

1. Ukoliko se koristi statičko pravilo djelokruga, na aktivacijski zapis koje procedure pokazuje kazaljka nelokalnih imena potprograma C.

```
Glavni(){  
    def A(){  
        B();  
    }  
    def B(){  
        C();  
    }  
    def C(){  
        print("C");  
    }  
    A();  
}
```

- a. A
b. **Glavni**
c. B
d. Ništa od navedenog
2. Označite produkciju koja zadovoljava uvjete L-atributne prijevodne gramatike u pojednostavljenom obliku pravila računanja vrijednosti svojstava.

- a. $\langle X \rangle_{h,i} \rightarrow a_j \langle Y \rangle_k \langle Z \rangle_{q,p} \{f\}_{L,m,n,o}$
 $L \leftarrow h, m \leftarrow j, n \leftarrow k, q \leftarrow o, i \leftarrow p$
b. $\langle X \rangle_i \rightarrow a_j \langle Y \rangle_k \{f\}_{L,m,n,o} \langle Z \rangle_{q,p}$
 $L \leftarrow k, m \leftarrow j, n \leftarrow k, q \leftarrow o, i \leftarrow p$
c. $\langle X \rangle_{h,i} \rightarrow a_j \langle Y \rangle_k \{f\}_{m,n,o} \langle Z \rangle_{q,p}$
 $k \leftarrow h, m \leftarrow j, n \leftarrow k, q \leftarrow o, i \leftarrow p$
d. $\langle X \rangle_{h,i} \rightarrow a_j \langle Y \rangle_k \{f\}_{L,m,n,o} \langle Z \rangle_{q,p}$
 $L \leftarrow h, m \leftarrow j, n \leftarrow k, q \leftarrow o, i \leftarrow p$
e. $\langle X \rangle_{h,i} \rightarrow a_j \{f\}_{L,m,n,o} \langle Y \rangle_k \langle Z \rangle_{q,p}$
 $L \leftarrow h, m \leftarrow j, n \leftarrow k, q \leftarrow o, i \leftarrow p$

3. Generator ciljnog programa kao izlaz može imati različite ciljne jezike. Za premjestivi ciljni program vrijedi:
- a. Program povezič izvodi postupak dorade adrese.
b. Adrese naredbi su u potpunosti izrađene.
c. Program punitelj povezuje zasebno prevedene procedure.
d. Adrese podataka su u potpunosti izrađene.
e. **Ništa od navedenog.**

4. Za zadani isječak programa koji broj mrežica će se odrediti primjenom algoritma bojanja grafova?

```
i = 1;  
j = i + 1;  
k = j + 1;  
l = i + j + k;  
i = 0;
```

- a. **5**
b. 3
c. 4
d. 10

e. 6

5. Budući da se zahtjeva da preuređeni međukôd tijekom sinteze ciljnog programa sačuva svoje izvorno značenje, prije pretvorbe potrebno je analizirati izvođenje programa.

Analizu izvođenja programa čini poredani slijed analiza:

- a. Analiza pseudonima, analiza tijeka izvođenja programa, analiza toka podataka, analiza zavisnosti podataka, postupci pretvorbe
 - b. Analiza tijeka izvođenja programa, analiza pseudonima, analiza toka podataka, analiza zavisnosti podataka, postupci pretvorbe
 - c. Analiza toka podataka, analiza pseudonima, analiza zavisnosti podataka, analiza tijeka izvođenja programa, postupci pretvorbe
 - d. Analiza tijeka izvođenja programa, analiza toka podataka, analiza zavisnosti podataka, analiza pseudonima, postupci pretvorbe**
 - e. Analiza toka podataka, analiza tijeka izvođenja programa, analiza pseudonima, analiza zavisnosti podataka, postupci pretvorbe
6. Promatramo potisni automat parsera od vrha prema dnu za zadanu L -atributnu prijevodnu gramatiku. Što se od navedenog ne nalazi na stogu tog automata na početku njegova rada?
- a. Početni nezavršni znak gramatike
 - b. Kazaljke koje pokazuju na mjesto zapisa vrijednosti izvedenih svojstava početnog nezavršnog znaka
 - c. Kazaljke koje pokazuju na mjesto zapisa vrijednosti nasljednih svojstava početnog nezavršnog znaka**
 - d. Oznaka dna stoga
 - e. Početne vrijednosti nasljednih svojstava početnog nezavršnog znaka
7. Parsiramo L -atributnu prijevodnu gramatiku metodom rekurzivnog spusta. Označite produkciju, ako takva postoji, kojoj je pridružen sljedeći potprogram.

```
S ( ) {  
    Slučaj (Ulaz.znak) {  
        'while': {  
            Lokalne varijable x, y;  
            Ulaz.znak = sljedeći znak niza w;  
            ako (Ulaz.znak != '(')  
                Odbaci();  
            A(&x);  
            Ulaz.vrijednost = vrijednost sljedećeg znaka niza w;  
            ako (Ulaz.znak != ')')  
                Odbaci();  
            y = 1 - x;  
            B(y);  
        }  
        Svi ostali znakovi:  
            Odbaci();  
    }  
}
```

a. $S \rightarrow while(A_y)B_x$
 $y \leftarrow 1 - x$

- b. $S_x \rightarrow while(A_x)B_y$
 $y \leftarrow 1 - x$
- c. $S \rightarrow while(A_x)B_y$
 $y \leftarrow 1 - x$
- d. Ništa od navedenog
- e. $S \rightarrow while_x(A_y)B_x$
 $y \leftarrow 1 - x$

8. Koji od navedenih nije jezični procesor s obzirom na stupanj pripremljenosti ciljnog programa za izvođenje?

- a. Ništa od navedenog
- b. Generator izvodivog ciljnog programa
- c. Generator zasebnih dijelova programa
- d. Spremi-i-pokreni jezični procesor
- e. Generator premjestivog ciljnog programa

9. Obilježja se dodjeljuju različitim dijelovima izvornog programa, odnosno različitim leksičkim i sintaksnim cjelinama.

Naredba pridruživanja definira se, između ostalog, sljedećom produkcijom gramatike:

$$\langle S \rangle \rightarrow IDN := \langle E \rangle$$

Obilježjima naredbi pridružuju se dvije vrijednosti: Bez Pogreške i Pogreška.

Produkcijama gramatike dodaju se sljedeće semantičke akcije:

$$\langle S \rangle_{V1} \rightarrow IDN_{V2} = \langle E \rangle_{V3}$$

```
{
  ako (1)
    2
  inače
    3
}
```

Napomena! Potrebno je odabrati ispravne vrijednosti za 1, 2 i 3.

- a. 1. $V2 == V3$
 2. $V2 = \text{BezPogreške}$
 3. $V3 = \text{Pogreška}$
- b. 1. $V2 != V3$
 2. $V3 = \text{BezPogreške}$
 3. $V3 = \text{Pogreška}$
- c. 1. $V2 == V3$
 2. $V2 = \text{Pogreška}$
 3. $V2 = \text{BezPogreške}$
- d. 1. $V2 != V3$
 2. $V1 = \text{Pogreška}$
 3. $V1 = \text{BezPogreške}$
- e. 1. $V2 == V3$
 2. $V1 = \text{Pogreška}$
 3. $V1 = \text{BezPogreške}$

10. Ako su leksičke jedinice opisane pravilima $r1$, $r2$ i $r3$, koje će jedinice prepoznati analizator prilikom analize niza *jedan1dva*?

r1: jedan/1

r2: dva/2

r3: (a|b|...|z)*(0|1|...9)*

- a. r1, r2
- b. r1, r3, r3**
- c. r1, r3, r2
- d. r1, r1, r2
- e. r1, r3

11. Neka je zadan sljedeći program:

- ```
1) i = j + 1;
2) ako (i > 5)
3) skoči L1
 inače
 {
4) n = j * k;
5) m = n + 1;
 }
6) L1: n = m / 2;
```

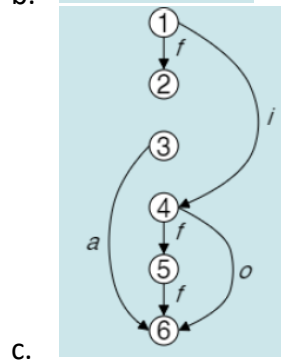
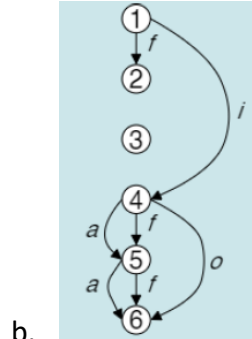
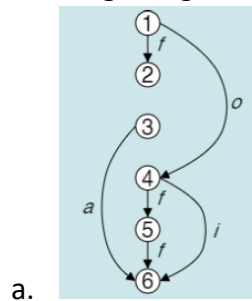
**Unaprijedno zavisne (f)** – prije je definirana i koristi se kasnije

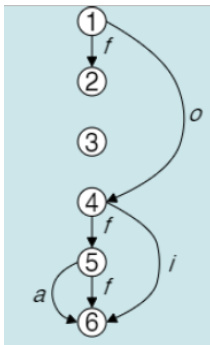
**Unazadno zavisne (a)** – sljedeća naredba mijenja podatak koji naredba prije koristi

**Zavisne u odnosu na odredište (o)** – s lijeve strane se treba pojaviti isto slovo

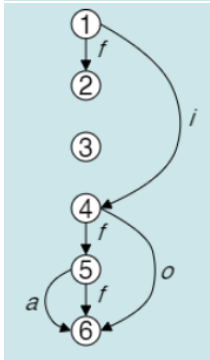
**Zavisne u odnosu na ishodište (i)** – s desne strane se pojavljuje isto slovo

Kako izgleda graf zavisnosti podataka za zadani program?





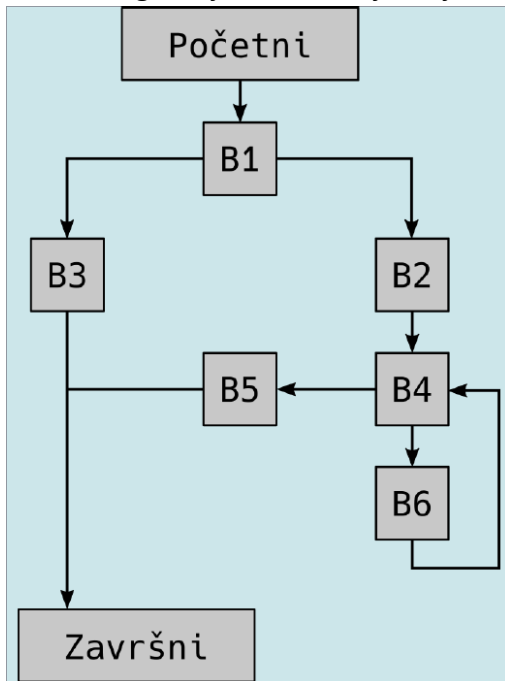
d.



e.

✓

12. Za zadani graf tijeka izvođenja, koja od ponuđenih tvrdnji dominacije je istinita?



- a.  $dom(Početni) = \{\}$
- b.  $dom(B4) = \{Početni, B1, B2, B4\}$
- c.  $dom(B2) = \{B1, B2\}$
- d.  $dom(B3) = \{Početni, B1\}$
- e.  $dom(B5) = \{Početni, B1, B3, B4, B5\}$

13. Što će ispisati sljedeći program ako se parametri razmjenjuju primjenom mehanizma razmjene adresa?

```
int x = 0;
int y = 1;
void f(a, b) {
 x = 7;
 b = 2;
```

```

 y = a + b;
 print(a, b, x, y);
 }
 f(x, y);
 print(x, y);

```

- a. 010101
- b. 797979**
- c. 727979
- d. 727901
- e. 727901

14. Parsiramo  $L$ -atributnu prijevodnu gramatiku potisnim automatom. Neka je zadana sljedeća produkcija

$$\langle S \rangle_{n1, i1} \rightarrow \{f\}_{n2, i2} a_x \langle A \rangle_{i3} b_y \{h\}_{n3, n4, i4} \langle B \rangle_{n5, n6, n7}$$

$$n_2 \leftarrow n_1, n_3 \leftarrow i_2, n_4 \leftarrow x, (i_1, n_5) \leftarrow i_4, n_6 \leftarrow i_3, n_7 \leftarrow y$$

gdje je  $\langle S \rangle$  početni nezavršni znak gramatike. Neka su elementi stoga, počevši od vrha prema dnu, indeksirani s brojevima 0, 1, 2... . Pod pretpostavkom da se na stogu nalazi početna konfiguracija potisnog automata, što će biti zapisano unutar stoga na indeksu broj 9 nakon jednog koraka rada potisnog automata u kojem pročitamo ulazni završni znak  $a$  i primijenimo zadanu produkciju?

- a.  $y$ . vrijednost
- b.  $i_4$ . vrijednost**
- c.  $\langle B \rangle$
- d.  $x$ . vrijednost
- e.  $i_3$ . vrijednost

15. Koliko čvorova ima graf tijeka izvođenja idućeg programa?

```

input(x)
x = x % 10
if x > 5:
 print("Yes")
 goto END
else:
 print("No")
END:
x = 0

```

- a. 2
- b. 1
- c. 6**
- d. 8

16. Koje troadresne naredbe generira sintaksnom upravljanje generiranje međukoda za sljedeći izraz:

$$a = b * (c + d) - e$$

Oznaka  $p^*$  se koristi za pomoćna imena.

- a.  $p1 = b * c;$   
 $p2 = p1 + d;$   
 $a = p2 - e;$
- b.  $p1 = c + d;$   
 $p2 = b * p1;$   
 $a = p2 - e;$**

- c.  $p1 = c + d;$   
 $p2 = p1 - e;$   
 $a = b * p2;$
- d.  $p1 = b * p1;$   
 $a = p1 - e;$
- e.  $p1 = c + d;$   
 $a = b * p1 - e;$

17. Zadana je gramatika s produkcijama:

$S \rightarrow aABC; S \rightarrow bBc; A \rightarrow \varepsilon; A \rightarrow cB; B \rightarrow bB; B \rightarrow a$

Čemu je jednak skup  $PRIMIJEI(A \rightarrow \varepsilon)$  ?

- a.  $\{a, b, c\}$
- b.  $\{a, c\}$
- c.  $\{\varepsilon\}$
- d.  $\{c\}$
- e.  $\{a, b\}$

18. U metodi rekurzivnog spusta za  $L$ -atributnu prijevodnu gramatiku, znakovima gramatike pridružuju se \_\_\_\_\_, a svojstvima znakova pridružuju se \_\_\_\_\_.

- a. **Potprogrami, programske varijable**
- b. Programske varijable, potprogrami
- c. Znakovi stoga, pravila preslikavanja
- d. Pravila preslikavanja, znakovi stoga
- e. Znakovi stoga, programske varijable

19. Dinamička provjera vrijednosti obilježja izvodi se tijekom:

- a. Leksičke analize
- b. Sintaksne analize
- c. Semantičke analize
- d. Generiranje međukoda
- e. **Izvođenje ciljnog programa**

20. Listu unaprijednih adresa i listu unazadnih adresa koristi:

- a. **Generator ciljnog programa**
- b. Semantički analizator
- c. Ciljni program
- d. Sintaksni analizator
- e. Potisni automat

21. U metodi rekurzivnog spusta za  $L$ -atributnu prijevodnu gramatiku, ako je varijabli pridruženo nasljedno svojstvo, razmjenjuje se njezina \_\_\_\_\_, a ako je varijabli pridruženo izvedeno svojstvo, razmjenjuje se njezina \_\_\_\_\_.

- a. Adresa, vrijednost
- b. **Vrijednost, adresa**
- c. Adresa, adresa
- d. Vrijednost, vrijednost
- e. Ništa od navedenog (nema razmjene varijabli)

22. Što od navedenog nije jedan od osnovnih mehanizama razmjene ulazno/izlaznih parametara procedura:

- a. Razmjena vrijednosti
- b. Povratna razmjena vrijednosti

- c. Razmjena adrese
  - d. Povratna razmjena adrese**
  - e. Razmjena imena
23. Prilikom generiranja ciljnog programa na temelju postfiksno sustava oznaka, izravnjanje sintaksnog stabla ostvaruje se primjenom:
- a. LR-parsera
  - b. LL(1)-parsera
  - c. Potisnog automata
  - d. Potisnog stoga**
  - e. Stabla dominacije
24. Analizu pseudonima čine dva dijela:
- a. Pretraživač i poveznik pseudonima
  - b. Pretraživač i konstruktor pseudonima
  - c. Skupljač i prenositelj pseudonima**
  - d. Skupljač i prevoditelj pseudonima
  - e. Konstruktor i destruktork pseudonima
25. Pri traženju slobodnog segmenta memorijskog prostora, pretpostavimo da nijedan slobodni segment pojedinačno nije dovoljno velik za traženu memoriju. Koji se od navedenih postupaka u tom slučaju najprije pokušava provesti?
- a. Pakiranje slobodnih segmenata
  - b. Pakiranje zauzetih segmenata
  - c. Združivanje susjednih zauzetih segmenata
  - d. Združivanje susjednih slobodnih segmenata**
  - e. Združivanje svih slobodnih segmenata
26. Dio opisnika procedure po kojem se razlikuju statičko i dinamičko pravilo djelokruga jest:
- a. Statička memorija
  - b. Upravljačka kazaljka
  - c. Kazaljka stoga
  - d. Kazaljka nelokalnih imena**
  - e. Kazaljka sata
27. U analizi dominacije koju provodimo nad grafom tijeka izvođenja programa  $dom(x)$  označava skup dominatora čvora  $x$ . Čvor  $d$  je neposredni dominator čvora  $a$  ako i samo ako za svaki drugi čvor  $c$  t.d. ( $c \neq d, c \neq a$ ) vrijedi:
- a.  $c \in dom(a)$  ili  $d \in dom(c)$
  - b.  $c \in dom(a)$  ili  $c \in dom(d)$
  - c.  $c \in dom(a)$  ili  $c \in dom(d)$
  - d.  $c \notin dom(a)$  ili  $d \notin dom(c)$**
  - e.  $c \notin dom(a)$  i  $d \notin dom(c)$
28. Što od navedenog nije dio opisnika procedure:
- a. Lokalni podatci
  - b. Statička memorija**
  - c. Upravljačka kazaljka
  - d. Kazaljka nelokalnih imena
  - e. Vrijednost ulaznih parametara



29. Izvođenje naredbi za koje vrijedi da se vrijednost relacije \_\_\_\_\_ ne mijenja određuje životni vijek pridruživanja imena
- a. Stanja
  - b. Okoline**
  - c. Naslovljavanja
  - d. Odnosa
  - e. Zavisnosti
30. Strojno nezavisni program virtualnog stroja u strojni program prevodi:
- a. Jezični procesor
  - b. Jezični postprocesor**
  - c. Program povezič
  - d. Program punitelj
  - e. Jezični preprocesor
31. Tijekom analize strukture pri analizi tijeka izvođenja programa, analiza strukture obilazi graf tijeka izvođenja programa, traži podgrafove uzorke, zamijeni ih jednim zamjenskim čvorom i gradi:
- a. Usmjereni graf izvornog programa
  - b. Stablo čvorova
  - c. Dominantno stablo
  - d. Graf uzoraka
  - e. Upravljačko stablo**
32. Što od navedenog nije jedan od tipova jezičnih procesora u podjeli s obzirom na stupanj pripremljenosti ciljnog programa za izvođenje?
- a. Spremi-i-pokreni jezični procesor
  - b. Generatori izvodivog ciljnog programa
  - c. Generatori produkcija strojnog jezika**
  - d. Generatori premjestivog ciljnog programa
  - e. Generatori zasebnih dijelova programa
33. Gradimo atributnu prijevodnu gramatiku koja generira troadresne naredbe za računanje logičkih izraza. Produkciju  $E_{ime1, kod1} \rightarrow \neg E_{ime2, kod2}$  ima smisla proširiti sljedećim akcijskim znakovima:
- a. `Ime1=novoIme(); Kod1=Generiraj(Ime1 || '':=not'' Ime2)`
  - b. `Ime2=novoIme(); Kod2=Generiraj(kod1 || '':=not'' Ime2)`
  - c. `Ime1=novoIme(); Kod1=Generiraj(kod1 || Ime1 '':=not'' Ime2)`
  - d. `Ime1=novoIme(); Kod1=Generiraj(kod2 || Ime1 '':=not'' Ime2)`**
  - e. `Ime2=novoIme(); Kod2=Generiraj(Ime1 || '':=not'' Ime2)`
34. Program Glavni sadrži procedure A(x), B(y), C(z) od kojih nijedna ne poziva samu sebe. Ne pozivaju se ni međusobno, osim procedure A koja u nekim slučajevima poziva B i C. Koja je najveća moguća dubina stabla aktiviranja procedura za program Glavni uz pretpostavku da je dubina korijena 1?
- a. 2
  - b. 3**
  - c. 4
  - d. 5
  - e. Neograničena
35. Pri optimiranju procedura međukoda srednje razine koristi se pretvorba rekurzivnih procedura u:
- a. Petlje**
  - b. Zasebne pozive za svaki parametar

- c. Procedure s jednim prijenosnim parametrom
  - d. Rekurzivne procedure nije moguće preoblikovati
  - e. Skokove
36. Koraci postupka analize izvornog programa redom su:
- a. Leksička, semantička i sintaksna analiza
  - b. Sintaksna, semantička i leksička analiza
  - c. Leksička, sintaksna i semantička analiza**
  - d. Semantička, sintaksna i leksička analiza
  - e. Semantička, leksička i sintaksna analiza
37. Tijekom prevođenja izvornog programa pomoću Co-No tablice, trenutni operator „;“ i sljedeći operator „+“ određuju sljedeću akciju generatora ciljnog programa:
- a. Pomnoži
  - b. Dohvati**
  - c. Zbroji
  - d. Obriši
  - e. Spremi
38. Pri izgradnji potisnog automata za atributnu prijevodnu gramatiku, temeljem produkcije  $A \rightarrow \zeta b \Phi \alpha$ , pri čemu su  $\zeta$  i  $\Phi$  izlazni znakovi, gradi se akcija:
- a. Izlaz( $\zeta \Phi$ ); Zamijeni( $\alpha^r$ ); Pomakni;**
  - b. Izlaz( $\zeta$ ); Zamijeni( $\alpha^r \Phi$ ); Pomakni;
  - c. Izlaz( $\zeta \Phi$ ); Izvuci; Zadrži;
  - d. Zamijeni( $\alpha^r \Phi$ ); Pomakni;
  - e. Izlaz( $\zeta \Phi$ ); Zamijeni( $\alpha^r$ ); Zadrži;
39. Za sljedeću kontekstno neovisnu gramatiku odredite relaciju *ReduciranZnakom* za završni znak  $c$ .
- $$S \rightarrow aABc; \quad S \rightarrow cBAB; \quad A \rightarrow aA; \quad A \rightarrow \varepsilon; \quad B \rightarrow bB; \quad B \rightarrow c;$$
- a.  $(c, \perp)$
  - b.  $(c, a), (c, \perp)$
  - c.  $(c, a), (c, b), (c, c)$
  - d.  $(c, a), (c, b), (c, c), (c, \perp)$**
  - e.  $(c, b), (c, \varepsilon)$
40. U LL(1) gramatici, za praznu produkciju  $A \rightarrow a$ , relacija *PRIMIJENI* izračunava se kao:
- a.  $ZAPOČINJE(\alpha) \cup SLIJEDI(A)$**
  - b.  $ZAPOČINJE(A) \cup SLIJEDI(\alpha)$
  - c.  $ZAPOČINJE(A) \cup SLIJEDI(A)$
  - d.  $ZAPOČINJE(\alpha) \cup SLIJEDI(\alpha)$
  - e.  $ZAPOČINJE(\alpha) \cap SLIJEDI(A)$
41. Dva su osnovna načina traženja slobodnog segmenta memorijskog prostora: potraga za prvim slobodnim segmentom dovoljne veličine i potraga za slobodnim segmentom koji je po svojoj veličini \_\_\_\_\_ tražene memorije.
- a. Jednak
  - b. Veći, ali najbliži veličini**
  - c. Veći od
  - d. Manji, ali najbliži veličini
  - e. Manji od

42. Leksički analizator slijedno čita tekst izvornog programa:

- a. **Znak po znak**
- b. Riječ po riječ
- c. Liniju po liniju
- d. Blok po blok
- e. Funkciju po funkciju

43. Ako je LR stavka potpuna, onda je oznaka točke na \_\_\_\_\_ mjestu \_\_\_\_\_ strane produkcije.

- a. Krajnje lijevom, desne
- b. Krajnje desnom, lijeve
- c. **Krajnje desnom, desne**
- d. Proizvoljnom, lijeve
- e. Proizvoljnom, desne

44. Za sljedeću gramatiku odredite vrstu:  $S \rightarrow Bc$ ;  $S \rightarrow cBb$ ;  $B \rightarrow \varepsilon$ ;  $B \rightarrow cb$ ;

- a. **Regularna gramatika**
- b. Q-gramatika
- c. S-gramatika
- d. LL(1) gramatika
- e. Operatorska gramatika

45. Za sljedeću gramatiku odredite vrstu:  $S \rightarrow Bb$ ;  $S \rightarrow cBb$ ;  $B \rightarrow \varepsilon$ ;  $B \rightarrow ab$ ;

- a. Regularna gramatika
- b. Q-gramatika
- c. S-gramatika
- d. **LL(1) gramatika**
- e. Operatorska gramatika

46. Za Q-gramatiku s produkcijama  $S \rightarrow bAS$ ;  $S \rightarrow a$ ;  $A \rightarrow \varepsilon$ ; odredite skup  $PRIMIJEI(A \rightarrow \varepsilon)$ .

- a.  $\{a\}$
- b.  $\{b\}$
- c.  $\{\varepsilon\}$
- d.  **$\{a, b\}$**
- e.  $\{a, b, \varepsilon\}$

47. Koje upravljačke tablice koristi parser Pomakni-Reduciraj?

- a. Reduciraj
- b. Reduciraj, Odbaci
- c. Stavi, NovoStanje
- d. Pomakni, Odbaci
- e. **Stavi, Pomakni/Reduciraj**

48. Za ciljni program prikazan u donjem dijelu stranice dodjeljuju se registri temeljem Cockeovog algoritma, a za bojenje grafa koristi se Chaitinov algoritam. Koliko se stvarnih registara dodjeljuje ciljnom programu?

- 1)  $i := 11$
- 2)  $j := 5$
- 4)  $m := m - m$
- 5)  $k := i * 10$
- 6)  $n := i - j$
- 7)  $i := k + 12$

- a. 1
- b. 2
- c. 3**
- d. 4
- e. 5

Za programski odsječak u donjem dijelu stranice

```
var i = 1
var b[3] = {5, 6, 7}
procedura f(x)
 i = 0
 x = 8
 ispis(b[0], b[1], b[2])
 i = 2
 x = 9
kraj
f(b[i])
ispis(i, b[0], b[1], b[2])
```

odredite ispis ako se pri pozivu potprograma koristi:

49. razmjena vrijednosti:

- a. 5,6,7; 2,5,9,7
- b. 5,6,7; 2,5,6,7**
- c. 8,6,7; 1,8,6,9
- d. 5,6,7; 1,5,6,7
- e. 5,7,6; 2,5,9,7

50. povratna razmjena vrijednosti:

- a. 5,6,7; 2,5,6,7
- b. 8,6,7; 1,8,6,9
- c. 5,6,7; 2,5,9,7**
- d. 5,6,7; 1,5,9,7
- e. 5,8,7; 2,5,9,7

51. razmjena imena:

- a. 8,6,7; 1,8,6,9
- b. 5,6,7; 2,5,9,7
- c. 5,6,7; 1,5,6,7
- d. 8,6,7; 2,8,6,9**
- e. 5,8,7; 2,5,9,7

52. razmjena adresa:

- a. 5,8,7; 2,5,9,7**
- b. 5,6,7; 1,5,6,7
- c. 5,6,7; 2,5,9,7
- d. 8,6,7; 2,8,6,9
- e. 5,8,7; 1,5,9,7

Za program u donjem dijelu stranice

```

01 Glavni()
02 int a = 1;
03 X(x)
04 {
05 vrati x + 1;
06 }
07 Y(y)
08 {
09 vrati X(y * a);
10 }
11 Z(z)
12 int a = 2;
13 {
14 vrati Y(z * a);
15 }
16 {
17 X(5);
18 Y(5);
19 Z(5);
20 }

```

nacrtajte stablo aktiviranja procedura i za to stablo odredite:

53. Broj čvorova stabla:

- a. 3
- b. 4
- c. 6
- d. 7**
- e. 10

54. Parametar poziva procedure u čvoru najveće dubine stabla ako koristimo **statičko** pravilo djelokruga:

- a. 5
- b. 10**
- c. 11
- d. 20
- e. 21

55. Parametar poziva procedure u čvoru najveće dubine stabla ako koristimo **dinamičko** pravilo djelokruga:

- a. 5
- b. 10
- c. 11
- d. 20**
- e. 21

56. Generiranje ciljnog programa na temelju postfiksno sustava oznaka: Ako se u međukodu pročita operator, onda generator primijeni akciju:

- a. stavi pročitani znak međukoda na vrh stoga, a glavu za čitanje zadrži na trenutnom znaku
- b. stavi pročitani znak međukoda na vrh stoga i pomakni glavu za čitanje na sljedeći znak
- c. generiraj naredbe ciljnog programa i pomakni glavu za čitanje na sljedeći znak

d. **uzmi s vrha stoga zadani broj operanada, generiraj naredbe ciljnog programa i stavi rezultirajući operand na vrh stoga**

e. uzmi s vrha stoga zadani broj operanada, generiraj naredbe ciljnog programa

57. Zadana je produkcija L-atributne prijevodne gramatike:  $X_{n_1, i_2, i_3} \rightarrow Y_{n_4} Z_{i_5} V_{i_6, n_7, n_8} W_{n_9}$  gdje su  $i_2, i_3, i_5$  te  $i_6$  izvedena svojstva, a  $n_1, n_4, n_7, n_8$  i  $n_9$  su nasljedna svojstva. Nasljedno svojstvo  $n_7$  može se računati na temelju svojstava:

a.  $n_1, i_2, i_3$

**b.  $n_1, n_4, i_5$**

c.  $n_1, i_2, i_3, n_4, i_5$

d.  $n_1, n_4, n_8, n_9$

e.  $i_2, i_3$

58. Za sljedeću kontekstno neovisnu gramatiku  $S \rightarrow pAmC$ ;  $A \rightarrow dSa$ ;  $C \rightarrow dA$ ;  $S \rightarrow bA$ ;  $A \rightarrow e$ ; izračunajte vrijednost relacije *IspodZnaka* za završni znak *b*.

a. *IspodZnaka(b, d), IspodZnaka(b, m)*

b. *IspodZnaka(b, d)*

c. *IspodZnaka(b, a), IspodZnaka(b, e)*

d. *IspodZnaka(b, e)*

**e. *IspodZnaka(b, d), IspodZnaka(b, e)***

59. Ako je stanje *LR(1)* parsera označeno stawkama  $X \rightarrow \alpha \bullet \{a, b, c\}$  i  $X \rightarrow \alpha \bullet g\beta\{d, e, f\}$ , parser će učitavanjem znaka *b* iz ulaznog niza izvesti akciju:

a. *pomakni*

**b. *reduciraj***

c. *prihvati*

d. *odbaci*

e. neće moći jednoznačno odlučiti zbog proturječja

60. Tekst se odnosi na sljedeća dva zadatka. Za program naveden na dnu druge stranice

```
01 Glavni()
02 def X(a)
03 def Y(x)
04 {
05 vrati x*3;
06 }
07 def Z(x)
08 {
09 vrati x%2;
10 }
11 {
12 vrati a*a;
13 }
14 {
15 ispiši Y(4);
16 }
```

Navedite gdje pokazuje kazaljka nelokalnih imena procedure Y u trenutku izvođenja naredbe 05 ako se koristi:

(i) statičko pravilo djelokruga ugniježđenih procedura

- a. opisnik procedure Glavni
- b. opisnik procedure X**
- c. opisnik procedure Y
- d. opisnik procedure Z
- e. ništa od navedenog

61. (ii) dinamičko pravilo djelokruga ugniježđenih procedura

- a. opisnik procedure Glavni**
- b. opisnik procedure X
- c. opisnik procedure Y
- d. opisnik procedure Z
- e. ništa od navedenog

Za programski odsječak prikazan na dnu stranice

```
var x = 0
var A[2] = {10, 20}
procedura P(a)
 x = 1
 a = 100
 ispiši(A[0], A[1])
 x = 0
 a = 101
kraj
P(A[x])
Ispiši(x, A[0], A[1])
```

Odredite ispis ako se kod poziva potprograma koristi:

62. (i) razmjena vrijednosti

- a. 10,20; 0,10,20**
- b. 100,20; 0,101,20
- c. 10,100; 0,101,100
- d. 10,20; 0,101,20
- e. 10,20; 0,101,100

63. (ii) razmjena adresa

- a. 10,20; 0,10,20
- b. 100,20; 0,101,20**
- c. 10,100; 0,101,100
- d. 10,20; 0,101,20
- e. 10,20; 0,101,100

64. (iii) razmjena imena

- a. 10,20; 0,10,20
- b. 100,20; 0,101,20
- c. 10,100; 0,101,100**
- d. 10,20; 0,101,20
- e. 10,20; 0,101,100

65. (iv) povratna razmjena vrijednosti

- a. 10,20; 0,10,20

- b. 100,20; 0,101,20
- c. 10,100; 0,101,100
- d. 10,20; 0,101,20**
- e. 10,20; 0,101,100

66. Navedena leksička pravila opišite regularnim izrazom (pripadni regularni izraz dopišite pored/ispod svakog pravila):

- a. Ključna riječ *ako*: **ako**
- b. Imena identifikatora koja počinju znakom *a* ili *b* i nastavljaju se nizom znakova *a*, *b* i *c* proizvoljne duljine:  **$(a|b)(a|b|c)^*$**
- c. Brojeve konstante koje se zapisuju u oktalnoj bazi pri čemu konstanta počinje znamenkom 0:  **$0(0|1|...|7)^+$**

67. U specifikaciji leksičkog analizatora zadani su regularni izrazi sljedećim redoslijedom:

$(a|b|...|z)^*(0|1|...|9)$ , dva/2, tri/3 i pet/5. Kako se niz **petpet5pet5pet3** grupira u leksičke jedine?

- a. petpet, 5, pet, 5, pet, 3
- b. petpet5, pet5, pet3**
- c. pet, pet5, pet5, pet, 3
- d. pet, pet, 5, pet, 5, pet, 3
- e. petpet5, pet, 5, pet, 3

68. Zadana je Q-gramatika s produkcijama:  $S \rightarrow bAS$ ,  $S \rightarrow a$ ,  $A \rightarrow \varepsilon$ . Odredite skup  $PRIMIJEI(A \rightarrow \varepsilon)$ .

- a.  $\{\varepsilon\}$
- b.  $\{a\}$
- c.  $\{b\}$
- d.  $\{a, b\}$**
- e.  $\{a, b, \varepsilon\}$

69. Koji se od navedenih postupaka koristi u pretvorbi LL(1) gramatike u Q-gramatiku?

- a. Zamjena krajnje desnih nezavršnih znakova na lijevoj strani produkcije.
- b. Zamjena krajnje desnih nezavršnih znakova na desnoj strani produkcije.
- c. Zamjena krajnje lijevih nezavršnih znakova na lijevoj strani produkcije.
- d. Zamjena krajnje lijevih nezavršnih znakova na desnoj strani produkcije.**
- e. Zamjena završnih znakova nezavršnim znakovima.

70. Do kakvog inače mogućeg proturječja nikada ne može dovesti grupiranje stanja koja imaju iste LR(0) stavke?

- a. Pomakni/Pomakni
- b. Pomakni/Reduciraj**
- c. Reduciraj/Reduciraj
- d. Pomakni/Odbaci
- e. Reduciraj/Odbaci

71. Relacija  $ReduciranZnakom(A, x)$  vrijedi ako je znak *A* početni nezavršni znak gramatike, a znak *x* je:

- a. završni ili nezavršni znak gramatike
- b. nezavršni znak gramatike
- c. završni znak gramatike
- d. oznaka kraja niza**
- e. oznaka dna stoga

72. Dio izvornog programa u kojem je važeća deklaracija naziva se \_\_\_\_\_ deklaracije.

- a. Djelokrug



73. Osnovne razine međukoda redom od onih bližem strojnom kodu.

- a. niži, srednji, viši

Sljedeća pitanja su iz MI, ali mi se čine kao varijacija pitanja koja su bila na ZI

1. Zadana je produkcija L-atributne prijevodne gramatike:  $X_{n_5, i_1, i_2} \rightarrow Y_{n_6} Z_{i_3} V_{i_4, n_7, n_8} W_{n_9}$  gdje su  $i_1, i_2, i_3$  te  $i_4$  izvedena svojstva, a  $n_5, n_6, n_7, n_8$  i  $n_9$  su nasljedna svojstva. Nasljedno svojstvo  $n_7$  može se računati na temelju svojstava:

- a.  $n_5, i_1, i_2$
- b.  $i_2, i_3$
- c.  $n_5, n_6, n_8, n_9$
- d.  $n_6, i_1, i_2, n_6, i_3$
- e.  **$n_5, n_6, i_3$**

2. Neka je leksički analizator zasnovan na regularnim izrazima iz sljedeće tablice:

| Oznaka    | Regularni izraz | Akcija      |
|-----------|-----------------|-------------|
| <b>R1</b> | $a(b c) * a$    | Ispiši „R1“ |
| <b>R2</b> | $b(a c) * b$    | Ispiši „R2“ |
| <b>R3</b> | $c(b a) * c$    | Ispiši „R3“ |

Koji od ponuđenih nizova predstavlja izlaz leksičkog analizatora na sljedećem ulaznom nizu: *bbaacabc*?

- a. R1R2R3
- b. R3R1R2
- c. R1R3R2
- d. R3R2R1
- e. **R2R1R3**

3. Pretraživanje desnog konteksta u leksičkoj analizi ostvaruje se primjenom:

- a. Odbacivanjem krajnje lijevog nezavršnog znaka
- b. Dodatnih stanja simulatora
- c. Odbacivanjem krajnje desnog znaka
- d. **Regularnih izraza oblika  $r/r'$**
- e. Potisnog automata

4. Koja je od sljedećih produkcija moguća unutar neke S-gramatike?

- a.  $S \rightarrow Cd$
- b.  $B \rightarrow XSY$
- c.  $C \rightarrow C$
- d.  **$B \rightarrow aBBa$**
- e.  $A \rightarrow \varepsilon$

5. Što se od navedenog primarno koristi u leksičkoj analizi?

- a. LR parser
- b. LL(1) parser
- c. Potisni automat
- d. Co-No tablica
- e. **Konačni automat**

6. Za sljedeću kontekstno neovisnu gramatiku  $S \rightarrow pAmC$ ;  $A \rightarrow dSa$ ;  $C \rightarrow dA$ ;  $S \rightarrow bA$ ;  $A \rightarrow e$ ; izračunajte vrijednost relacije *IspodZnaka* za završni znak  $p$ .

- a. *IspodZnaka*( $p, d$ )

- b.  $IspodZnaka(p, d), IspodZnaka(p, m)$
  - c.  $IspodZnaka(p, a), IspodZnaka(p, e)$
  - d.  $IspodZnaka(p, e)$
  - e.  **$IspodZnaka(p, d), IspodZnaka(p, e)$**
7. Kako se niz  **$abc42de1$**  grupira u leksičke jedinice ako je u specifikaciji leksičkog analizatora zadan regularni izraz  $(a|b| \dots |z) * (1|2| \dots |9)$ ?
- a.  $abc42de1$
  - b.  **$abc4, 2, de1$**
  - c.  $abc32, de1$
  - d.  $a, b, c, 42, d, e, 1$
  - e.  $abc4, 2, de, 1$
8. U kojem se dijelu jezičnog procesora (prevoditelja) primjenjuje sljedeće pravilo odgovarajućeg programskog jezika:  
*Ako su varijable A i B cjelobrojne, onda je i varijabla C u naredbi  $C = A + B$  cjelobrojna.*
- a. U pripremi izvođenja
  - b. U sintaksoj analizi
  - c. U leksičkoj analizi
  - d. U optimiranju
  - e. **U semantičkoj analizi**
9. Pomakni/Pronađi gradi se izravno na temelju relacija
- a. Slijedi i Primjeni
  - b. Započinje i Primjeni
  - c. IspredZnaka i ReduciranZnakom
  - d. **IspodZnaka i ReduciranZnakom**
10. Ako je stanje  $LR(1)$  parsera označeno stavkama  $X \rightarrow \alpha \bullet \{a, b, c\}$  i  $X \rightarrow \alpha \bullet g\beta\{d, e, f\}$ , parser će učitavanjem znaka  $e$  iz ulaznog niza izvesti akciju:
- a. *pomakni*
  - b. *reduciraj*
  - c. *prihvati*
  - d. ***odbaci***
  - e. neće moći jednoznačno odlučiti zbog proturječja
11. Poredajte gramatike LALR(1), SLR(1), LR(0) i LR(1) uzlazno po općenitosti. Razred gramatika X je općenitiji od razreda gramatika Y ako se svaki jezik definiran gramatikom Y može definirati gramatikom iz razreda X.
- a. LALR(1), SLR(1), LR(0) i LR(1)
  - b. LR(0), LALR(1), SLR(1) i LR(1)
  - c. LR(0), SLR(1), LR(1) i LALR(1)
  - d. **LR(0), SLR(1), LALR(1) i LR(1)**
  - e. SLR(1), LR(0), LALR(1) i LR(1)
12. U kojem se dijelu jezičnog procesora (prevoditelja) primjenjuje sljedeće pravilo odgovarajućeg programskog jezika:  
*Identifikator sadrži slova i znamenke, a započinje slovom.*
- a. U semantičkoj analizi
  - b. U sintaksoj analizi
  - c. U optimiranju
  - d. U pripremi izvođenja
  - e. **U leksičkoj analizi**