
>sl_stanje_broj, tek->zamena, tek->pomeraj_glave);
        }
        else {
            if (tek->sl_stanje_broj == 98 || tek->sl_stanje_broj == 49) tek-
>sl_stanje_broj -= 48;
            printf("f(q%d,%c)=(q%d,%c,+%c)\n", tek->q, tek->uslov, tek-
>sl_stanje_broj, tek->zamena, (char)tek->pomeraj_glave);
        }
    }
}
}

```

```

int main() {
    Turing *lst;
    Program *lstP;
    lst = NULL;
    lstP = NULL;
    time_t pocetak, kraj;
    time(&pocetak);
    podaci();
}

```

```

    lst = učitavanje_trake(lst);
    lstP = učitavanje_programa(lstP);
    ispisi_program(lstP);
    lst = obrada(lst, lstP);
    ispisi_trake(lst);
    lst = obrisi1(lst);
    lstP = obrisi2(lstP);
    time(&kraj);
    puts("");
    puts("");
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf sekund(i,a,e).\n",
diffime(kraj, pocetak));
    printf("Hvala sto ste koristili ovaj program!\n");
    puts("-----");
-----");
    puts("-----");
-----");
    system("pause");
    return 0;
}

```

TEST PRIMERI

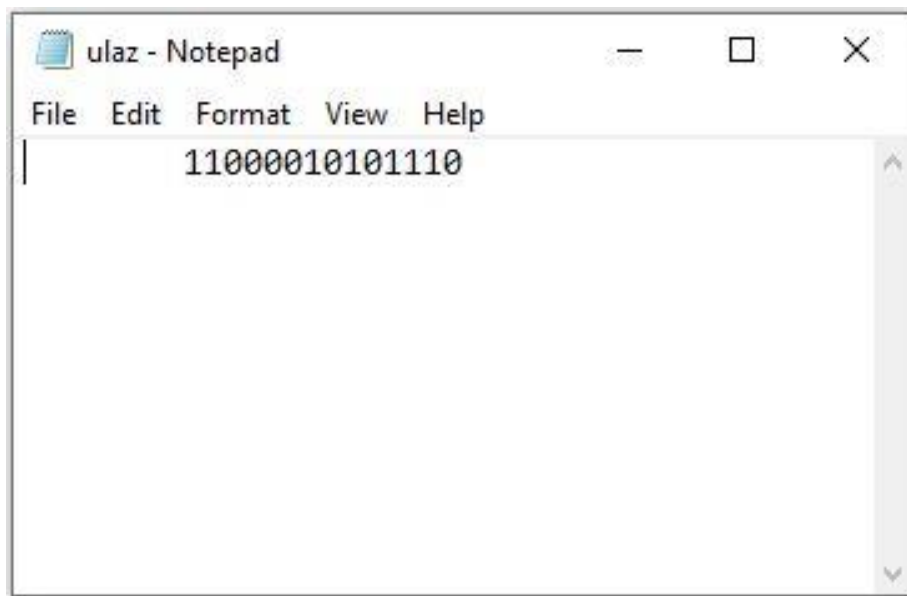
Test primer 1 – prvi komplement zadanog broja

Turing-ova mašina radi sa azbukom $\{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale celije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom $'$ zadanog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se ispisuje prvi komplement datog broja (broj dobijen invertovanjem svakog bita datog broja).

Ovaj program ima trivijalno rešenje(svega dva stanja q_0 i q_1). Sadržaj ulazne datoteke program.txt treba da bude sledeći:

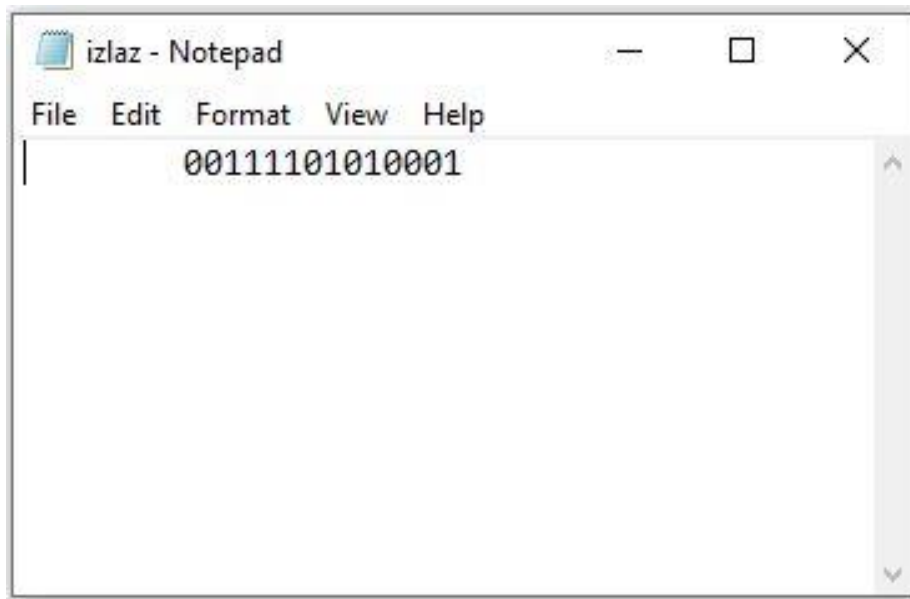
```
f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,1,+1)
f(q0,1)=(q1,0,+1)
f(q1,b)=(q+,b,-1)
f(q1,0)=(q1,1,+1)
f(q1,1)=(q1,0,+1)
```

dok sam sadržaj datoteke ulaz.txt biramo(u njoj zadajemo izgled trake Turing-ove mašine na početku). Zbog preglednosti korišćenja blanko znakova sadržaj ove datoteke je dat na slici 2.



Slika 2 – Sadržaj ulazne datoteke

Nakon pokretanja programa, kao rezultat rada program formira izlaznu datoteku čiji je sadržaj dat na slici 3.



Slika 3 – Sadržaj novoformirane izlazne datoteke nakon obrade

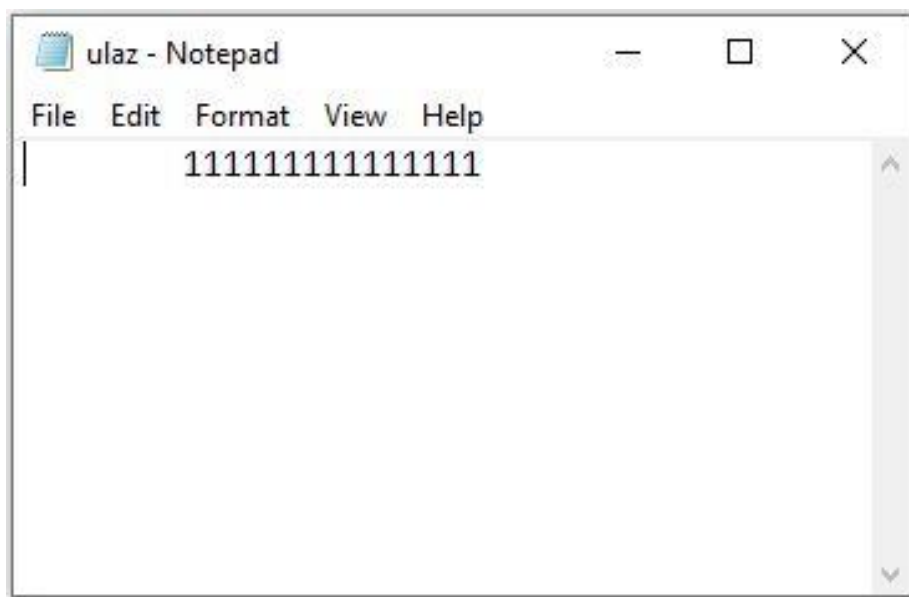
Test primer 2 – šiftovanje za jedno mesto ulevo, azbuka {1,b}

Turing-ova mašina radi sa azbukom {1,b}, gde je b prazan simbol. Neka je broj zadat kao niz jedinica između dva prazna simbola. U sve ostale ćelije trake Turing-ove mašine je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom 1. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se zadati broj pomera za jednu celiju ulevo.

Ovaj program ima trivijalno rešenje(svega tri stanja q_0, q_1, q_2 i q_3). Sadržaj ulazne datoteke program.txt treba da bude sledeći:

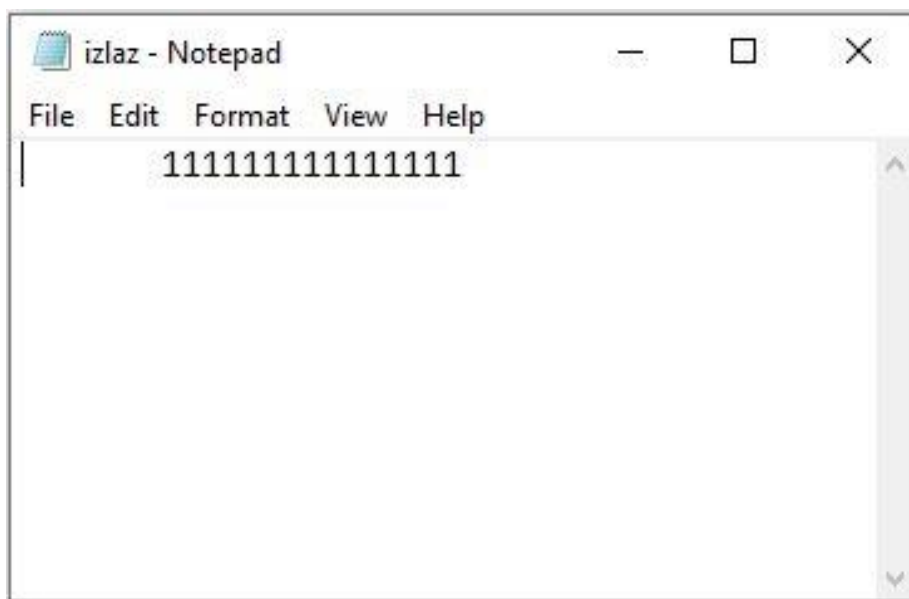
```
f(q0,b)=(q-,b,+1)
f(q0,1)=(q1,1,-1)
f(q1,b)=(q2,1,+1)
f(q1,1)=(q-,1,+1)
f(q2,b)=(q3,b,-1)
f(q2,1)=(q2,1,+1)
f(q3,b)=(q-,b,+1)
f(q3,1)=(q+,b,-1)
```

dok sam sadržaj datoteke ulaz.txt biramo(u njoj zadajemo izgled trake Turing-ove mašine na početku). Zbog preglednosti korišćenja blanko znakova sadržaj ove datoteke je dat na slici 4.



Slika 4 – Sadržaj ulazne datoteke

Nakon pokretanja programa, kao rezultat rada formira se izlazna datoteka čiji je sadržaj dat na slici 5. Slike se lakše mogu porediti poređenjem položaja zadnje jedinice s desne strane i opcije *Help*.



Slika 5 – Sadržaj novoformirane izlazne datoteke nakon obrade

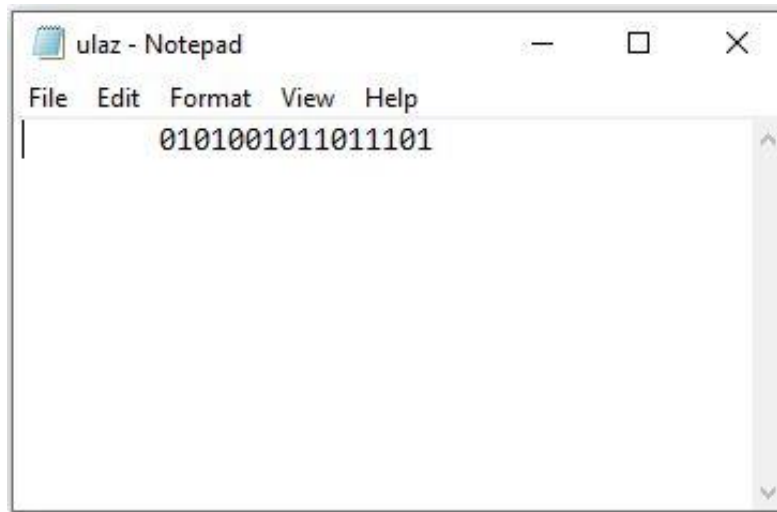
Test primer 3 – šiftovanje za jedno mesto ulevo, azbuka $\{0,1,b\}$

Turing-ova mašina radi sa azbukom $\{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale celije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se zadati broj pomera za jednu celiju ulevo.

Sadržaj ulazne datoteke program.txt treba da bude sledeći:

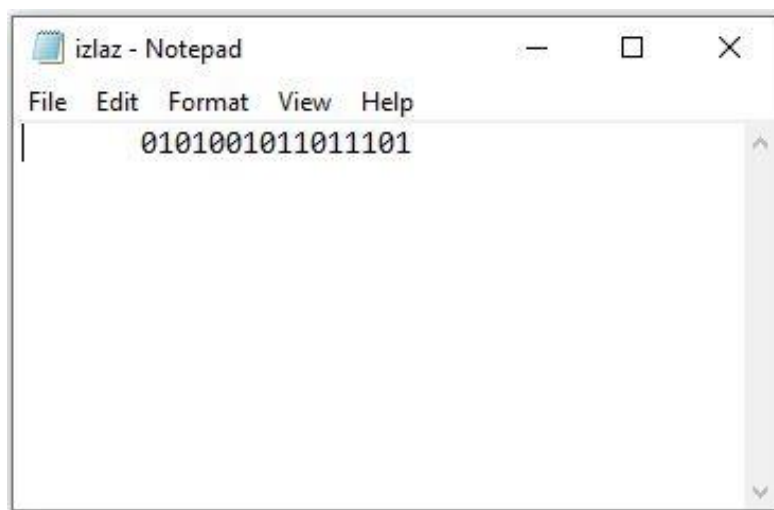
```
f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,b,-1)
f(q0,1)=(q2,b,-1)
f(q1,b)=(q3,0,+1)
f(q1,0)=(q3,0,+1)
f(q1,1)=(q3,0,+1)
f(q2,b)=(q3,1,+1)
f(q2,0)=(q3,1,+1)
f(q2,1)=(q3,1,+1)
f(q3,b)=(q4,b,+1)
f(q3,0)=(q4,b,+1)
f(q3,1)=(q4,b,+1)
f(q4,b)=(q+,b,+1)
f(q4,0)=(q1,b,-1)
f(q4,1)=(q2,b,-1)
```

dok sam sadržaj datoteke ulaz.txt biramo(u njoj zadajemo izgled trake Turing-ove mašine na početku). Zbog preglednosti korišćenja blanko znakova sadržaj ove datoteke je dat na slici 6.



Slika 6 – Sadržaj ulazne datoteke

Nakon pokretanja programa, kao rezultat rada formira se izlazna datoteka čiji je sadržaj dat na slici 7. Slike se lakše mogu porediti poređenjem položaja zadnje jedinice s desne strane i opcije *Help*.



Slika 7 – Sadržaj novoformirane izlazne datoteke nakon obrade

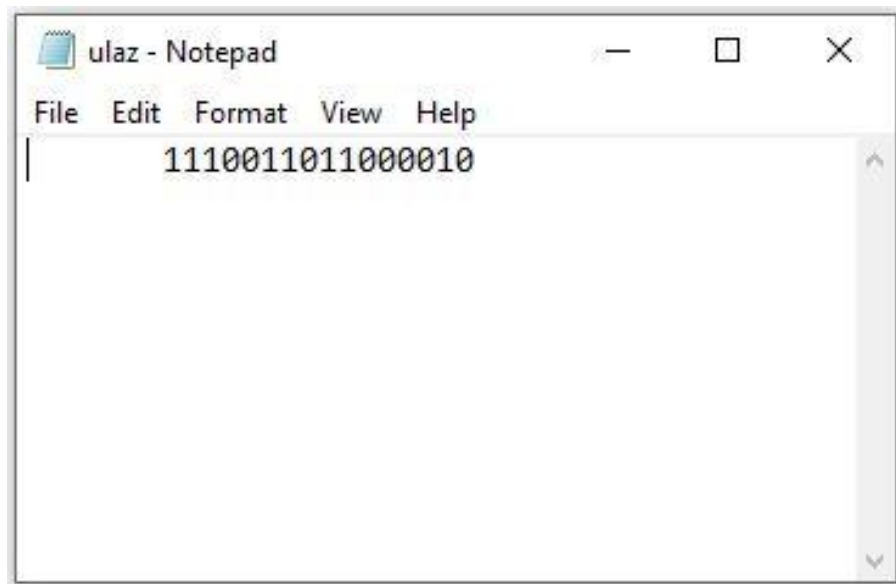
Test primer 4 – dodavanje broja jedan u binarnom sistemu

Turing-ova mašina radi sa azbukom $\{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale celije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se zadatom broju dodaje broj 1 u binarnom sistemu.

Sadržaj ulazne datoteke program.txt treba da bude sledeći:

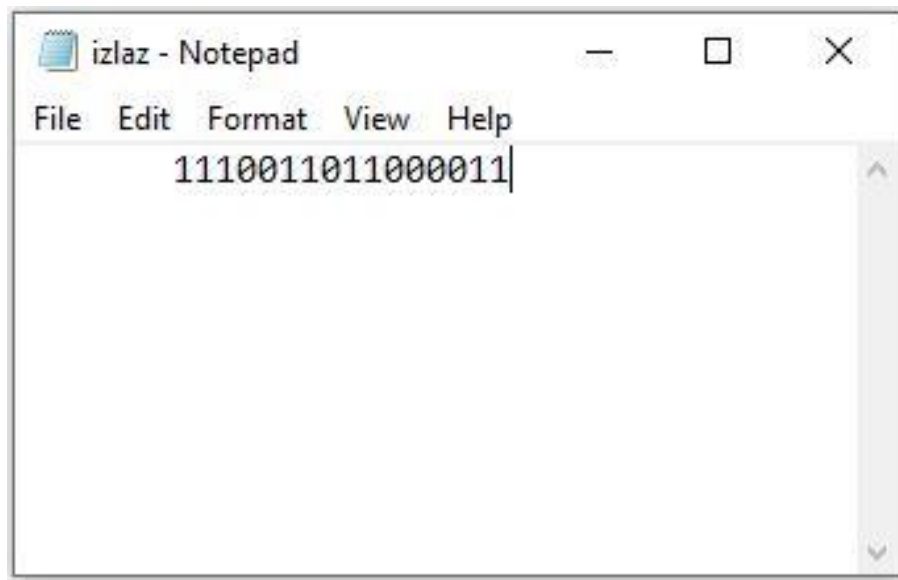
```
f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,0,+1)
f(q0,1)=(q1,1,+1)
f(q1,b)=(q2,b,-1)
f(q1,0)=(q1,0,+1)
f(q1,1)=(q1,1,+1)
f(q2,b)=(q+,1,+1)
f(q2,0)=(q+,1,+1)
f(q2,1)=(q2,0,-1)
```

dok sam sadržaj datoteke ulaz.txt biramo(u njoj zadajemo izgled trake Turing-ove mašine na početku). Zbog preglednosti korišćenja blanko znakova sadržaj ove datoteke je dat na slici 8.



Slika 8 – Sadržaj ulazne datoteke

Nakon pokretanja programa, kao rezultat rada formira se izlazna datoteka čiji je sadržaj dat na slici 9.



Slika 9 – Sadržaj novoformirane izlazne datoteke nakon obrade

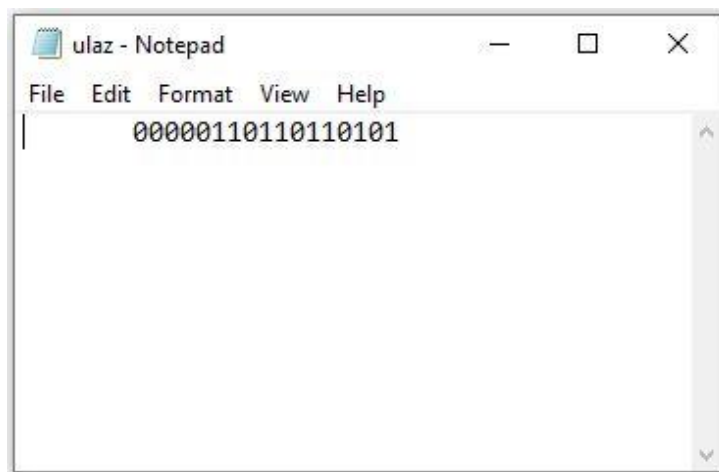
Test primer 5 – oduzimanje broja jedan u binarnom sistemu

Turing-ova mašina radi sa azbukom $\{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale ćelije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom \cdot zadatog broja. Konstruisati program za Turing-ovu mašinu $f: Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se od zadatog broja oduzima broj 1 u binarnom sistemu. Ukoliko je zadati broj nula, kao rezultat ispisuje se znak 0.

Sadržaj ulazne datoteke program.txt treba da bude sledeći:

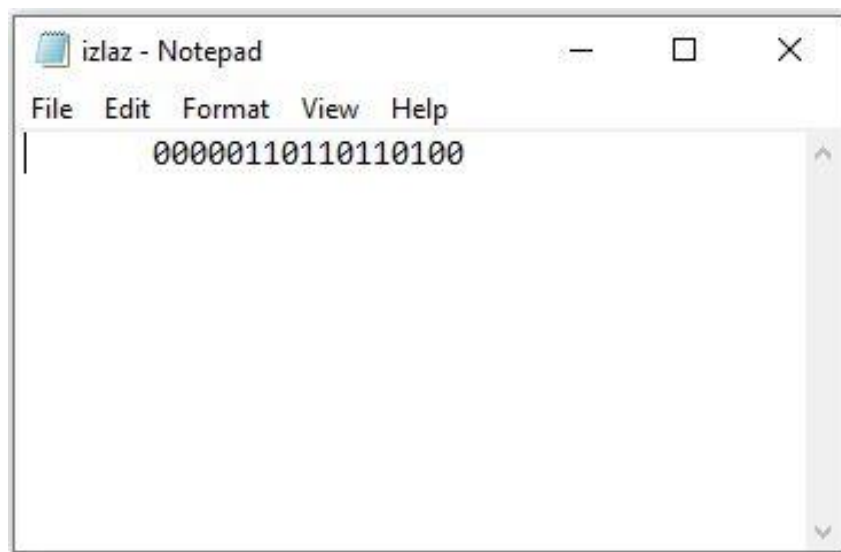
```
f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,0,+1)
f(q0,1)=(q1,1,+1)
f(q1,b)=(q2,b,-1)
f(q1,0)=(q1,0,+1)
f(q1,1)=(q1,1,+1)
f(q2,b)=(q-,b,+1)
f(q2,0)=(q3,1,-1)
f(q2,1)=(q+,0,+1)
f(q3,b)=(q4,b,+1)
f(q3,0)=(q3,1,-1)
f(q3,1)=(q+,0,+1)
f(q4,b)=(q+,0,+1)
f(q4,0)=(q-,0,+1)
f(q4,1)=(q4,b,+1)
```

dok sam sadržaj datoteke ulaz.txt biramo(u njoj zadajemo izgled trake Turing-ove mašine na početku). Zbog preglednosti korišćenja blanko znakova sadržaj ove datoteke je dat na slici 10.



Slika 10 – Sadržaj ulazne datoteke

Nakon pokretanja programa, kao rezultat rada formira se izlazna datoteka čiji je sadržaj dat na slici 11.



Slika 11 – Sadržaj ulazne datoteke

PRILOG

Specijalizovani programi

Programi u ovom prilogu su programirani tako da rade isključivo jednu radnju. Prethodno dati i testirani glavni program je univerzalni i radi bilo šta dok su ovde navedeni programi usko specijalizovani na samo jednu radnju. Ovaj deo je prilog i ne spada u domen obrađivanog projekta koji je za cilj imao programiranje univerzalne Turing-ove mašine koja radi u svim mogućim slučajevima i programima.

U nastavku će biti dati samo konkretni programski kodovi za čije funkcionisanje važe isti uslovi kao i za sve prethodne test primere sa izuzetkom da se na nazive ulaznih datoteka dodaje broj u zavisnosti koji se test primer pokreće (npr. ulaz1, ulaz2, ulaz3...) o čemu je dato uputstvo u samom programskom kodu svakog programa.

Program 1 – nula funkcije

Turing-ova mašina radi sa azbukom $\{1, b\}$, gde je b prazan simbol. Neka je broj $n \in \mathbb{N}_0$ zadat kao niz od $n+1$ jedinica između dva prazna simbola. U sve ostale ćelije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ koji racuna nula funkciju $\chi_N : \mathbb{N}_0 \rightarrow \mathbb{N}_0$, $N(n) = 0$.

Programirani kod rešenja izvršava sledeće:

$$\begin{aligned} f(q_0, b) &= (q_-, b, +1) \\ f(q_0, 1) &= (q_1, 1, +1) \\ f(q_1, b) &= (q_+, b, +1) \\ f(q_1, 1) &= (q_1, b, +1). \end{aligned}$$

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

```
typedef struct turing {
    char pod;
    struct turing *sled;
} Turing;
```

```
void podaci() {
    time_t vreme;
```

```

time(&vreme);
puts("-----");
-----");
puts("-----");
-----");
printf("Univerzitet u Beogradu\n");
printf("Elektrotehnicki fakultet Beograd\n");
printf("Bulevar kralja Aleksandra 73\n");
puts(" ");
printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
printf("Razlog: predispitne obaveze\n");
printf("Format: rad jednog studenta\n");
printf("Tema: Racunarska realizacija Turing-ove masine\n");
printf("Test primer [1/5]: nula funkcija\n");
printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
printf("Student: Nemanja Karapetrovic [broj indeksa: ****/****]\n");
printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz1.txt i da se nalazi gde i
izvorni fajl TuringMachine1.c .\n");
printf("Vreme: %s", ctime(&vreme));
puts("-----");
-----");
puts("-----");
-----");
puts("");
}

```

```

Turing* učitavanje(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz1.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspelo!\n");
        exit(1);
    }
    while (fscanf(p, "%c", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
    }
}

```



```

    }
    novi->pod = pod1;
    novi->sled = NULL;
    if (lst == NULL) {
        lst = novi;
    }
    else {
        posl->sled = novi;
    }
    posl = novi;
}
fclose(p);
return lst;
}

```

```

Turing* obrada(Turing *lst) {
    Turing *glava;
    glava = lst;

    /*          f(q0,b)=(q-,b,+1)
                f(q0,1)=(q1,1,+1)
                f(q1,b)=(q+,b,+1)
                f(q1,1)=(q1,b,+1)  */

    while (glava->pod == ' ') {
        glava = glava->sled; // Ova petlja ce pozicionirati pokazivac glava na krajnji levi
znak 1. Ovim se izbegava q- stanje.
    }

    /*  q0 stanje => f(q0,b)=(q-,b,+1)
                                   f(q0,1)=(q1,1,+1)  */

    if (glava->pod == '1') {
        glava->pod = '1';
        glava = glava->sled;
    }
    else {
        printf("ERROR!!!\n"); // q- stanje
        if (glava->pod == ' ') glava->pod = ' ';
        glava = glava->sled;
        exit(1); // exit jer ako pocetak nije dobar, nema sta da se radi.
    }
}

```

```

    }

    /* stanje q1 =>f(q1,b)=(q+,b,+1)
       f(q1,1)=(q1,b,+1)    */

    while (glava->pod != ' ') {
        if (glava->pod == '1') {
            glava->pod = ' ';
            glava = glava->sled;
        }
    }
    return lst;
}

void ispis(Turing *lst) {
    Turing *tek;
    FILE *p;
    char izbor = 'a', izbor1;

    printf("Program:\n");

    printf("f(q0,b)=(q-,b,+1)\n");
    printf("f(q0,1)=(q1,1,+1)\n");
    printf("f(q1,b)=(q+,b,+1)\n");
    printf("f(q1,1)=(q1,b,+1)\n");

    puts(" ");

    printf("Da li zelite da se kreira posebna datoteka u kojoj ce se nalaziti rezultat obrade
    ulazne datoteke? \n");
    printf("Molim Vas unesite izbor: 1-DA  0-NE\n");
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    // Zastita od nekorektnog unosa.
    scanf("%c", &izbor);
    if (izbor != '\n') {
        while (izbor != '\n') {

```

```

        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo unese.
    }
}
while (izbor1 != '1' && izbor1 != '0') {
    printf("Nekorektan unos!!!\n");
    printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");
    while (izbor != '\n')scanf("%c", &izbor);
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    scanf("%c", &izbor);
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.
    }
}
puts(" ");

printf("Rezultat: ");
if (izbor1 == '1') {
    p = fopen("izlaz1.txt", "w");
    if (p == NULL) {
        printf("Neuspesno kreiranje datoteke!!!\n");
        exit(1);
    }
    for (tek = lst; tek != NULL; tek = tek->sled) {
        fprintf(p, "%c", tek->pod);
        printf("%c", tek->pod);
    }
    fclose(p);
}
else {
    for (tek = lst; tek != NULL; tek = tek->sled) {
        printf("%c", tek->pod);
    }
}

```

```
        if (izbor1 == '1') printf("Vasa datoteka je uspesno formirana. Njen naziv je  
izlaz1.txt. Nalazi se na adresi izvornog fajla TuringMachine1.c .\n");
```

```
        puts(" ");  
        puts(" ");  
    }  
}
```

```
Turing* obrisi(Turing *lst) {  
    Turing *stari, *tek;  
    tek = lst;  
    while (tek != NULL) {  
        stari = tek;  
        tek = tek->sled;  
        free(stari);  
    }  
    lst = NULL;  
    return lst;  
}
```

```
int main() {  
    Turing *lst;  
    lst = NULL;  
    time_t pocetak, kraj;  
    time(&pocetak);  
    podaci();  
    lst = učitavanje(lst);  
    lst = obrada(lst);  
    ispis(lst);  
    lst = obrisi(lst);  
    time(&kraj);  
    puts(" ");  
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf sekund(i,a,e).\n",  
diffime(kraj, pocetak));  
    printf("Hvala sto ste koristili ovaj program!\n");  
    puts("-----");  
    puts("-----");  
    puts("-----");  
    system("pause");
```

```

    return 0;
}

```

Program 2 – funkcija sledbenik

Turing-ova mašina radi sa azbukom $\{1, b\}$, gde je b prazan simbol. Neka je broj $n \in \mathbb{N}_0$ zadat kao niz od $n+1$ jedinica između dva prazna simbola. U sve ostale ćelije je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f: Q \times S \rightarrow Q \times S \times \{+1, -1\}$ koji racuna funkciju sledbenik $S: \mathbb{N}_0 \rightarrow \mathbb{N}_0$, $S(n) = n+1$.

Programirani kod rešenja izvršava sledeće:

```

f(q0, b) = (q-, b, +1)
f(q0, 1) = (q1, 1, +1)
f(q1, b) = (q+, 1, +1)
f(q1, 1) = (q1, 1, +1).

```

```

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

```

```

typedef struct turing {
    char pod;
    struct turing *sled;
} Turing;

```

```

void podaci() {
    time_t vreme;
    time(&vreme);
    puts("-----");

```

```

-----");
    puts("-----");
    -----");

```

```

printf("Univerzitet u Beogradu\n");
printf("Elektrotehnicki fakultet Beograd\n");
printf("Bulevar kralja Aleksandra 73\n");
puts(" ");
printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
printf("Razlog: predispitne obaveze\n");
printf("Format: rad jednog studenta\n");
printf("Tema: Racunarska realizacija Turing-ove masine\n");

```

```

printf("Test primer [2/5]: funkcija sledbenik\n");
printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
printf("Student: Nemanja Karapetrovic [broj indeksa: *****/****]\n");
printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz2.txt i da se nalazi
gde i izvorni fajl TuringMachine2.c .\n");
printf("Vreme: %s", ctime(&vreme));
puts("-----");
-----");
puts("-----");
-----");
puts("");
}

```

```

Turing* učitavanje(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz2.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspelo!\n");
        exit(1);
    }
    while (fscanf(p, "%c", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
        novi->pod = pod1;
        novi->sled = NULL;
        if (lst == NULL) {
            lst = novi;
        }
        else {
            posl->sled = novi;
        }
        posl = novi;
    }
    fclose(p);
    return lst;
}

```

```

Turing* obrada(Turing *lst) {
    Turing *glava;
    glava = lst;
}

```

while (glava->pod == ' ') glava = glava->sled; // Petlja pozicionira glavu Turing-ove masine na krajnji levi simbol koji nije blanko znak.

```
/*    q0 =>          f(q0,b)=(q-,b,+1)
                        f(q0,1)=(q1,1,+1)
```

```
*/
if (glava->pod == ' ') {
    printf("ERROR!!!\n");
    glava->pod = ' ';
    glava = glava->sled;
}
else {
    glava->pod = '1';
    glava = glava->sled;
}
```

```
/*          q1 =>          f(q1,b)=(q+,1,+1)
                        f(q1,1)=(q1,1,+1)
```

```
*/
```

```
while (glava->pod != ' ') {
    if (glava->pod == '1') {
        glava->pod = '1';
        glava = glava->sled;
    }
    glava = glava->sled; // Pokazivac glava se pozicionirao na krajnju desnu
    poziciju.
}
```

```
return lst; //q+ stanje
```

```
}
```

```
void ispis(Turing *lst) {
    Turing *tek;
    FILE *p;
    char izbor = 'a', izbor1;
```

```
printf("Program:\n");
```

```

printf("f(q0,b)=(q-,b,+1)\n");
printf("f(q0,1)=(q1,1,+1)\n");
printf("f(q1,b)=(q+,1,+1)\n");
printf("f(q1,1)=(q1,1,+1)\n");

```

```

puts(" ");

```

obrade ulazne datoteke? \n");

```

printf("Molim Vas unesite izbor: 1-DA 0-NE\n");

```

```

printf("Vas izbor: ");

```

```

scanf("%c", &izbor);

```

```

izbor1 = izbor;

```

```

// Zastita od nekorektnog unosa.

```

```

scanf("%c", &izbor);

```

```

if (izbor != '\n') {

```

```

    while (izbor != '\n') {

```

```

        scanf("%c", &izbor);

```

```

        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo

```

unesu.

```

    }

```

```

}

```

```

while (izbor1 != '1' && izbor1 != '0') {

```

```

    printf("Nekorektan unos!!!\n");

```

```

    printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");

```

```

    while (izbor != '\n')scanf("%c", &izbor);

```

```

    printf("Vas izbor: ");

```

```

    scanf("%c", &izbor);

```

```

    izbor1 = izbor;

```

```

    scanf("%c", &izbor);

```

```

    while (izbor != '\n') {

```

```

        scanf("%c", &izbor);

```

```

        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.

```

```

    }

```

```

}

```

```

puts(" ");

```

```

printf("Rezultat: ");

```

```

if (izbor1 == '1') {

```

```

    p = fopen("izlaz2.txt", "w");

```

```

    if (p == NULL) {

```

```

        printf("Neuspesno kreiranje datoteke!!!\n");

```

```

        exit(1);

```

```

    }

```

```

    for (tek = lst; tek != NULL; tek = tek->sled) {

```

```

        fprintf(p, "%c", tek->pod);

```



```

        printf("%c", tek->pod);
    }
    fclose(p);
}
else {
    for (tek = lst; tek != NULL; tek = tek->sled) {
        printf("%c", tek->pod);
    }

}
puts(" ");
if (izbor1 == '1') printf("Vasa datoteka je uspesno formirana. Njen naziv je
izlaz2.txt. Nalazi se na adresi izvornog fajla TuringMachine2.c .\n");
}

```

```

Turing* obrisi(Turing *lst) {
    Turing *stari, *tek;
    tek = lst;
    while (tek != NULL) {
        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lst = NULL;
    return lst;
}

```

```

int main() {
    Turing *lst;
    lst = NULL;
    time_t pocetak, kraj;
    time(&pocetak);
    podaci();
    lst = učitavanje(lst);
    lst = obrada(lst);
    ispis(lst);
    lst = obrisi(lst);
    time(&kraj);
    puts(" ");
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf
sekund(i,a,e).\n", difftime(kraj, pocetak));
    printf("Hvala sto ste koristili ovaj program!\n");
    puts("-----");
    puts("-----");
    puts("-----");
}

```

```

    system("pause");
    return 0;
}

```

Program 3 – funkcija projekcije na prvu koordinatu

Turing-ova mašina radi sa azbukom $\{0,1,b\}$, gde je b prazan simbol. Neka je na traci Turingove mašine dat ureden par brojeva $(n,m) \in \mathbb{N}^2$, i to tako što je broj n zapisan kao $n+1$ jedinica, zatim je zapisana nula, i broj m koji je predstavljen sa $m+1$ jedinica. U sve ostale celije trake Turing-ove mašine je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom 1. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ koji racuna (a) funkciju projekcije na prvu koordinatu $U_1^2: \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$, $U_1^2(n,m) = n$;

Programirani kod rešenja izvršava sledeće:

```

f(q0,b)=(q-,b,+1)
f(q0,0)=(q-,0,+1)
f(q0,1)=(q1,1,+1)
f(q1,b)=(q-,b,+1)
f(q1,0)=(q2,b,+1)
f(q1,1)=(q1,1,+1)
f(q2,b)=(q+,b,-1)
f(q2,0)=(q-,0,+1)
f(q2,1)=(q2,b,+1).

```

```

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

```

```

typedef struct turing {
    char pod;
    struct turing *sled;
} Turing;

```

```

void podaci() {
    time_t vreme;
    time(&vreme);
    puts("-----");
    -----");
}

```

```

puts("-----");
-----");
printf("                                Univerzitet u Beogradu\n");
printf("                                Elektrotehnicki fakultet Beograd\n");
printf("                                Bulevar kralja Aleksandra 73\n");
puts(" ");
printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
printf("Razlog: predispitne obaveze\n");
printf("Format: rad jednog studenta\n");
printf("Tema: Racunarska realizacija Turing-ove masine\n");
printf("Test primer [3/5]: Funkcija projekcije na prvu koordinatu\n");
printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
printf("Student: Nemanja Karapetrovic [broj indeksa: ****/****]\n");
printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz3.txt i da se nalazi
gde i izvorni fajl TuringMachine3.c .\n");
printf("Vreme: %s", ctime(&vreme));
puts("-----");
-----");
-----");
puts("");
}

```

```

Turing* učitavanje(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz3.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspelo!\n");
        exit(1);
    }
    while (fscanf(p, "%c", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
        novi->pod = pod1;
        novi->sled = NULL;
        if (lst == NULL) {
            lst = novi;
        }
        else {
            posl->sled = novi;
        }
    }
}

```

```

        posl = novi;
    }
    fclose(p);
    return lst;
}

Turing* obrada(Turing *lst) {
    Turing *glava;
    glava = lst;
    while(glava->pod==' ') glava=glava->sled; // Petlja pozicionira glavu na krajnji
levi simbol koji nije blanko znak.
    int q1 = 0; /* Indikator: q1=1, program je bio u q1 stanju
                    q1=0, program nije bio u q1 stanju */

    /* Stanje q0 => f(q0,0)=(q-,0,+1);
                    f(q0,1)=(q1,1,+1); */

    // Kroz stanje q0 se neminovno prolazi pa indikator nije
poteban kao za stanje q1.

    if (glava->pod == '0') {
        printf("ERROR!!!"); //q- stanje
        glava->pod = '0';
        glava = glava->sled;
        exit(1);
    }
    else { //f(q0,1)=(q1,1,+1);
        glava->pod = '1';
        glava = glava->sled;
    }

    /* Stanje q1 => f(q1,0)=(q2,b,+1);
                    f(q1,1)=(q1,1,+1); */

    // f(q1,0)=(q2,b,+1);

    while (glava != NULL) {
        if (q1 == 0) {
            // f(q1,b)=(q-,b,+1)
            if (glava->pod == ' ') {
                printf("ERROR!!!"); //q- stanje
                glava->pod = '0';
                glava = glava->sled;
                exit(1);
            }
        }
    }
}

```

```

        if (glava->pod == '0') {
            glava->pod = ' ';
            glava = glava->sled;
            q1 = 1; // Prelazi se u q2 stanje.
        }
        else { // f(q1,1)=(q1,1,+1)
            glava->pod = '1';
            glava = glava->sled;
        }

    }

    /* Stanje q2 => f(q2,0)=(q-,0,+1);
       f(q2,1)=(q2,b,+1); */
    if (q1 == 1) {

        if (glava->pod == '0') {
            printf("ERROR!!!"); //q- stanje
            glava->pod = '0';
            glava = glava->sled;
            exit(1);
        }
        else { //f(q2,1)=(q2,b,+1);
            glava->pod = ' ';
            glava = glava->sled;
        }

    }

}
return lst; // q+ stanje
}

void ispis(Turing *lst) {
    Turing *tek;
    FILE *p;
    char izbor = 'a', izbor1;

    printf("Program:\n");
    printf("f(q0,b)=(q-,b,+1)\n");
    printf("f(q0,0)=(q-,0,+1)\n");
    printf("f(q0,1)=(q1,1,+1)\n");
    printf("f(q1,b)=(q-,b,+1)\n");
    printf("f(q1,0)=(q2,b,+1)\n");

```

```

printf("f(q1,1)=(q1,1,+1)\n");
printf("f(q2,b)=(q+,b,-1)\n");
printf("f(q2,0)=(q-,0,+1)\n");
printf("f(q2,1)=(q2,b,+1)\n");
puts(" ");

printf("Da li zelite da se kreira posebna datoteka u kojoj ce se nalaziti rezultat
obrade ulazne datoteke? \n");
printf("Molim Vas unesite izbor: 1-DA 0-NE\n");
printf("Vas izbor: ");
scanf("%c", &izbor);
izbor1 = izbor;
// Zastita od nekorektnog unosa.
scanf("%c", &izbor);
if (izbor != '\n') {
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo
unese.
    }
}
while (izbor1 != '1' && izbor1 != '0') {
    printf("Nekorektan unos!!!\n");
    printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");
    while (izbor != '\n')scanf("%c", &izbor);
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    scanf("%c", &izbor);
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.
    }
}
puts(" ");

printf("Rezultat: ");
if (izbor1 == '1') {
    p = fopen("izlaz3.txt", "w");
    if (p == NULL) {
        printf("Neuspesno kreiranje datoteke!!!\n");
        exit(1);
    }
    for (tek = lst; tek != NULL; tek = tek->sled) {
        fprintf(p, "%c", tek->pod);
        printf("%c", tek->pod);
    }
}

```

```

        }
        fclose(p);
    }
    else {
        for (tek = lst; tek != NULL; tek = tek->sled) {
            printf("%c", tek->pod);
        }

    }

    puts(" ");
    if (izbor1 == '1') printf("Vasa datoteka je uspesno formirana. Njen naziv je
izlaz3.txt. Nalazi se na adresi izvornog fajla TuringMachine3.c .\n");
}

```

```

Turing* obrisi(Turing *lst) {
    Turing *stari, *tek;
    tek = lst;
    while (tek != NULL) {
        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lst = NULL;
    return lst;
}

```

```

int main() {
    Turing *lst;
    lst = NULL;
    time_t pocetak, kraj;
    time(&pocetak);
    podaci();
    lst = učitavanje(lst);
    lst = obrada(lst);
    ispis(lst);
    lst = obrisi(lst);
    time(&kraj);
    puts(" ");
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf
sekund(i,a,e).\n", difftime(kraj, pocetak));
    printf("Hvala sto ste koristili ovaj program!\n");
    puts("-----");
    puts("-----");
    puts("-----");
}

```

```

        system("pause");
        return 0;
    }

```

Program 4 – deljivost zadatog broja sa 4

Turing-ova mašina radi sa azbukom $S = \{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale ćelije trake je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se utvrđuje da li je zadati broj deljiv sa 4.

Broj je deljiv sa 4 ako i samo ako su poslednje dve cifre u njegovom binarnom zapisu nula. Program Turing-ove mašine $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$, $Q = \{q_0, q_1, q_2, q_-, q_+\}$, dat je nizom instrukcija:

```

f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,0,+1)
f(q0,1)=(q1,1,+1)
f(q1,b)=(q2,b,-1)
f(q1,0)=(q1,0,+1)
f(q1,1)=(q1,1,+1)
f(q2,b)=(q-,b,+1)
f(q2,0)=(q3,0,-1)
f(q2,1)=(q-,1,+1)
f(q3,b)=(q-,b,+1)
f(q3,0)=(q+,0,+1)
f(q3,1)=(q-,1,+1).

```

```

#include<stdio.h>
#include<time.h>
#include<stdlib.h>
#include<string.h>

typedef struct turing {
    char pod;
    struct turing *pret;
    struct turing *sled;
} Turing;

void podaci() {
    time_t vreme;
    time(&vreme);
}

```



```

        puts("-----");
        puts("-----");
        printf("Univerzitet u Beogradu\n");
        printf("Elektrotehnicki fakultet Beograd\n");
        printf("Bulevar kralja Aleksandra 73\n");
        puts(" ");
        printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
        printf("Razlog: predispitne obaveze\n");
        printf("Format: rad jednog studenta\n");
        printf("Tema: Racunarska realizacija Turing-ove masine\n");
        printf("Test primer [4/5]: m+n\n");
        printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
        printf("Student: Nemanja Karapetrovic [broj indeksa: ****/****]\n");
        printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz4.txt i da se nalazi
gde i izvorni fajl TuringMachine4.c .\n");
        printf("Vreme: %s", ctime(&vreme));
        puts("-----");
        puts("-----");
        puts("");
    }

```

```

Turing* učitavanje(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz4.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspelo!\n");
        exit(1);
    }
    while (fscanf(p, "%c ", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
        novi->pod = pod1;
        novi->sled = NULL;
        if (lst == NULL) {
            lst = novi;
        }
        else {
            posl->sled = novi;

```

```

        novi->pret = posl;
    }
    posl = novi;
}
fclose(p);
return lst;
}

Turing* obrada(Turing *lst) {
    Turing *glava;
    glava = lst;
    int q1 = 0; /* Indikator: q1=1, program je bio u q1 stanju
                  q1=0, program nije bio u q1 stanju */

    /* Stranje q0 => f(q0,b)=(q-,b,+1);
       f(q0,0)=(q-,0,+1);
       f(q0,1)=(q1,b,+1); */

    // Kroz stanje q0 se neminovno prolazi pa indikator nije
    potreban kao za stanje q1.

    while (glava->pod == ' ') {
        glava = glava->sled; // Ova petlja ce pozicionirati pokazivac glava na
        krajnji levi znak 1.
    }

    /* Stranje q0 => f(q0,b)=(q-,b,+1);
       f(q0,0)=(q-,0,+1); */

    if (glava->pod == '0' || glava->pod == ' ') {
        printf("ERROR!!!"); //q- stanje
        if (glava->pod == ' ') glava->pod = ' ';
        if (glava->pod == '0') glava->pod = '0';
        glava = glava->sled;
        exit(1);
    }
    else { //f(q0,1)=(q1,b,+1);
        glava->pod = ' ';
        glava = glava->sled;
    }

    /* Stranje q1 => f(q1,b)=(q2,b,-1);
       f(q1,0)=(q1,1,+1);
       f(q1,1)=(q1,1,+1); */

```

```

// f(q1,b)=(q2,b,-1);
while (glava != NULL) {
    if (q1 == 0) {
        if (glava->pod == ' ') {
            glava->pod = '1';
            glava = glava->pret;
            q1 = 1;
        }
        else { /* f(q1,0)=(q1,1,+1)
                f(q1,1)=(q1,1,+1) */
            glava->pod = '1';
            glava = glava->sled;
        }
    }

    /* Stranje q2 => f(q2,b)=(q-,b,+1);
       f(q2,0)=(q-,0,+1);
       f(q2,1)=(q2,b,+1); */
    if (q1 == 1) {
        if (glava->pod == '0' || glava->pod == ' ') {
            printf("ERROR!!!"); //q- stanje
            if (glava->pod == ' ') glava->pod = '1';
            if (glava->pod == '0') glava->pod = '0';
            glava = glava->sled;
            exit(1);
        }
        else { //f(q2,1)=(q2,b,+1);
            glava->pod = '1';
            glava = glava->sled;
        }
    }
}
return lst;
}

void ispis(Turing *lst) {
    Turing *tek;
    FILE *p;
    char izbor = 'a', izbor1;

    printf("Program:\n");

    printf("f(q0,b)=(q-,b,+1)\n");
    printf("f(q0,0)=(q-,0,+1)\n");

```

```

printf("f(q0,1)=(q1,b,+1)\n");

printf("f(q1,b)=(q2,b,-1)\n");
printf("f(q1,0)=(q1,1,+1)\n");
printf("f(q1,1)=(q1,1,+1)\n");

printf("f(q2,b)=(q-,b,+1)\n");
printf("f(q2,0)=(q-,0,+1)\n");
printf("f(q2,1)=(q+,b,-1)\n");

puts(" ");

printf("Da li zelite da se kreira posebna datoteka u kojoj ce se nalaziti rezultat
obrade ulazne datoteke? \n");
printf("Molim Vas unesite izbor: 1-DA 0-NE\n");
printf("Vas izbor: ");
scanf("%c", &izbor);
izbor1 = izbor;
// Zastita od nekorektnog unosa.
scanf("%c", &izbor);
if (izbor != '\n') {
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo
unese.
    }
}
while (izbor1 != '1' && izbor1 != '0') {
    printf("Nekorektan unos!!!\n");
    printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");
    while (izbor != '\n')scanf("%c", &izbor);
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    scanf("%c", &izbor);
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.
    }
}
puts(" ");

printf("Rezultat: ");
if (izbor1 == '1') {
    p = fopen("izlaz4.txt", "w");
    if (p == NULL) {
        printf("Neuspesno kreiranje datoteke!!!\n");

```

```

        exit(1);
    }
    for (tek = lst; tek != NULL; tek = tek->sled) {
        fprintf(p, "%c", tek->pod);
        printf("%c", tek->pod);

    }
    fclose(p);
}
else {
    for (tek = lst; tek != NULL; tek = tek->sled) {
        printf("%c", tek->pod);
    }

}
puts(" ");
if (izbor1 == '1') printf("Vasa datoteka je uspesno formirana. Njen naziv je
izlaz4.txt. Nalazi se na adresi izvornog fajla TuringMachine4.c .\n");
}

```

```

Turing* obrisi(Turing *lst) {
    Turing *stari, *tek;
    tek = lst;
    while (tek != NULL) {
        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lst = NULL;
    return lst;
}

```

```

int main() {
    Turing *lst;
    lst = NULL;
    time_t pocetak, kraj;
    time(&pocetak);
    podaci();
    lst = učitavanje(lst);
    lst = obrada(lst);
    ispis(lst);
    lst = obrisi(lst);
    time(&kraj);
    puts(" ");
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf
sekund(i,a,e).\n", difftime(kraj, pocetak));
    printf("Hvala sto ste koristili ovaj program!\n");
}

```

```
        puts("-----");  
-----");  
-----");  
    system("pause");  
    return 0;  
}
```

Program 5 – deljivost zadatog broja sa 8

Turing-ova mašina radi sa azbukom $S = \{0,1,b\}$, gde je b prazan simbol. Neka se na traci Turing-ove mašine prirodni broj predstavlja svojim binarnim zapisom, između dva prazna simbola. U sve ostale ćelije trake je upisan prazan simbol. Neka se glava mašine nalazi nad krajnjim levim znakom zadatog broja. Konstruisati program za Turing-ovu mašinu $f : Q \times S \rightarrow Q \times S \times \{+1, -1\}$ kojim se utvrđuje da li je zadati broj deljiv sa 8.

Programirani kod rešenja izvršava sledeće:

$$\begin{aligned} f(q_0, b) &= (q_-, b, +1) \\ f(q_0, 0) &= (q_1, 0, +1) \\ f(q_0, 1) &= (q_1, 1, +1) \\ f(q_1, b) &= (q_2, b, -1) \\ f(q_1, 0) &= (q_1, 0, +1) \\ f(q_1, 1) &= (q_1, 1, +1) \\ f(q_2, b) &= (q_-, b, +1) \\ f(q_2, 0) &= (q_3, 0, -1) \\ f(q_2, 1) &= (q_-, 1, +1) \\ f(q_3, b) &= (q_-, b, +1) \\ f(q_3, 0) &= (q_4, 0, -1) \\ f(q_3, 1) &= (q_-, 1, +1) \\ f(q_4, b) &= (q_-, b, +1) \\ f(q_4, 0) &= (q_+, 0, +1) \\ f(q_4, 1) &= (q_-, 1, +1). \end{aligned}$$

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

```
typedef struct turing {
    char pod;
    struct turing *pret;
    struct turing *sled;
} Turing;
```

```
void podaci() {
    time_t vreme;
    time(&vreme);
    puts("-----");
    puts("-----");
    printf("
    printf("
```

Univerzitet u Beogradu\n");
Elektrotehnicki fakultet Beograd\n");

```

printf("                                Bulevar kralja Aleksandra 73\n");
puts(" ");
printf("Predmet: Numericka analiza i diskretna matematika [13E082NAD]\n");
printf("Razlog: predispitne obaveze\n");
printf("Format: rad jednog studenta\n");
printf("Tema: Racunarska realizacija Turing-ove masine\n");
printf("Test primer [5/5]: deljivost zadatog broja sa 8\n");
printf("Mentor: dr Branko Malesevic, vanredni profesor\n");
printf("Student: Nemanja Karapetrovic [broj indeksa: ****/****]\n");
printf("Napomena: Potrebno je da se ulazna datoteka zove ulaz5.txt i da se nalazi
gde i izvorni fajl Turing_machine_5/5.c .\n");
printf("Vreme: %s", ctime(&vreme));
puts("-----");
-----");
puts("-----");
-----");
puts("");
}

```

```

Turing* učitavanje(Turing *lst) {
    FILE *p;
    Turing *novi, *posl;
    char pod1;
    p = fopen("ulaz5.txt", "r");
    if (p == NULL) {
        printf("Otvaranje datoteke nije uspelo!\n");
        exit(1);
    }
    while (fscanf(p, "%c", &pod1) != EOF) {
        novi = malloc(sizeof(Turing));
        if (novi == NULL) {
            printf("Alokacija memorije nije uspela!\n");
            exit(1);
        }
        novi->pod = pod1;
        novi->sled = NULL;
        if (lst == NULL) {
            lst = novi;
        }
        else {
            posl->sled = novi;
            novi->pret = posl;
        }
        posl = novi;
    }
    fclose(p);
}

```



```

        return lst;
    }

    Turing* obrada(Turing *lst, int *deljiv) {
        Turing *glava;
        glava = lst;
        bool iskoci = false;
        int q1 = 0, q2 = 0, q3 = 0, q4 = 0; /* Indikator: qi=0, program je bio u q1 stanju
nije bio u q1 stanju */
                                                    qi=0, program

                                                    /* Stranje q0

=> f(q0,b)=(q-,b,+1);

        f(q0,0)=(q1,0,+1);

        f(q0,1)=(q1,1,+1); */

        while (glava->pod == ' ') {
            glava = glava->sled; // Ova petlja ce pozicionirati pokazivac glava na
krajnji levi znak 1.
        }

        /* Stranje q0 => f(q0,b)=(q-,b,+1);
        f(q0,0)=(q1,0,+1);
        f(q0,1)=(q1,1,+1); */

        if (glava->pod == ' ') {
            printf("ERROR!!!\n"); // q- stanje
            if (glava->pod == ' ') glava->pod = ' ';
            glava = glava->sled;
            exit(1); // exit jer ako pocetak nije dobar, nema sta da se radi.
        }
        else { /* f(q0,0)=(q1,0,+1);
                f(q0,1)=(q1,1,+1); */

            if (glava->pod == '0') {
                glava->pod = '0';
                glava = glava->sled;
            }
            else {
                glava->pod = '1';
                glava = glava->sled;
            }
        }
    }

```

```

/* Stranje q1 => f(q1,b)=(q2,b,-1);
f(q1,0)=(q1,0,+1);
f(q1,1)=(q1,1,+1); */

// f(q1,b)=(q2,b,-1);

while ((q1 != 1 || q2 != 1 || q3 != 1 || q4 != 1) && !iskoci) { // Argument deljiv je
prosledjen kao false.
    if (q1 == 0 && q2 == 0 && q3 == 0 && q4 == 0) {
        if (q1 == 0) {
            if (glava->pod == ' ') {
                glava->pod = ' ';
                glava = glava->pret;
                q1 = 1;
            }
            else { /* f(q1,0)=(q1,0,+1)
                    f(q1,1)=(q1,1,+1) */
                if (glava->pod == '0') {
                    glava->pod = '0';
                    glava = glava->sled;
                }
                else {
                    glava->pod = '1';
                    glava = glava->sled;
                }
            }
        }
    }

    /* Stranje q2 => f(q2,b)=(q-,b,+1);
    f(q2,0)=(q3,0,-1);
    f(q2,1)=(q-,1,+1); */

    /* f(q2,b)=(q-,b,+1);
    f(q2,1)=(q-,1,+1); */
    if (q1 == 1 && q2 == 0 && q3 == 0 && q4 == 0) {
        if (glava->pod == '' || glava->pod == '1') {
            //q- stanje
            if (glava->pod == ' ') glava->pod = ' ';
            if (glava->pod == '1') glava->pod = '1';
            glava = glava->sled;
            iskoci = true;
        }
        else { //f(q2,0)=(q3,0,-1);
            glava->pod = '0';
            glava = glava->pret;

```

```

        q2 = 1;
    }

}
/* Stranje q3 => f(q3,b)=(q-,b,+1);
f(q3,0)=(q4,0,-1);
f(q3,1)=(q-,1,+1); */

/* f(q3,b)=(q-,b,+1);
f(q3,1)=(q-,1,+1); */

if (q1 == 1 && q2 == 1 && q3 == 0 && q4 == 0) {
    if (glava->pod == '' || glava->pod == '1') {
        //q- stanje
        if (glava->pod == ' ') glava->pod = ' ';
        if (glava->pod == '1') glava->pod = '1';
        glava = glava->sled;
        iskoci = true;
    }
    else { //f(q3,0)=(q4,0,-1);
        glava->pod = '0';
        glava = glava->pret;
        q3 = 1;
    }
}

/* Stranje q4 => f(q4,b)=(q-,b,+1);
f(q4,0)=(q+,0,+1);
f(q4,1)=(q-,1,+1); */

/* f(q4,b)=(q-,b,+1);
f(q4,1)=(q-,1,+1); */

if (q1 == 1 && q2 == 1 && q3 == 1 && q4 == 0) {
    if (glava->pod == '' || glava->pod == '1') {
        //q- stanje
        if (glava->pod == ' ') glava->pod = ' ';
        if (glava->pod == '1') glava->pod = '1';
        glava = glava->sled;
        iskoci = true;
    }
    else { //f(q4,0)=(q+,0,+1);
        glava->pod = '0';

```

```

        glava = glava->sled;
        q4 = 1;

    }
}

if (q4 == 1 && !iskoci) {
    *deljiv = true;
}

}

}
return lst;
}
void ispis(Turing *lst, bool deljiv) {
    Turing *tek;
    FILE *p;
    char izbor = 'a', izbor1;
    for (tek = lst; tek->pod != ' '; tek = tek->sled); // Petlja pozicionira pokazivac tek
na mesto prvog simbola koji nije blanko znak.

```

```

printf("Program:\n");

```

```

printf("f(q0,b)=(q-,b,+1)\n");
printf("f(q0,0)=(q1,0,+1)\n");
printf("f(q0,1)=(q1,1,+1)\n");

```

```

printf("f(q1,b)=(q2,b,-1)\n");
printf("f(q1,0)=(q1,0,+1)\n");
printf("f(q1,1)=(q1,1,+1)\n");

```

```

printf("f(q2,b)=(q-,b,+1)\n");
printf("f(q2,0)=(q3,0,-1)\n");
printf("f(q2,1)=(q-,1,+1)\n");

```

```

printf("f(q3,b)=(q-,b,+1)\n");
printf("f(q3,0)=(q4,0,-1)\n");
printf("f(q3,1)=(q-,1,+1)\n");

```

```

printf("f(q4,b)=(q-,b,+1)\n");
printf("f(q4,0)=(q+,0,+1)\n");
printf("f(q4,1)=(q-,1,-1)\n");

```

```

puts(" ");

```

```

printf("Da li zelite da se kreira posebna datoteka u kojoj ce se nalaziti rezultat
obrade ulazne datoteke? \n");
printf("Molim Vas unesite izbor: 1-DA 0-NE\n");
printf("Vas izbor: ");
scanf("%c", &izbor);
izbor1 = izbor;
// Zastita od nekorektnog unosa.
scanf("%c", &izbor);
if (izbor != '\n') {
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123 ako se to prvo
unese.
    }
}
while (izbor1 != '1' && izbor1 != '0') {
    printf("Nekorektan unos!!!\n");
    printf("Molim Vas ponovo unesite svoj izbor: 1-DA 0-NE\n");
    while (izbor != '\n')scanf("%c", &izbor);
    printf("Vas izbor: ");
    scanf("%c", &izbor);
    izbor1 = izbor;
    scanf("%c", &izbor);
    while (izbor != '\n') {
        scanf("%c", &izbor);
        izbor1 = 'a'; // Da trazi ponovni unos za npr. 123.
    }
}
puts(" ");

printf("Rezultat: ");
if (izbor1 == '1') {
    p = fopen("izlaz5.txt", "w");
    if (p == NULL) {
        printf("Neuspesno kreiranje datoteke!!!\n");
        exit(1);
    }
    if (deljiv) {
        fprintf(p, "%s", "Zadati broj je deljiv sa 8.\n");
        fclose(p);
    }
    else {
        fprintf(p, "%s", "Zadati broj nije deljiv sa 8.\n");
        fclose(p);
    }
}

```

```

    }

    if (deljiv) {
        printf("Zadati broj je deljiv sa 8.\n");
    }
    else {
        printf("Zadati broj nije deljiv sa 8.\n");
    }

    if (izbor1 == '1') printf("Vasa datoteka je uspesno formirana. Njen naziv je
    izlaz5.txt. Nalazi se na adresi izvornog fajla Turing_machine_5/5.c .\n");

    puts(" ");
    puts(" ");
}

Turing* obrisi(Turing *lst) {
    Turing *stari, *tek;
    tek = lst;
    while (tek != NULL) {
        stari = tek;
        tek = tek->sled;
        free(stari);
    }
    lst = NULL;
    return lst;
}

int main() {
    Turing *lst;
    lst = NULL;
    bool deljiv = false;
    time_t pocetak, kraj;
    time(&pocetak);
    podaci();
    lst = učitavanje(lst);
    lst = obrada(lst, &deljiv);
    ispis(lst, deljiv);
    lst = obrisi(lst);
    time(&kraj);
    printf("Vreme rada programskog dela obrade do ovog trenutka je %.2lf
    sekund(i,a,e).\n", difftime(kraj, pocetak));
    printf("Hvala sto ste koristili ovaj program!\n");
}

```

```
        puts("-----");
-----");
        puts("-----");
-----");
        system("pause");
        return 0;
    }
```

Literatura

https://sr.wikipedia.org/sr-el/Алан_Тјуринг

https://www.google.rs/search?q=alan+turing&rlz=1C1GCEA_enRS755RS755&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjastvdoIPYAhWIo6QKHTzKAswQ_AUICigB&biw=1370&bih=593#imgsrc=Rq1EtQw1gGaPSM

Branko Malešević, Ivana Jovović, *Složenost algoritama i odabrane metode optimizacije za student elektrotehnike*, Akademska misao, Beograd, 2015.