

# Rješenja nekih zadataka

Ponavljanje 1

## 2. Objasnite kratko što će se dogoditi u svakoj liniji sljedećeg koda:

```
from Stack import *
s = Stack()
print(s.isEmpty())
s.push(2)
s.push(1)
s.push(9)
print(s)
s.pop()
s.push(3)
s.push(12)
s.pop()
print(s)
print(s.pop())
```

Kratko objašnjenje:

Uključujemo klasu stack u program i kreiramo prazan stog.

Metoda isEmpty() nam vraća true jer je trenutno stog prazan. Metodom push() stavljamo 2, 1 i 9 na stog. Ispisuje se stog: [2, 1, 9]. Zatim s pop() mičemo 9 i stavljamo opet na stog 3 i 12 te na kraju mičemo s pop() 12. Printa se [2, 1, 3], a onda miče sa stoga i printa 3. s = [2, 1]

### 3. Objasnite kratko što će se dogoditi u svakoj liniji sljedećeg koda:

```
from Queue import *
q = Queue()
q.enqueue("A")
q.enqueue("X")
q.enqueue("C")
print(q.isEmpty())
q.enqueue("Y")
q.enqueue("Z")
q.dequeue()
q.dequeue()
print(q)
q.enqueue("B")
q.dequeue()
q.dequeue()
print(q)
```

Kratko objašnjenje:

Uključujemo klasu Queue u program i kreiramo prazan red. U red stavljamo redom slova A, X, C pomoću metode enqueue(). Metoda isEmpty() nam vraća False jer imamo neke elemente u redu. Opet stavljamo u red Y i Z, a s dequeue() mičemo prve koje smo stavili to su A i X. Ispisujemo [C, Y, Z]. Dodajemo B i mičemo redom C i Y. Na kraju je u redu spremljeno i ispisano samo [Z, B]

# Rješenja nekih zadataka

Ponavljanje 3

1.

PREFIX: - \* - 7 2 4 10      POSTFIX: 7 2 - 4 \* 10 -

2.

PREFIX: - 8 / - 15 3 + 4 2      POSTFIX: 8 15 3 - 4 2 + / -

3.

$((1+3)/(2-1))*5$

4.

$(19 - (1 + 2)*3) / 5$

5. Rj. 1

6. Rj. 2

7.

a) C

b) 3

c) A, D; G

d) 0. razina; 2. razina

e) F

f) A, G, E, H, J

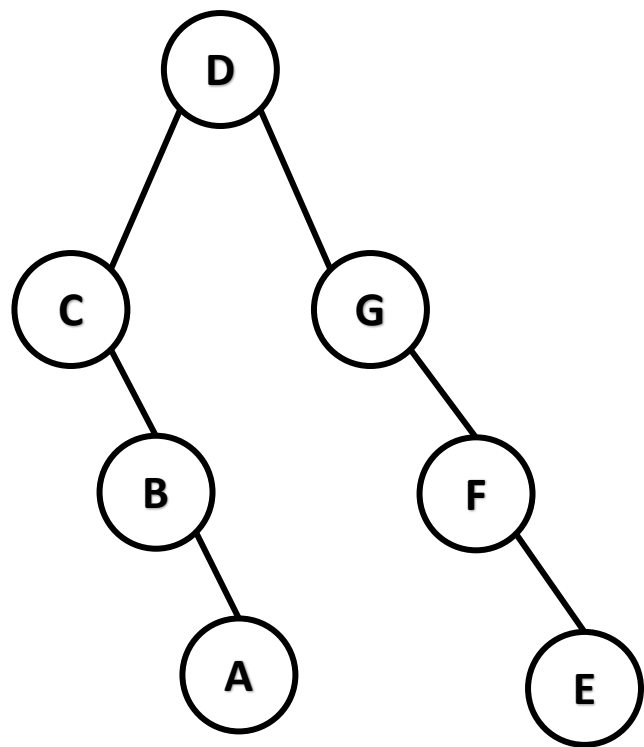
g) Nije puno niti potpuno

h) CBADGFEIHJ

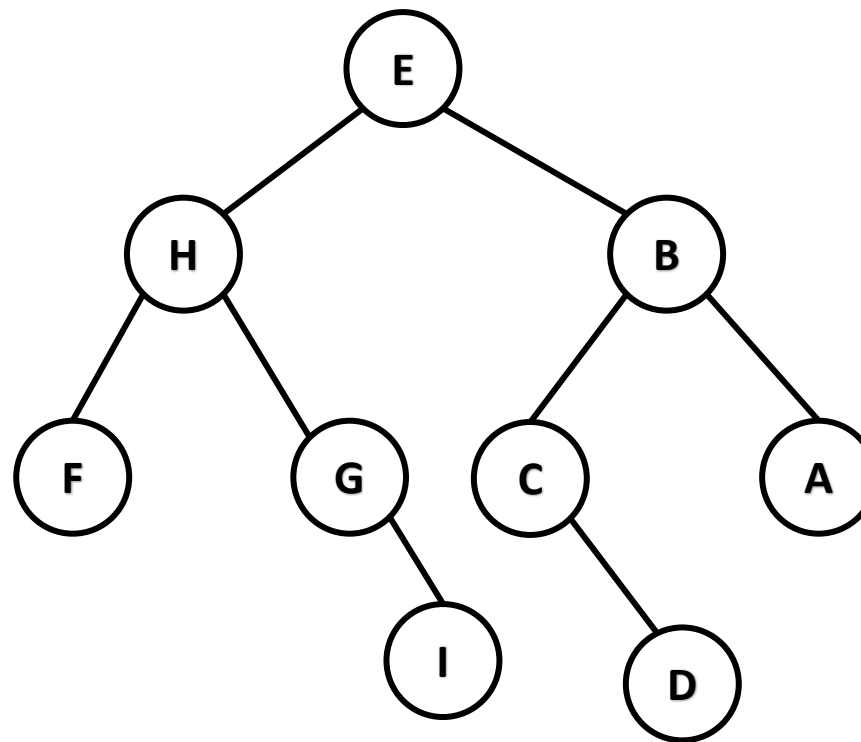
i) ABDGCEFHIJ

j) AGDBEHJIFC

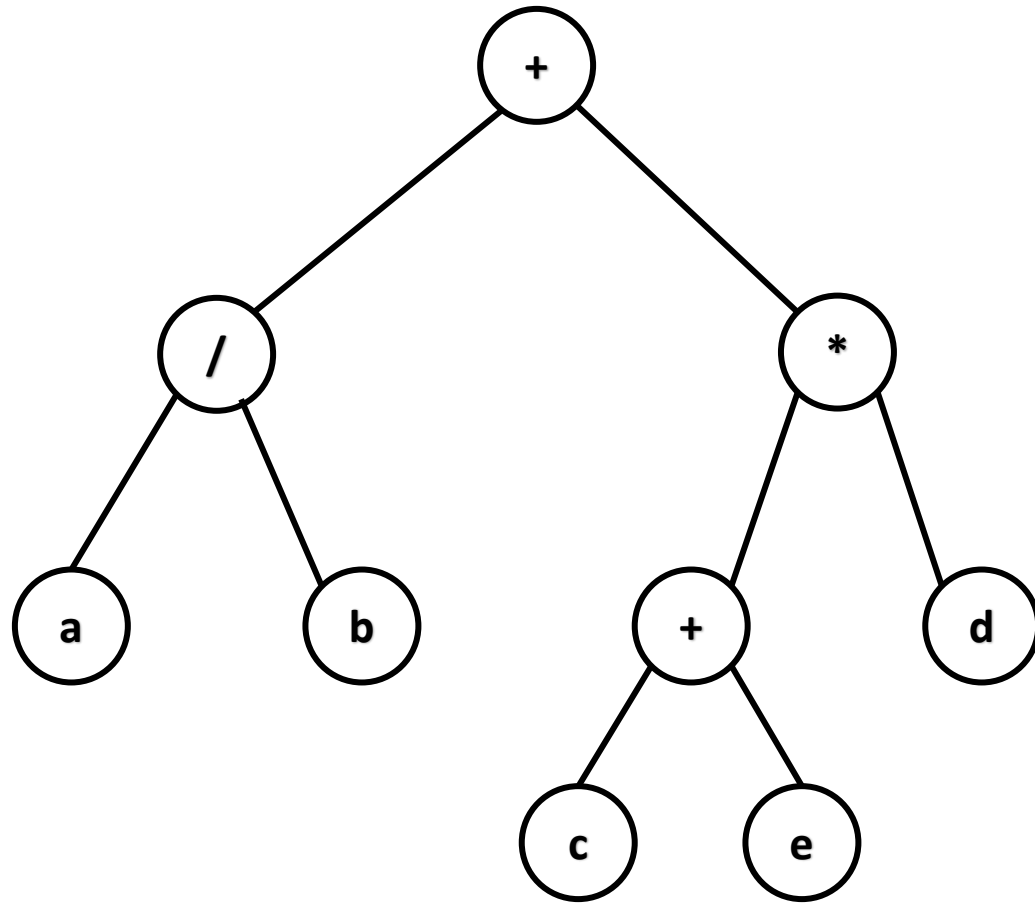
8.



9.



10.



PREORDER: + / a b \* + c e d    POSTORDER: a b / c e + d \* +