

Teorija

Šta je algoritam?

- opis obrade podataka u kom su precizirane operacije od kojih se obrada sastoji i redosled njihovog obavljanja
 - postupak izražen u programskom jeziku
 - postupak pravljenja programa
-

Kako se naziva opis obrade podataka u kome su precizirane operacije od kojih se obrada sastoji i redosled njihovog obavljanja?

- algoritam
 - izvršilac
 - programiranje
 - programski jezik
-

Izvršilac algoritma može biti:

- isključivo čovek
 - isključivo računar
 - oba ponuđena
-

Kako se naziva algoritam izražen u programskom jeziku?

- program
 - izvršilac
 - programiranje
 - programski jezik
 - binarni kod
 - proces
-

Skup svih algoritama je:

- prebrojiv
 - neprebrojiv
 - kontinualan
 - konačan
-

Ako su dva algoritma funkcionalno ekvivalentni, tada im se poklapaju i:

- kodeksi
- algoritamski sistemi
- algoritamska teza

U sintetičkoj definiciji algoritma, algoritam je uređena četvorka (X, Y, G, Z) . Šta predstavlja G ?

- alfabet
- ulazni alfabet
- izlazni alfabet
- kodeks

Element dijagrama toka sa jednom ulaznom i jednom izlaznom linijom u kome se obavlja obrada ili prenos podataka naziva se:

- proces
- predikat
- kolektor

Koja je osnova binarnog brojnog sistema?

- 2
- 4
- 6
- 8
- 10
- 16
- {0, 1}
- {0, 1, 2, 3, 4, 5, 6, 7}
- {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Koja je osnova oktalnog brojnog sistema?

- 8
- 2
- 4
- 6
- 10
- 16
- {0, 1}
- {0, 1, 2, 3, 4, 5, 6, 7}
- {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Koja je osnova heksadecimalnog brojnog sistema?

- 16
- 2
- 4
- 6

- 8
 - 10
 - {0, 1}
 - {0, 1, 2, 3, 4, 5, 6, 7}
 - {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
-

Koje su cifre binarnog brojnog sistema?

- {0, 1}
 - 2
 - 4
 - 6
 - 8
 - 10
 - 16
 - {0, 1, 2, 3, 4, 5, 6, 7}
 - {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
-

Koje su cifre oktalnog brojnog sistema?

- {0, 1, 2, 3, 4, 5, 6, 7}
 - 2
 - 4
 - 6
 - 8
 - 10
 - 16
 - {0, 1}
 - {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
-

Koje su cifre heksadecimalnog brojnog sistema?

- {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
 - 2
 - 4
 - 6
 - 8
 - 10
 - 16
 - {0, 1}
 - {0, 1, 2, 3, 4, 5, 6, 7}
-

Parametri funkcije su:

- lokalne promenljive
- globalne promenljive

- nisu promenljive
 - neinicijalizovane lokalne promenljive
 - neinicijalizovane globalne promenljive
-

Oblast važenja parametra funkcije je:

- funkcija
 - oblast važenja parametra nije ograničena
 - zaglavlje funkcije
 - celokupna datoteka izvornog koda
-

Listanje sadržaja direktorijuma u Linux terminalu vrši se upotrebom komande:

- ls
 - cd
 - mkdir
-

Promena direktorijuma u Linux terminalu vrši se upotrebom komande:

- cd
 - dir
 - ls
 - mkdir
-

Prelazak u roditeljski direktorijum u Linux terminalu vrši se upotrebom komande:

- cd ..
 - cd
 - cd .
 - cd ./
 - cd ~
 - dir ~
 - ls .
 - mkdir ..
-

Kompajliranje u Linux terminalu vrši se upotrebom komande:

- gcc -o zadatak zadatak.c
 - gcc -o zadatak zadatak
 - gcc -o zadatak.c zadatak
 - gcc -o zadatak.c zadatak.c
-

Kompajliranje, sa uključivanjem matematičke biblioteke, u Linux terminalu vrši se upotrebom komande:

- gcc -o zadatak zadatak.c -lm

- `gcc -o zadatak zadatak.c`
 - `gcc -o zadatak.c zadatak`
 - `gcc -o zadatak.c zadatak -lm`
-

U Linux terminalu, nakon poziva komande `gcc zadatak.c`, dobijena izvršna datoteka pokreće se komandom:

- `./a.out`
 - `a.out`
 - `zadatak`
 - `/zadatak`
 - `./zadatak`
 - `./zadatak.c`
-

U Linux terminalu, nakon poziva komande `gcc -o zadatak zadatak.c`, dobijena izvršna datoteka pokreće se komandom:

- `./zadatak`
 - `a.out`
 - `./a.out`
 - `zadatak`
 - `/zadatak`
 - `./zadatak.c`
-

Kako se može uneti prvih 8 vrednosti niza realnih brojeva `a`?

- `for(i=0;i<8;i++) scanf("%f",&a[i]);`
 - `for(i=0;i<8;i++) scanf("%f",a[i]);`
 - `for(i=0;i<9;i++) scanf("%f",a[i]);`
 - `for(i=0;i<9;i++) scanf("%f",&a[i]);`
-

Da li je `NULL` pokazivač isto što i neinicijalizovani pokazivač?

- `ne`
 - `da`
-

Unarni operator referenciranja `&`:

- **vraća adresu promenljive**
 - omogućuje posredan pristup nekom podatku putem njegove memorijske adrese
 - vraća sadržaj pokazivačke promenljive
-

Unarni operator dereferenciranja `*`:

- **omogućuje posredan pristup nekom podatku putem njegove memorijske adrese**
- vraća adresu promenljive

- vraća sadržaj pokazivačke promenljive
-

Pokazivač tipa `void *` može da predstavlja:

- adresu bilo kog objekta
 - vrednost bilo kog objekta
 - tip bilo kog objekta
-

Ograničeni pokazivač dobija se primenom kvalifikatora:

- `restrict`
 - `extern`
 - `const`
 - `inline`
-

Između ograničenog pokazivača i objekta na koji on pokazuje postoji sledeća veza:

- tokom životnog veka pokazivača, objekat se može menjati ili mu se može pristupati samo putem tog pokazivača
 - tokom životnog veka pokazivača, objekat se ne može menjati
 - tokom životnog veka objekta, pokazivač se ne može menjati
-

Promenljivi pokazivač na promenljive podatke se deklarise sa:

- `tip *p`
 - `tip *const p`
 - `const tip *p`
 - `const tip *const p`
-

Nepromenljivi pokazivač na promenljive podatke se deklarise sa:

- `tip *const p`
 - `tip *p`
 - `const tip *p`
 - `const tip *const p`
-

Promenljivi pokazivač na nepromenljive podatke se deklarise sa:

- `const tip *p`
 - `tip *p`
 - `tip *const p`
 - `const tip *const p`
-

Nepromenljivi pokazivač na nepromenljive podatke se deklarise sa:

- `const tip *const p`
 - `tip *p`
 - `tip *const p`
 - `const tip *p`
-

Pokazivač `p` iz izraza `tip *p` je:

- **promenljivi pokazivač na promenljive podatke**
 - nepromenljivi pokazivač na promenljive podatke
 - promenljivi pokazivač na nepromenljive podatke
 - nepromenljivi pokazivač na nepromenljive podatke
-

Pokazivač `p` iz izraza `tip *const p` je:

- **nepromenljivi pokazivač na promenljive podatke**
 - promenljivi pokazivač na promenljive podatke
 - promenljivi pokazivač na nepromenljive podatke
 - nepromenljivi pokazivač na nepromenljive podatke
-

Pokazivač `p` iz izraza `const tip *p` je:

- **promenljivi pokazivač na nepromenljive podatke**
 - promenljivi pokazivač na promenljive podatke
 - nepromenljivi pokazivač na promenljive podatke
 - nepromenljivi pokazivač na nepromenljive podatke
-

Pokazivač `p` iz izraza `const tip *const p` je:

- **nepromenljivi pokazivač na nepromenljive podatke**
 - promenljivi pokazivač na promenljive podatke
 - nepromenljivi pokazivač na promenljive podatke
 - promenljivi pokazivač na nepromenljive podatke
-

Kako se pravilno uvezuje pokazivač `pi` na promenljivu `i`?

- **`pi = &i;`**
 - `pi = i;`
 - `*pi = i;`
 - `*pi = &i;`
-

U programskom jeziku C dozvoljeno je dodavanje proizvoljne celobrojne vrednosti na postojeću vrednost pokazivača.

- **tačno**

- netačno
- dozvoljena je isključivo upotreba operatora inkrementatora i dekrementatora

Šta od navedenog ne može da bude naziv promenljive u programskom jeziku C?

- `lbrojac`
 - `_123456`
 - `JakoDugackiNazivPromenljive`
 - `jako_dugacak_naziv_promenljive`
-

Šta od navedenog može da bude naziv promenljive u programskom jeziku C?

- `Break`
 - `goto`
 - `continue`
 - `2drugi_op`
 - `prvi-op`
-

Šta od navedenog može da bude naziv promenljive u programskom jeziku C?

- `prvi_op`
 - `break`
 - `goto`
 - `continue`
 - `2drugi_op`
-

Šta od navedenog može da bude naziv promenljive u programskom jeziku C?

- `arm64`
 - `goto`
 - `continue`
 - `2drugi_op`
 - `Break`
-

Kako se pravilno ispisuje promenljiva dužine osam bita, u decimalnom formatu:

- `printf("%d", promenljiva);`
 - `printf("%b", promenljiva);`
 - `printf("%c", promenljiva);`
 - `printf("%c", &promenljiva);`
 - `printf("%d", &promenljiva);`
 - `scanf("%b", promenljiva);`
 - `scanf("%c", promenljiva);`
 - `scanf("%c", &promenljiva);`
 - `scanf("%d", promenljiva);`
 - `scanf("%d", &promenljiva);`
-

Kako se pravilno ispisuje promenljiva dužine jedan bajt, u ASCII formatu:

- `printf("%c", promenljiva);`
 - `printf("%b", promenljiva);`
 - `printf("%c", &promenljiva);`
 - `printf("%d", promenljiva);`
 - `printf("%d", &promenljiva);`
 - `scanf("%b", promenljiva);`
 - `scanf("%c", promenljiva);`
 - `scanf("%c", &promenljiva);`
 - `scanf("%d", promenljiva);`
 - `scanf("%d", &promenljiva);`
-

Kako se pravilno učitava promenljiva dužine jedan bajt, u decimalnom formatu:

- `scanf("%d", &promenljiva);`
 - `printf("%b", promenljiva);`
 - `printf("%c", promenljiva);`
 - `printf("%c", &promenljiva);`
 - `printf("%d", promenljiva);`
 - `printf("%d", &promenljiva);`
 - `scanf("%b", promenljiva);`
 - `scanf("%c", promenljiva);`
 - `scanf("%c", &promenljiva);`
 - `scanf("%d", promenljiva);`
-

Kako se pravilno učitava promenljiva dužine osam bita, u ASCII formatu:

- `scanf("%c", &promenljiva);`
 - `printf("%b", promenljiva);`
 - `printf("%c", promenljiva);`
 - `printf("%c", &promenljiva);`
 - `printf("%d", promenljiva);`
 - `printf("%d", &promenljiva);`
 - `scanf("%b", promenljiva);`
 - `scanf("%c", promenljiva);`
 - `scanf("%d", promenljiva);`
 - `scanf("%d", &promenljiva);`
-

Kako se pravilno učitava celobrojna promenljiva n:

- `scanf("%d", &n);`
- `scanf("%d", n);`
- `scanf("%d", %n);`
- `scanf("%f", &n);`
- `scanf("%n");`
- `scanf("d", &n);`

- `scanf(n);`
 - `scanf(&n);`
-

Kako se pravilno ispisuje promenljiva tipa `double`:

- `printf("%lf", celzijusi);`
- `printf("%.2", celzijusi);`
- `printf("%.2", &celzijusi);`
- `printf("%d", celzijusi);`
- `printf("%d", &celzijusi);`
- `printf("%lf", &celzijusi);`

Data je reč "`0123456789`", najmanji string u koji se ona može upisati u celosti je:

- `char str[11]`
 - `char str[9]`
 - `char str[10]`
-

Data je reč "`012346789`", najmanji string u koji se ona može upisati u celosti je:

- `char str[10]`
 - `char str[9]`
 - `char str[11]`
-

Data je reč "`123456789`", najmanji string u koji se ona može upisati u celosti je:

- `char str[10]`
 - `char str[9]`
 - `char str[11]`
-

Data je reč "`12346789`", najmanji string u koji se ona može upisati u celosti je:

- `char str[9]`
 - `char str[10]`
 - `char str[11]`
-

Najmanji string u koji se podatak "" može upisati u celosti je:

- `char str[1]`
 - `char str[0]`
 - `char str[2]`
-

Kako se sledeći string može ispisati u celosti na standardni izlaz:

```
char str[] = "Primer prvog stringa";
```

- sa oba ponuđena
 - isključivo sa `printf("%s", str);`
 - isključivo sa `puts(str);`
-

Kako se sledeći string može ispisati u celosti na standardni izlaz:

```
char str[] = "Primer prvog stringa";
```

- isključivo sa `puts(str);`
 - isključivo sa `gets(str);`
 - sa oba ponuđena
-

Kako se sledeći string može učitati u celosti sa standardnog ulaza:

```
char str[] = "Primer prvog stringa";
```

- isključivo sa `gets(str);`
 - isključivo sa `scanf("%s", str);`
 - sa oba ponuđena
-

Kako se sledeći string može učitati u celosti sa standardnog ulaza:

```
char str[] = "Primer_prvog_stringa";
```

- sa oba ponuđena
 - isključivo sa `gets(str);`
 - isključivo sa `scanf("%s", str);`
-

Kako se sledeći string može učitati u celosti sa standardnog ulaza:

```
char str[] = "Primer prvog stringa";
```

- isključivo sa `gets(str);`
 - isključivo sa `puts(str);`
 - sa oba ponuđena
-

Nakon poziva funkcije `strcmp(str1, str2)` dobili smo povratnu vrednost 0, što nam govori da:

- su `str1` i `str2` istog sadržaja
 - su `str1` i `str2` različiti
 - je `str1` manji od `str2`
 - je `str2` manji od `str1`
 - smo uspešno kopirali vrednost stringa `str2` u `str1`
 - smo uspešno kopirali vrednost stringa `str1` u `str2`
-

Nakon poziva funkcije `strcmp(str1, str2)` dobili smo povratnu vrednost 1, što nam govori da:

- su `str1` i `str2` različiti
 - su `str1` i `str2` istog sadržaja
 - smo uspešno kopirali vrednost stringa `str2` u `str1`
 - smo uspešno kopirali vrednost stringa `str1` u `str2`
-

Koji od navedenih znakova nije podržan u tipu `char`?

- svi navedeni znakovi su podržani
 - '0'
 - 'Q'
 - '#'
 - '?'
 - '%'
-

Binarni hip je:

- nelinearna struktura
 - linearna struktura
 - LIFO
 - FIFO
 - FUFU
-

Kojim bitwise operatorom se postavlja vrednost željenog bita na 0?

- `&`
 - `~`
 - `!`
 - `&&`
 - `|`
 - `||`
 - `^`
-

Kojim bitwise operatorom se dobavlja vrednost željenog bita?

- `&`
 - `~`
 - `!`
 - `&&`
 - `|`
 - `||`
 - `^`
-

Kojim bitwise operatorom se invertuje vrednost željenog bita?

- `^`

- ~
- !
- &
- &&
- |
- ||

Koja će biti vrednost izraza $1 \ \& \ 31$?

- 1
- 0
- 30
- 31
- 32
- 33

Koja će biti vrednost izraza $0 \ \&\& \ 31$?

- 0
- 1
- 30
- 31
- 32
- 33

Koja će biti vrednost izraza $1 \ | \ 32$?

- 33
- 0
- 1
- 30
- 31
- 32

Koja će biti vrednost izraza $0 \ || \ 31$?

- 31
- 0
- 1
- 30
- 32
- 33

Koja će biti vrednost izraza $1 \ \wedge \ 31$?

- 30
 - 0
 - 1
 - 31
 - 32
 - 33
-

U programskom jeziku C za pristup datotekama koriste se:

- funkcije standardne biblioteke
 - naredbe programskog jezika
 - programski jezik C ne podržava rad sa datotekama, već samo sa standardnim tokovima podataka
-

Deskriptor spregnute realizacije deka sadrži:

- dva pokazivača
 - jedan pokazivač
 - tri pokazivača
 - deskriptor reda ne sadrži pokazivače
-

Prilikom spregnute realizacije deka memorija za čvorove se zauzima:

- dinamički
 - statički
 - sekvencijalno
-

U programskom jeziku C za dinamičko upravljanje memorijom koriste se:

- funkcije standardne biblioteke
 - naredbe programskog jezika
 - programski jezik C ne podržava dinamičko upravljanje memorijom
-

Prilikom definisanja skalarnih promenljiva memorija se zauzima:

- statički
 - dinamički
 - upotrebom funkcije `malloc`
 - upotrebom funkcije `free`
-

Prilikom definisanja spregnutih struktura memorija se zauzima:

- dinamički
- statički
- sekvencijalno

"Curenje" memorije je efekat koji nastaje u slučaju:

- dinamičkog zauzimanja memorije, bez njenog pravovremenog oslobađanja
- statičkog zauzimanja memorije
- nedovoljne količine memorije u računaru
- čestih poziva funkcije `free`

U programskom jeziku C viseći pokazivači:

- pokazuju na deo memorije koji više nije zauzet
- nalaze se na kraju kružne liste
- koriste se za završetak deka
- koriste se za završetak liste
- rešavaju problem lažne popunjenosti liste
- rešavaju problem lažne popunjenosti reda

U poslednjem čvoru dvostruko spregnute liste, vrednost pokazivača na sledeći čvor je:

- `NULL`
- `END`
- `EOF`
- `FF`
- `glava`
- `void`

Drugi naziv za dvostruko spregnute liste je:

- simetrična lista
- lista sa zaglavljem
- pokazivačka lista
- težinska lista

Za operaciju brisanja čvora iz dvostruko spregnute liste dovoljno je znati:

- pokazivač na čvor koji brišemo (`TEKUCI`)
- pokazivač na prethodni čvor (`PRETHODNI`)
- pokazivač na prethodni čvor (`PRETHODNI`) i pokazivač na čvor koji brišemo (`TEKUCI`)
- pokazivač na sledeći čvor (`SLEDECI`)
- pokazivač na sledeći čvor (`SLEDECI`) i pokazivač na čvor koji brišemo (`TEKUCI`)
- pokazivač na sledeći čvor (`SLEDECI`) i pokazivač na prethodni čvor (`PRETHODNI`)

U poslednjem čvoru jednostruko spegnute liste, vrednost pokazivača na sledeći čvor je:

- `NULL`
- `END`

- EOF
 - FF
 - glava
 - void
-

Koja vrsta liste poseduje specijalni čvor koji: pokazuje na prvi čvor liste, razlikuje se od svih drugih čvorova po specijalnoj vrednosti u svom polju `INFO`, pri čemu ta vrednost polja `INFO` ne može biti validni sadržaj nekog čvora?

- kružna lista sa zaglavljem
 - jednostruko spregnuta lista
 - dvostruko spregnuta lista
 - kružna lista
-

Operacija spajanja dve kružne liste `CONCATENATE (glava1, glava2)` zahteva prolazak do kraja prve liste.

- netačno
 - tačno
 - `CONCATENATE` je operacija koja se ne izvršava nad kružnim listama
-

Dvostruko spregnute liste mogu biti u kružnom obliku.

- tačno
 - netačno
 - samo jednostruko spregnute liste mogu imati kružni oblik
-

U poslednjem čvoru kružne liste, vrednost pokazivača na sledeći čvor je:

- glava
 - END
 - EOF
 - FF
 - NULL
 - Void
-

Za koju strukturu podataka važi sledeće: prvi element nema prethodnika, poslednji element nema sledbenika, a svi ostali elementi imaju i prethodnika i sledbenika?

- linearna lista
 - uopšteno stablo
 - n-arno stablo
 - binarno stablo
 - binarno stablo pristupa
-

Pronalaženje prethodnika (ili sledbenika) posmatranog elementa je operacija nad:

- linearnom listom
 - uopštenim stablom
 - n-arnim stablom
 - binarnim stablom
 - binarnim stablom pristupa
-

Ako se nad informacionim sadržajem elementa može definisati funkcija poređenja i ako su na osnovu te funkcije svi elementi liste poređani u rastućem (ili opadajućem) redosledu, za takvu listu se kaže da je:

- uređena
 - homogena
 - spregnuta
 - sekvencijalna
-

Lista je homogena ako su svi njeni elementi:

- istovetnog tipa
 - istovetne vrednosti
 - popunjeni (tj. nema praznih mesta u listi)
 - u zatom opsegu vrednosti
 - manji od unapred zadate maksimalne vrednosti MAX
 - veći od unapred zadate minimalne vrednosti MIN
-

Kada su elementi linearne liste "razbacani" po memoriji, logički linearni poredak se održava tako što:

- elementi liste sadrže eksplicitnu adresu sebi narednog elementa
 - elemente liste posmatramo sekvencijalno
 - ne može se uspostaviti logički linearni poredak
-

Ubacivanje novog elementa u unutrašnjost linearne liste je značajno jednostavnije kod:

- fizičke realizacije sa direktnim adresiranjem
 - sekvencijalne fizičke realizacije
 - ne zavisi od fizičke realizacije
-

Red je struktura tipa:

- FIFO
 - LIFO
 - FUF
-

Deskriptor spregnute realizacije reda sadrži:

- dva pokazivača
 - jedan pokazivač
 - tri pokazivača
 - deskriptor reda ne sadrži pokazivače
-

Provera da li je red prazan je:

- izvedena operacija nad redom
 - osnovna operacija nad redom
 - operacija koja se ne može izvršiti nad redom
 - primitivna funkcija nad redom
-

Kod kompletnog stabla (reda n):

- svi elementi, osim listova, imaju izlazni stepen n
 - svi putevi od korena do listova su iste dužine
 - broj elemenata na istom nivou stabla razlikuje se najviše za 1
-

Obilazak binarnog stabla u sledećem redosledu je:

```
pristupi levom podstablu  
pristupi desnom podstablu  
pristupi nadredjenom
```

- obilazak sa dna ka vrhu
 - obilazak sa vrha ka dnu
 - obilazak sleva-udesno
-

Obilazak binarnog stabla u sledećem redosledu je:

```
pristupi nadredjenom  
pristupi levom podstablu  
pristupi desnom podstablu
```

- obilazak sa vrha ka dnu
 - obilazak sa dna ka vrhu
 - obilazak sleva-udesno
-

Kako se naziva orijentisani skup čvorova uopštenog stabla, koji su povezani granama:

- put
- lanac
- graf
- digraf

Stek je struktura tipa:

- LIFO
- FIFO
- FUFU

Stek je:

- linearna struktura
- nelinearna struktura
- u zavisnosti od fizičke realizacije može biti ili linearna ili nelinearna struktura

Najjednostavnija implementacija steka je:

- sekvencijalna realizacija
- spregnuta realizacija
- realizacija upotrebom jednostruko spregnute liste
- realizacija upotrebom dvostruko spregnute liste
- realizacija upotrebom binarnog stabla

Prilikom sekvencijalne realizacije steka memorija za čvorove se zauzima:

- statički
- dinamički
- upotrebom funkcije `malloc`
- upotrebom funkcije `free`

Prilikom spregnute realizacije steka memorija za čvorove se zauzima:

- dinamički
- statički
- sekvencijalno

Zajednički imenitelj za sve linearne strukture podataka je:

- postojanje serije podataka jednodimenzionalnog poretka
 - upotreba funkcije `malloc`
 - upotreba funkcije `free`
-

Struktura podataka koja sadrži pokazivač na objekat istog tipa kao što je i ona sama naziva se:

- samo-upućujuća struktura podataka
 - dinamička struktura podataka
 - statička struktura podataka
-

Skalarni tip podatka koji sadrži adresu nekog objekta u memoriji naziva se:

- pokazivač
 - dinamička struktura podataka
 - samo-upućujuća struktura podataka
 - statička struktura podataka
-

Dinamička struktura podataka koja je linearna po svakoj relaciji uspostavljenoj među njenim podacima naziva se:

- lista
 - matrica
 - stablo
 - hip
-

Linearna struktura podataka koja se sastoji od konačno mnogo elemenata istog skalarnog (ili strukturnog) tipa naziva se:

- niz
 - stablo
 - hip
-

Kod koje linearne strukture podataka se poklapaju njena logička i fizička reprezentacija?

- niz
 - stek
 - red
 - dek
-

Koja od sledećih struktura podataka koristi FIFO metodu pristupa?

- red
 - stek
 - n-arno stablo
 - binarno stablo
 - binarno stablo pristupa
 - uopšteno stablo
-

Koja struktura podataka je definisana kao uređeni par $D = (S(D), r(D))$, sa sledećim osobinama:

struktura je linearna

dozvoljen je pristup svakom elementu

ukloniti se mogu samo svi elementi odjednom

novi element se uvek dodaje na kraj

- **sekvenca**

- dek
- red
- stek
- n-arno stablo
- binarno stablo
- binarno stablo pristupa
- uopšteno stablo

Koja linearna struktura podataka intuitivno modelira proces čekanja na kasi u prodavnici?

- **red**

- stek
- dek
- sekvenca
- n-arno stablo
- binarno stablo
- binarno stablo pristupa
- uopšteno stablo

Struktura podataka koja po svojim osobinama predstavlja sintezu reda i steka naziva se:

- **dek**

- red
- stek
- sekvenca
- n-arno stablo
- binarno stablo
- binarno stablo pristupa
- uopšteno stablo

Jednostruko spregnuta lista može se upotrebiti za fizičku realizaciju:

- **i reda i steka**

- reda, ali ne i steka
- steka, ali ne i reda
- ni reda ni steka

U programskom jeziku C jedna struktura ima isto memorijsko zauzeće kao i:

- **zbirno memorijsko zauzeće svih njenih elemenata**

- zauzeće njenog memorijski najzahtevnijeg elementa
 - zauzeće memorijski najzahtevnijeg podržanog tipa podatka u programskom jeziku C
 - zauzeće srodne joj unije, koja se sastoji od istovetnih elemenata
-

U programskom jeziku C jedna unija ima isto memorijsko zauzeće kao i:

- zauzeće njenog memorijski najzahtevnijeg elementa
 - zbirno memorijsko zauzeće svih njenih elemenata
 - zauzeće memorijski najzahtevnijeg podržanog tipa podatka u programskom jeziku C
 - zauzeće srodne joj strukture, koja se sastoji od istovetnih elemenata
-

Za identifikaciju pojedine pozicije u vektoru koristi se:

- indeks
 - pokazivač
 - generički pokazivač
 - viseći pokazivač
-

Dvodimenzionalna generalizacija vektora zove se:

- matrica
 - binarno stablo
 - stek
 - dek
-

Čemu je ekvivalentan izraz `p[i][j]`, ako je dvostruki pokazivač `p` definisan na sledeći način:

```
int **p;  
p = calloc(10, sizeof(int*));
```

- `*(*(p+i)+j)`
 - `*(p+i+j)`
 - `** (p+i+j)`
 - `**p+i+j`
 - `**p+(i+j)`
 - `*p+(i+j)`
-