TEST 1

1. Šta je rekurzija?

   - Pod rekurzivnim metodom se podrazumeva metod koji poziva sam sebe. Rekurzija omogućava znatno jednostavnije rešavanje pojedinih problema.

2. Šta podrazumeva rekurzivno rešavanje problema?

   - **Rekurzivno rešavanje problema** podrazumeva da definišemo programski deo za jedan element složene strukture, pa da ga onda višestruko pozivamo da to ponovi. Kako se pozivi vrše iz prethodno relizovanog programskog dela, to se rekurzija hijerarhijski ponavlja.

3. Šta je rekurzivni metod?

   - Pod **Rekurzivnim metodom** se podrazumeva metoda koji poziva sam sebe.

4. Šta je beskonačna rekurzija?

   - **Bekonačna rekurzija** je rekurzija koja se nikada ne završava.

5. Koliko puta se poziva metoda factorijal() u programu datom za klasu ComputeFactorijal?

   - Poziva se sve dok se parametar koji se prenosi u samom metodu i smanjuje za 1, ne smanji na vrednost 0.

6. Šta je Direktna rekurzija?

   - **Direktna rekurzija** je rekurzija kada metoda poziva sam sebe.

7. Šta je indirektna rekurzija?

   - **Indirektna rekurzija** je kada jedan metoda poziva drugi metoda a on oet poziva prvo metod.

8. Kako biste problem Fibonačijevog niza podelili na rekurzivne potprobleme?

   - Podelili bismo ga na dva potproblema:
     1) Određivanje brojeva fib(index - 2) i
     2) Određivaenj brojeva fib(index - 1).

9. Koliko će metoda fib biti pozvan u listingu koji određuje Fibonačijeve brojeve za fib(6)?

- 12 puta. Jer u opštem slučaju računanje fib(index) zahteva 2 puta više rekurzivnih poziva, nego što bi broj rekurzivnihpoziva imalo izvršenje fib(index – 1).

10. Opišite karakteristike rekurzivnih metoda.

- Karakteristike su:
  1) Metod upotrebljava if-else ili swich iskaze koji vode do različitih slučajeva.
  2) Jedan ili više osnovnih slučajeva se koriste za zaustavljanje rekurzije.
  3) Svaki rekurzivni poziv smanjuje početni problem, dovodeći ga bliže do osnovnog slučaja, sve dok ne postane i on takav slučaj.

11. Koji je osnovni slučaj (base case) metoda isPŠalindrome() opisanog u materijalima? Koliko puta će ovaj metod bit pozvan za slučaj poziva isPalindrome(„abdxcxdba“)?

- Osnovni slučaj: “Provera da li je prva oznaka (character) jednaka poslednjoj oznaci stringa“. Ovo je ujedno i prvi potproblem a drugi potproblem je da li string dužine 0 ili 1. Metod će biti pozvan 5 puta.

12. Nacrtati ili napisati stek poziva (call stack) za slučaj da se funkcija poziva funkcija isPalindrome(„abcba“)?

- Poziva se 3 puta: fib(“abcba”) → fib(“bcb”) → fib(“c”)

13. Šta je rekurzivni pomoćni mteod (recursive helper method)?

- Praksa pri rekurzivnm programiranju je da se definiše drugi metoda koji prima dodatne parametre. Takav metod je poznat kao **rekurzivni pomoćni metod**. Ovi metodi su korisni kod rekuzivnog rešavanja problema koji se javljaju sa stringovima i nizovima.

14. Kako biste problem Hanojskih kula podelili na rekurzivne potprobleme?

- Tako što stubove (podstabla) prebacimo pomoću sledećeg algoritma:
  1) Prebaciti stablo koje sadrži n-1 diskova sa A na B.
  2) Preostali (najveći) disk prebaciti sa A na C.
  3) Prebaciti podstablo sa B na C.

15. Koliko puta će metoda moveDiscs() biti pozvam za slučaj moveDiscs(5, 'A','B','C')?

- 31 put.

16. Koja od sledećih tvrdnji je tačna?

- Rekurzivni metodi zahtevaju više vremena I memorije za izvršenje nego nerekurzivni metodi.

17. Šta je repna rekurzija?

- Pod **Repnom rekurzijom** (tail recursion), podrazumeva se rekurzija koja nema zaostale operacije na čekanju a koje bi trebalo izvršiti po završetku jednog rekurzivnog poziva.

18. Kada biste koristili repnu rekurziju?

- Repnu rekurziju bih koristio kada treba na efikasan način da smanjim potrebnu veličinu steka.

19. Identifikovati primere ove lekcije koji koriste repnu rekurziju?

- Klasa ComputeFactorialTailRecursion ima pomoćni repni rekurzivni metoda za faktorijel po imenu factorial().

TEST 2

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

TEST 3

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

TEST 4

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

TEST 5

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

TEST 6

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.