

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Testiranje softvera (SI3TS)  
*Nastavnik:* doc. dr Dragan Bojić  
*Asistent:* dipl. ing. Dražen Drašković  
*Ispitni rok:* Septembar 2013.  
*Datum:* 01.09.2013.

*Kandidat* \* : \_\_\_\_\_

*Broj indeksa* \* : \_\_\_\_\_

*Ispit traje 3 sata, prvih sat vremena nije dozvoljeno napuštanje ispita.  
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /6	<i>Zadatak 5</i>	_____ /8
<i>Zadatak 2</i>	_____ /6	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 3</i>	_____ /9	<i>Zadatak 7</i>	_____ /10
<i>Zadatak 4</i>	_____ /11		

*Ukupno na ispitu:* \_\_\_\_\_ /60      *Ukupno na domaćem* \* : \_\_\_\_\_ /40

*Rok u kome je odbranjen domaći* \* : \_\_\_\_\_ (primer: januar 2013)

**Ukupno:** \_\_\_\_\_ /100

**Ocena:** \_\_\_\_\_ ( \_\_\_\_\_ )

**Napomena:** Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**. \* popunjava student.

1. [6]

- a) Koliko različitih totalno uređenih SYN sekvenci postoji za sledeći segment programa?  
Napisati sve te sekvence. Početne vrednosti semafora su 0.

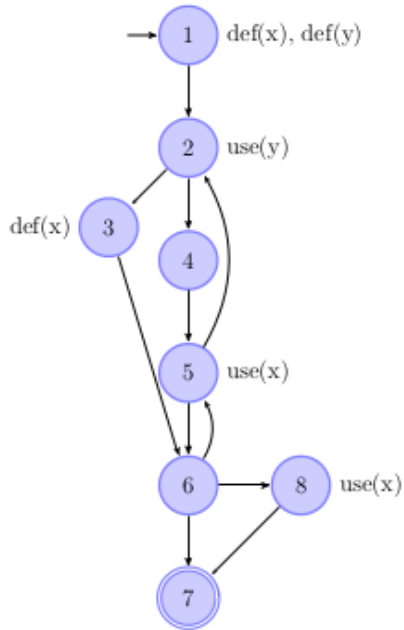
Thread 1	Thread 2	Thread 3
s2. V()	s2.P()	s2.V()
s1. P()	s2. P()	s1. V()
s1. P()	s1. V()	
s1. P()	s1. V()	

- b) Da li i koja od ovih sekvenci dovodi do deadlocka? Obrazložiti.
- c) Kako se modifikuje P() operacija semafora da bi se podržale SYN sekvence?

2. [6] Pretpostavimo da je u specifikaciji zahteva nastalo 10 grešaka i da će svaka takva greška indukovati dve u dizajnu. Dodatno je ubačeno 20 dizajnerskih grešaka. Greška u dizajnu uzrokuje 1.5 grešku u kodu, pored toga u fazi kodiranja napravljeno je dodatnih 30 grešaka. Pod pretpostavkom da jedinično testiranje otkriva 30% svih grešaka, integraciono 30% preostalih grešaka, a validaciono 50% preostalih grešaka. Inspekcije (review) nisu pravljene. Koliko grešaka će biti prisutno u softveru koji ide krajnjem korisniku?

3. [9] Dat je sledeći graf toka sa naznakama gde se promenljive x i y definišu i koriste.

a) Razmata se sledeći skup test putanja: [1, 2, 3, 6, 8, 7], [1, 2, 4, 5, 6, 7], [1, 2, 4, 5, 2, 3, 6, 5, 6, 7]. Koje kriterijume ovaj skup zadovoljava: pokrivanje svih putanja PAT, pokrivanje bazičnih putanja BAS, pokrivanje svih iskaza STA, pokrivanje svih odluka DEC, pokrivanje svih du parova DU, pokrivanje svih upotreba USES, pokrivanje svih definicija DEFS. Za svaki određni odgovor navesti šta nije pokriveno.



4. [11] Korisnici kasko osiguranja automobila u zavisnosti od godina starosti korisnika - vozača i broja prekršaja koji je taj vozač načinio u prethodnoj godini, dobijaju određene povoljnije tarife plaćanja ili osiguravajuća kuća može da prekine osiguranje tog automobila. Ako korisnik ima 25 i manje godina i nema ni jedan ili ima samo jedan prekršaj u prethodnoj godini ili ima između 26 i 45 godina i nema ni jedan prekršaj u prethodnoj godini, osiguranje u tekućoj godini je 30 evra. Ako korisnik ima 25 i manje godina i ima 2-4 prekršaja, ili ima između 26 i 45 godina i ima jedan prekršaj, takvi korisnici plaćaju 50 evra u tekućoj godini. Korisnici između 26 i 45 godina sa 2-4 prekršaja u prethodnoj godini plaćaju 70 evra osiguranje. Korisnik sa 46 i više godina, koji uz to nije napravio ni jedan prekršaj plaća 50 evra, ako je napravio jedan prekršaj 70 evra, a ako je napravio između 2-4 prekršaja plaća 100 evra. Ako korisnik ima 5 i više prekršaja u prethodnoj godini, bez obzira na starost vozača, korisnika usluge osiguranja, osiguranje raskida ugovor u tekućoj godini sa tim korisnikom. Svi korisnici koji imaju 2-4 prekršaja u prethodnoj godini dobijaju od osiguranja i pismo upozorenja. Nacrtati uzročno-posledični graf u zavisnosti od godina starosti vozača i broja prekršaja u prethodnoj godini, prikazati sva ograničenja među uzrocima i kreirati test primere za opisani sistem.

5. [8] Neka je u programskom jeziku Java data sledeća metoda `convert` (`String broj`) koja konvertuje broj sa ulaza u binarni broj:

```
public String convert(String broj){
    int num=-1;
    result = "Uneti broj je van dozvoljenog opsega!";
    try{
        num=Integer.parseInt(broj);
    }
    catch (Exception e){}

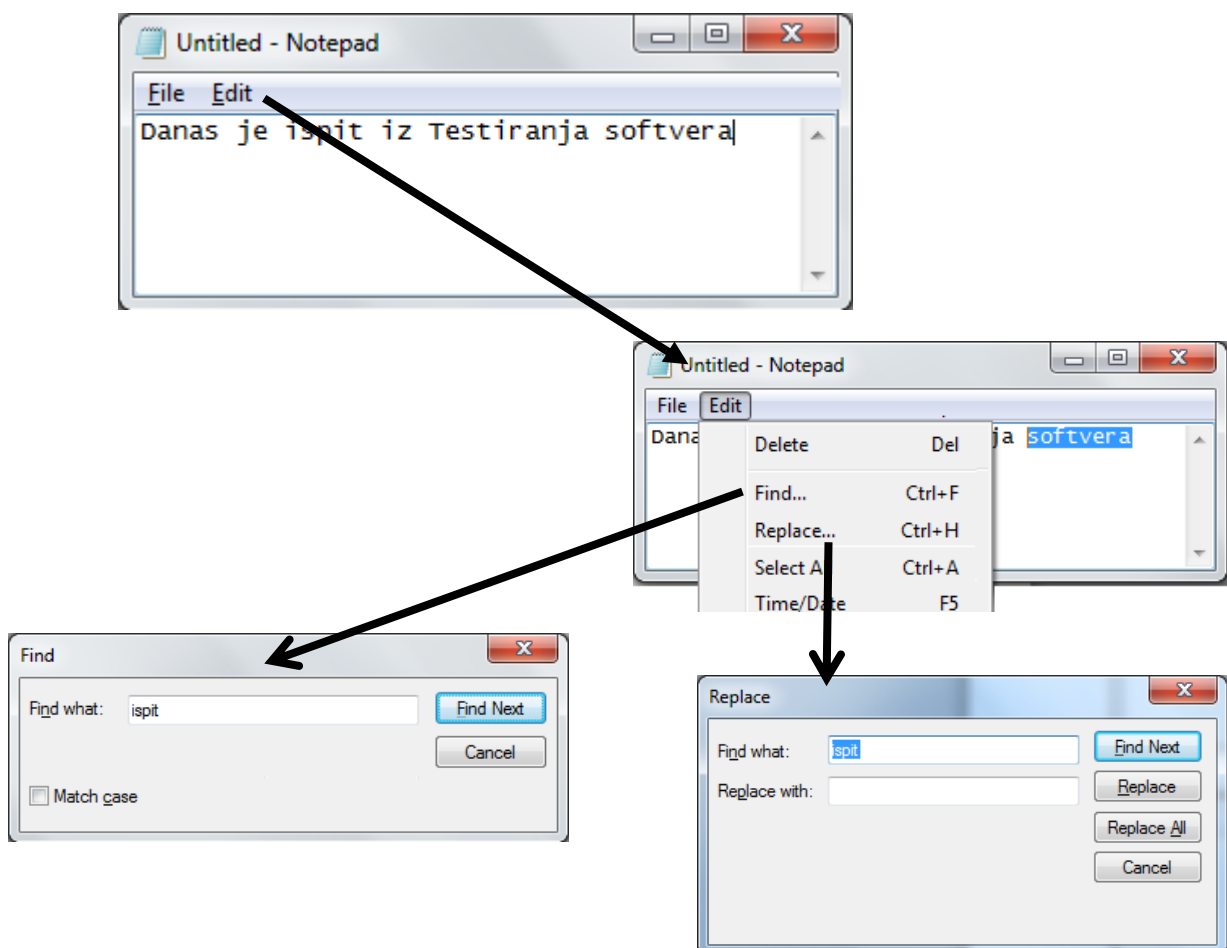
    if (num>=0 && num<=2147483647){
        result="";
        int exp = 0;
        int temp = num;
        while (temp > 0) {
            exp++;
            temp/=2;
        }
        exp -= 1;

        while (exp >= 0) {
            if (num >= Math.pow(2, exp)) {
                result += "1";
                num -= Math.pow(2, exp);
            }
            else result += "0";
            exp-=1;
        }
    }
    if (result=="") result="0";
    return result;
}
```

Testirati sve petlje u ovoj funkciji i detaljno obrazložiti koliko se puta može koja petlja izvršiti. Za svako izvršavanje petlje, napisati koliku vrednost treba imati ulazna promenljiva broj.

6. [10] *Notepad Light Edition* je *Windows* aplikacija za obradu teksta. Stavke i postavke menija date su na slici. U meniju postoje *File* i *Edit*. Stavka *Edit* omogućava rad sa tekstem. Opcijom brisanje (*Delete*) može se obrisati tekst, ukoliko je selektovan. Takođe, nakon selektovanja određenog dela teksta, moguće su opcije *Find* i *Replace*, koje služe za pretraživanje ili zamenu određenog pojma koji se traži. Ukoliko ništa nije selektovano nije moguće vršiti pretragu/zamenu. U podmeniju *Edit* postoje još opcija *Select All* (za selektovanje celog teksta ukoliko postoji nešto unešeno) i opcija *Time/Date* (za dodavanje trenutnog vremena i datuma u tekst na kraj dokumenta).

Konstruisati kompletan FSM (*Finite State Machine*) graf za ovu aplikaciju i za podgraf “*Save As*” napisati interaktivne parove, neispravne interaktivne parove i neispravne sekvence interakcije.



7. [10] Neka je dat sledeći program za izračunavanje prostih brojeva.

- a) Odrediti sve LCSAJ za dati program (navesti početno i krajnje mesto sekvence i mesto skoka)
- b) Odrediti minimalan skup testova koji pokrivaju sve linije koda u programu i maksimalan broj LCSAJ sekvenci, koje se tom prilikom izvršavaju.

```
1      READ (Num);
2      WHILE NOT End of File DO
3          Prime := TRUE;
4          FOR Factor := 2 TO Num DIV 2 DO
5              IF Num - (Num DIV Factor)*Factor = 0 THEN
6                  WRITE (Factor, ` is a factor of', Num);
7                  Prime := FALSE;
8              ENDIF;
9          ENDFOR;
10         IF Prime = TRUE THEN
11             WRITE (Num, ` is prime');
12         ENDIF;
13         READ (Num);
14     ENDWHILE;
15     WRITE (`End of prime number program');
```