

1. Заокружити број испред кључне речи којом се при кодирању у програмском језику Ц, у наредби вишеструког гранања обележавају вредности за које се улази у поједине гране:

- 1) case
- 2) break
- 3) return
- 4) switch

РЕШЕЊЕ:

У наредби вишеструког гранања кључна реч switch се користи да прими неку вредност (обично променљиву) а преко case-ова проверавамо која је то вредност и у зависности од тога програм се извршава на различите начине.

Пример:

```
switch(opcija) {
    case 1:
        printf("Izabrana je opcija 1\n");
        break;
    case 2:
        printf("Izabrana je opcija 2\n");
        break;
    case 3:
        printf("Izabrana je opcija 3\n");
        break;
    default:
        printf("Nijedna od ponuđenih opcija nije izabrana\n");
}
```

2. Код дат у тексту задатка треба реализовати помоћу једне if наредбе.

```
if(x>1)
{
    if(x<6)
        y=4; }
```

Заокружити број испред понуђеног тачног одговора:

- 1) if(x<1 && x>6) y=4;
- 2) if(x>1 || x<6) y=4;
- 3) if(x>1 && x<6) y=4;
- 4) if(!(x<=1 || x>=6)) y=4;

РЕШЕЊЕ:

Оператор && (логичко И) се користи када нам је потребно да су услови са обе стране испуњени, док се оператор || (логичко ИЛИ) користи када нам је потребно да је испуњен бар један од услова са обе стране оператора.

3. Заокружити бројеве испред тачних исказа који се односе на дефиницију while циклуса:

- 1) у while циклусу се увек прво проверава да ли је услов логичка истина, те ако јесте наредба се извршава
- 2) while циклус се извршава све док је услов логичка неистина (једнак нули)
- 3) while циклус се користи када се зна колико ће се пута циклус извршавати
- 4) код while циклуса се може десити да се тело циклуса не изврши ниједном (на почетку услов није задовољен).

РЕШЕЊЕ:

Потребно је запамтити да се тело циклуса не мора извршити уколико је услов на почетку логичка неистина, јер се прво проверава дати услов па тек ако је он логичка истина прелази се на тело циклуса.

4. Наведени су искази који се односе на дефиницију do-while циклуса. Заокружити слова испред тачних исказа:

- 1) Користи се када се не зна колико ће се пута циклус понављати
- 2) Прво се извршава тело циклуса, а затим израчунава вредност логичког израза. Ако се добије логичка неистина, циклус се поновно извршава
- 3) Циклус се извршава барем једном
- 4) Циклус се завршава када услов добије вредност логичке истине

РЕШЕЊЕ:

За разлику од while циклуса код do-while циклуса тело самог циклуса ће се извршити бар једном јер се услов проверава тек на крају и затим понавља тело циклуса ако је услов једнак логичкој истини.

5. Дата је if-else наредба:

```
if(a==3 || a==5) p++;  
else p--;
```

Заокружити бројеве испред понуђених switch наредби које су еквивалентне датој if-else наредби:

- | | |
|---|---|
| <p>1) switch(a){
 case 3: p++; break;
 case 5: p++; break;
 default: p--; }</p> <p>2) switch(a){
 case 3: case 5: p++; break;
 default: p--; }</p> | <p>3) switch(a){
 case 3: case 5: p++; break;
 p--; }</p> <p>4) switch(a){
 case 3: case 5: p++;
 default: p--; }</p> |
|---|---|

РЕШЕЊЕ:

Трећи пример је неисправан јер фали кључна реч default, док је четврти пример неисправан јер фали кључна реч break.

6. Одредити вредности које ће променљиве x и y имати по извршењу следећег кода:

```
int x=10;  
int y=20;  
if(x>50)  
    x-=10;  
    y+=10;
```

Уписати добијене вредности на предвиђене линије: x = 10 y = 30

РЕШЕЊЕ:

Променљива x задржава своју вредност (10) јер није испуњен услов (x>50). Променљива y мења своју вредност у 30 јер се због изостанка витичастих заграда услов x>50 не односи на израз y+=10.

7. Са леве страни дати су математички изрази, а са десне запис израза на програмском језику Ц. На линију испред записа израза, унети број којим је означен одговарајући израз:

- | | | |
|---|---|---|
| <p>1. $y = \frac{\sqrt{x+10}}{a+ b }$</p> <p>2. $y = \frac{\sqrt{x+10}}{a} + b$</p> | <p>3. $y = \frac{\sqrt{x+10}}{a+ b }$</p> <p>4. $y = \sqrt{x} + \frac{10}{a} + b$</p> | <p>2 y = sqrt(x+10) / a+fabs(b)</p> <p>1 y = (sqrt(x)+10) / (a+fabs(b))</p> |
|---|---|---|
- 4 y = sqrt(x)+10 / a+fabs(b)
3 y = sqrt(x+10) / (a+fabs(b))

РЕШЕЊЕ:

Функција sqrt представља кореновање, док функција fabs представља апсолутну вредност.

8. Са десне стране наведене су неке од функција библиотеке `ctype.h`, а са леве су дати њихови описи. Испред назива сваке од наведених функција, уписати редни број под којим је дат одговарајући опис:

- | | |
|--|----------------------------------|
| 1) Да ли је с штампајући знак (укључујући и размак)? | <u>6</u> <code>isspace(c)</code> |
| 2) Да ли је с слово? | <u>2</u> <code>isalpha(c)</code> |
| 3) Да ли је с знак интерпункције? | <u>5</u> <code>isdigit(c)</code> |
| 4) Да ли је с управљачки знак? | <u>7</u> <code>isupper(c)</code> |
| 5) Да ли је с децимална цифра? | <u>1</u> <code>isprint(c)</code> |
| 6) Да ли је с знак бели знак? | <u>4</u> <code>isctrl(c)</code> |
| 7) Да ли је с велико слово? | |
| 8) Да ли је с хекса-децимална цифра? | |

9. Дати су типови променљивих у програмском језику C#. Одредити како се назива променљива која је дефинисана унутар неког метода. Заокружити број испред очекиваног одговора:

- 1) Глобална променљива
- 2) Статичка променљива
- 3) Локална променљива
- 4) Блоковска променљива

РЕШЕЊЕ:

Локална променљива је она која је креирана унутар метода и може се користити само унутар тог истог метода, дакле не постоји ван њега.

10. Одредити какви могу бити чланови класе (поља и методе) у програмском језику C#. Заокружити број испред очекиваног одговора:

- 1) Статички (класни) и нестатички (објектни)
- 2) Локални и глобални
- 3) Процедурални и непроцедурални
- 4) Спољашњи и унутрашњи

РЕШЕЊЕ:

Статичка (класна) поља су она која имају исту вредност за сваки објект и из тог разлога се називају и класна поља јер припадају самој класи а не појединачним објектима. Нестатичка поља су она која се могу разликовати за сваки појединачни објект.

Пример: Сви ученици вашег разреда имају идентичан број разреда (4-7) што би било статичко поље, а ваша појединачна имена, презимена, датуми рођења, јмбг би били нестатичка поља.

Ово решење се односи и на наредна два питања.

11. Одредити која поља су заједничка и јединствена за све креиране објекте неке класе дефинисане у објектно оријентисаном програмском језику C#. Заокружити број испред очекиваног одговора:

- | | | |
|-------------|--------------|-------------|
| 1) Јавна | 3) Статичка | 5) Објектна |
| 2) Приватна | 4) Инстанцна | |

12. У програмском језику C# класа може да садржи статичка и не-статичка (инстанцна) поља. Дате су изјаве које се односе на статичка поља класе и међу њих је уметнута једна изјава која се односи на не-статичка (инстанцна) поља класе. Заокружити број испред изјаве која се односи на не-статичка поља класе:

- 1) Поље које се може се користити без конструисања иједног објекта те класе
- 2) Поље чија се вредност може разликовати за сваки појединачни објект неке класе
- 3) Поље које има исту вредност за све креиране објекте неке класе
- 4) Поље које се може користити унутар статичких метода класе, као и унутар метода инстанце

13. Заокружити број испред исказа који представља исправан наставак дате тврдње: При креирању објеката изведене класе...

- 1) извршава се само конструктор изведене класе
- 2) прво се извршава конструктор родитељске класе, али само ако је позван кључном речју base
- 3) обавезно се прво извршава конструктор родитељске, а потом конструктор изведене класе
- 4) обавезно се прво извршава конструктор изведене, а потом конструктор родитељске класе

14. Дат је код програма у програмском језику C#:

```
namespace TestPrimer {  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine(fun(17));  
        }  
        public int fun(int n) { return n; }  
        public void fun(int n){ Console.WriteLine(n); }  
    }  
}
```

Анализирати код и заокружити број испред очекиваног одговора:

- 1) Програм има грешку, јер је друга верзија преоптерећеног метода fun(...) дефинисана али се нигде не позива.
- 2) Програм има грешку, јер се не може одредити коју верзију преоптерећеног метода fun(...) треба позвати.
- 3) Програм се нормално извршава и приказује 17 једанпут.
- 4) Програм се нормално извршава и приказује 17 двапут.

РЕШЕЊЕ:

Два метода могу имати исто име једино ако имају различит тип или број параметара.

Пример исправног коришћења истог имена две методе:

```
public void fun(int n, int z) { ... }  
public void fun(int n) { ... }
```

15. У програмском језику C# дата је декларација једне стринг и једне целобројне променљиве, као и део кода:

```
string str = "Primer";  
int broj = 66;  
Console.WriteLine(str + broj + 65);  
Console.WriteLine(broj + 65 + str);
```

Анализирати код и проценити шта ће се приказати на екрану након његовог извршења. Заокружити број испред очекиваног одговора:

- 1) Primer6665 131Primer
- 2) Primer6665 6665Primer
- 3) Primer131 131Primer
- 4) PrimerBA BAPrimer

РЕШЕЊЕ:

Уколико се приликом исписа у конзолу на првом месту израза налази string, цео израз у загради ће се гледати као string (текст). Уколико је на првом месту нумеричка вредност вршиће се математичке операције док се не дође до string-а (текста).

16. Дат је код у програмском језику C#, који дефинише рекурзивни метод. Анализирати код и одредити резултат извршавања задатог метода.

```
public int fun(int n) { return n * fun(n - 1); }
```

Заокружити број испред очекиваног одговора:

- 1) Резултат позива fun(3) је 1.
- 2) Резултат позива fun(3) је 2.
- 3) Резултат позива fun(3) је 6.
- 4) Позив fun(3) изазива грешку јер производи бесконачан ланац позива истог метода fun(...).

РЕШЕЊЕ:

С обзиром да нигде не мењамо вредност параметра n функција ће се бесконачно позивати:

3 * fun(2) -> 2 * fun(1) -> 1*fun(0) -> 0*fun(-1) -> -1*fun(-2) ...

17. Дата је дефиниција класе у програмском језику C#. Проценити где у дефиницији класе (испред које методе) треба заменити знакове ??? службеном речју static.

```
public class Test {  
    private int broj;
```

```
    public ??? int getBroj() { return broj; }  
    public ??? int kvadrant(int n) { return n * n; }  
}
```

Заокружити број испред тачне изјаве:

- 1) Метода getBroj мора да буде статичка, док метода kvadrant може и не мора.
- 2) Метода getBroj може и не мора да буде статичка, а метода kvadrant не сме да буде статичка.
- 3) Метода kvadrant може и не мора да буде статичка, а метода getBroj не сме да буде статичка.
- 4) Обе методе не смеју бити статичке.

РЕШЕЊЕ:

Метода која у себи има нестатичко поље (у овом случају то је поље broj – знамо да је нестатичко јер поред њега нема кључне речи static) не сме да буде статичка.

18. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће се приказати на излазу.

```
try  
{ int x = 0;  
  int y = 5 / x; }  
catch (Exception e)  
{ Console.WriteLine("Exception"); }  
catch (ArithmeticException ae)  
{ Console.WriteLine("Arithmetic Exception"); }  
Console.WriteLine("finished");
```

Заокружи број испред тачног одговора:

- 1) Ништа. Дешава се грешка приликом компајлирања
- 2) Приказује се текст: Exception
- 3) Приказује се текст: Arithmetic Exception
- 4) Приказује се текст: finished

РЕШЕЊЕ:

Дешава се грешка приликом компајлирања јер основни блок изузетка (catch (Exception e)) мора да буде на крају уколико имамо више catch блокова.

19. Заокружити број испред исказа који представља исправан наставак дате реченице: Ако try-catch наредба има више catch блокова у којима "хватамо" изузетак основне Exception класе, заједно са изузецима других класа изведених из класе Exceptions...

- 1) онда се изузетак основне Exception класе може „хватати“ у било ком catch блоку (редослед није битан, битно је да се наведу све могуће грешке)
- 2) онда се изузетак основне Exception класе мора „хватати“ у првом catch блоку
- 3) онда се изузетак основне Exception класе мора „хватати“ у последњем catch блоку
- 4) основна Exception класа се не комбинује у истој наредби са класама изведеним из ње јер их основна класа „маскира“

20. Дати су изкази који се односе на правила писања try-catch-finally блокова за руковање изузетцима. Заокружити бројеве испред исказа који су тачни:

- 1) Блок try мора имати бар један catch или један finally блок
- 2) Блок try не сме да има више catch блокова
- 3) Ако блок try има више catch блокова, изузетак основне Exception класе мора се хватати у првом catch блоку
- 4) Ако блок try има више catch блокова, битан је редослед њиховог писања
- 5) Блок try мора имати бар један finally блок
- 6) Блок try може имати више catch блокова

21. Дата је наредба кода у програмском језику C# која представља декларацију низа. Проценити које од доле наведених декларација су тачне. Заокружити бројеве испред очекиваних одговора:

- 1) `int niz = new int(30);`
- 2) `double[] niz = new double[];`
- 3) `int[] niz = { 3, 4, 3, 2 };`
- 4) `char[] niz = new char[30];`
- 5) `char[] niz = new char[] { 'a', 'b', 'c', 'd' };`
- 6) `char[] niz = new char { 'a', 'b' };`

РЕШЕЊЕ:

Први пример је неисправан јер нема угласте заграде ([]) ни са леве ни са десне стране израза. Други пример је неисправан јер су угласте заграде са десне стране израза празне (смеју бити празне једино ако одмах наведемо чланове низа као у петом примеру). Шести пример није исправан јер нема угласте заграде са десне стране израза.

22. Дат је код на C#-у којим су креиране три класе у ланцу наслеђивања. Имајући у виду класификаторе приступа пољима класа, заокружити бројеве испред поља која ће бити видљива унутар класе Sin:

```
public class Deda {  
    private double penzija;  
    protected string adresa;  
    public string ime; }
```

```
public class Otac : Deda {  
    private double plata;  
    protected string struka; }
```

```
public class Sin : Deda {  
    public int razred; }
```

1. penzija
2. adresa
3. ime
4. plata
5. struka
6. razred

РЕШЕЊЕ:

Унутар изведене класе су видљива сва поља те класе као и public и protected поља класе коју наслеђује.

23. У методи Main() класе Program креиран је објекат s класе Sin (Sin s = new Sin();) Заокружити бројеве испред поља која ће бити видљива у креираном објекту s класе Sin:
- ```
public class Deda {
 private string penzija;
 protected string adresa;
 public double ime; }

public class Otac : Deda {
 private double plata;
 protected string firma;
 public string struka; }

public class Sin : Otac {
 private double prosek;
 protected int razred;
 public string skola; }
```
1. penzija
  2. adresa
  3. ime
  4. plata
  5. struka
  6. firma
  7. prosek
  8. razred
  9. skola

**РЕШЕЊЕ:**

Унутар објекта класе су видљива само public поља изведене класе и класа које наслеђује.

24. Дати су искази који у програмском језику C# дефинишу конструктор. Заокружити бројеве испред очекиваних одговора:

- 1) Подразумевани конструктор без параметара се класи аутоматски додаје уколико у њој није експлицитно дефинисан ниједан конструктор.
- 2) Подразумевани конструктор без параметара се увек аутоматски додаје класи.
- 3) У класи се мора експлицитно дефинисати бар један конструктор.
- 4) Конструктори немају тип резултата, чак ни void.

25. У класи Figura дат је подразумевани (default) конструктор и конструктор са 4 параметра:

```
public Figura() {...}
public Figura(string ime, string boja, int pozX, int pozY) {...}
```

Заокружити бројеве испред исправно написаних наредби креирања објекта класе Figura:

- 1) Figura f = new Figura("lovac", "beli", 3);
- 2) Figura f = Figura("lovac", "beli", 7, 3);
- 3) Figura f = new Figura();
- 4) Figura f = new Figura("beli", "lovac", 7, 3);
- 5) Figura f = new Figura("lovac", 3, 7, "beli");
- 6) Figura f = new Figura("lovac", "beli", 3, 7);

**РЕШЕЊЕ:**

Потребно је обратити пажњу на редослед типова параметара (уколико их има), у овом случају string, string, int, int. Други пример је неисправан јер недостаје кључна реч new.

26. Дате су започете изјаве које се односе на делове кода за обраду изузетака. Довршити започете реченице:

Наредбе које могу изазвати грешку стављају се унутар блока

try

Наредбе које се извршавају и ако се деси и ако се не деси грешка стављају се унутар блока

finally

Наредбе које се извршавају у случају настанка грешке стављају се унутар блока

catch

**27.** Са десне стране су наведене области видљивости појединих елемената класе, а са леве стране класификатори приступа којима се врши контрола области видљивости. На линију испред описа области видљивости унети редни број под којим је наведен одговарајући класификатор приступа:

- |              |          |                                                                        |
|--------------|----------|------------------------------------------------------------------------|
| 1. protected | <u>1</u> | видљив унутар класе у којој је дефинисан, као и унутар изведених класа |
| 2. public    | <u>3</u> | видљив само унутар класе у којој је дефинисан                          |
| 3. private   | <u>4</u> | видљив унутар пројекта у коме је дефинисан                             |
| 4. internal  | <u>2</u> | видљив и ван своје класе у којој је дефинисан                          |

**28.** Са леве стране су наведени делови/елементи класе, а са десне стране улоге појединих класних елемената. На линију испред описа улоге унети редни број под којим је наведен одговарајући елеменат класе:

- |                        |          |                                    |
|------------------------|----------|------------------------------------|
| 1. поље (атрибут)      | <u>1</u> | Опис особина објекта класе         |
| 2. деструктор          | <u>5</u> | Контрола приступа пољима класе     |
| 3. конструктор         | <u>4</u> | Опис функционалности објекта класе |
| 4. метод               | <u>3</u> | Креирање објекта класе             |
| 5. својство / property | <u>2</u> | Уништавање објекта класе           |

**29.** Заокружити број испред исказа који тачно дефинише појам веб сајта:

- 1) **скуп веб-страница које могу да садрже текст, слике, видео снимке и други мултимедијални садржај састављен у једну целину**
- 2) скуп протокола за отпремање и преузимање података са интернета, као и протоколи за комуникацију
- 3) веб сајт чини интернет прегледач - програм који се користи за читање хипертекст докумената
- 4) интернет сервис (www) који омогућава корисницима размену докумената која се састоје од текста, слика и мултимедијалних компоненти

**30.** Заокружити број испред тачног одговора. Основни протокол који користи веб сервер је:

- 1) SMTP
- 2) HTML
- 3) **HTTP**
- 4) WWW

**31.** Заокружити број испред тачног одговора. Која је основна улога CSS-а (Cascading Style Sheet):

- 1) Дефинише структуру и садржај странице
- 2) Дефинише којим језиком веб сервер извршава динамичку веб страницу
- 3) Дефинише протокол за приступ одређеној веб страници
- 4) **Дефинише конкретан изглед елемената странице**

**32.** Заокружити број испред тачног одговора. Унутар HTML странице, CSS правила се креирају навођењем селектора елемента, својстава и вредности. Како се дефинишу селектори HTML елемента:

- 1) Селектори се дефинишу искључиво на основу имена HTML елемента
- 2) Селектори се дефинишу на основу атрибута сваког HTML елемента
- 3) **Селектори се дефинишу на основу имена елемента, назива класе или идентификатора елемента**
- 4) Селектори се дефинишу коришћењем кључне речи selector и атрибута



33. Заокружити број испред тачног одговора. Дефинисан је стил елемента у оквиру кога се налази позадинска слика. Коју вредност својство background-repeat треба да има уколико слика не треба да се понавља у оквиру елемента:
- 1) background-repeat: fixed;
  - 2) background-repeat: null;
  - 3) background-repeat: repeat-x;
  - 4) background-repeat: no-repeat;
34. Заокружити број испред тачног одговора. Селектором a:visited омогућава се стилизовање:
- 1) посећеног линка
  - 2) активног линка
  - 3) линка када се преко њега пређе мишем
  - 4) елемента са називом класе: a
35. Заокружити број испред тачног одговора. Које од понуђених својстава омогућава постизање ефекта слојевитости елемената у оквиру HTML стране?
- 1) Својство z-index
  - 2) Својство layer
  - 3) Својство overlap
  - 4) Не постоји решење за постизање слојевитости елемената HTML стране
36. Заокружити број испред тачног одговора. Анализирајте следећи HTML код и одаберите која од понуђених опција описује резултат приказа HTML кода: `<b><i> Šesto pitanje </i></b>`
- 1) Текст ће бити исписан подебљано и искошено
  - 2) Текст ће бити само подебљан
  - 3) Текст ће бити само искошен
  - 4) Неће се применити никакав ефекат на текст
37. Заокружити број испред линије кода којом се у HTML страницу укључује екстерна CSS датотека style.css са циљем дефинисања изгледа елемената дате странице:
- 1) `<style type="text/css">....</style>`
  - 2) `<a href=" style.css ">CSS датотека</a>`
  - 3) `<body style="background-color:red;">...</body>`
  - 4) `<link rel="stylesheet" type="text/css" href="style.css"/>`
38. Заокружити број испред тачног одговора. Који је од наведених начина за укључивање CSS стила најпогоднији за стилизовање, а посебно за касније одржавање и ажурирање HTML странице:
- 1) Дефинисан стил у оквиру елемента HTML стране - Inline style
  - 2) Дефинисан интерни CSS стил у оквиру head секције `<style type="text/css">...</style>` - Internal Style Sheet
  - 3) Увежена екстерна CSS датотека преко хипервезе у заглављу HTML веб странице - External Style Sheet
  - 4) Не постоји решење за укључивање CSS стила у стилизовање HTML странице

39. Заокружити број испред тачног одговора. Дефинисан је HTML елемент:

```
<div id="container"> Deseto pitanje </div>
```

Које CSS правило дефинише стил за дати елемент:

- 1) `#container {  
font-size: 1.5em;  
}`
- 2) `.container {  
font-size: 1.5em;  
}`

- 3) `container {  
font-size: 1.5em;  
}`
- 4) `selector container {  
font-size: 1.5em;  
}`

**РЕШЕЊЕ:**

За означавање идентификатора (ID-ja) користимо предзнак # док за означавање класе користимо предзнак . .

40. Уколико је стил једног DIV елемента дефинисан са три класе blue, black и top, заокружити линију кода којом је елемент исправно дефинисан у оквиру HTML стране:

- 1) `<div id="blue black top">Osmo pitanje</div>`
- 2) `<div id="blue" class="black" class="top">Osmo pitanje</div>`
- 3) `<div class="blue black top">Osmo pitanje</div>`
- 4) `<div class="blue" class="black" class="top">Osmo pitanje</div>`

**РЕШЕЊЕ:**

Потребно је запамтити да кључну реч class можемо користити само једном унутар html елемента, а уколико том елементу желимо да доделимо више класа њихове називе раздвајамо размаком.

41. Заокружити број испред тачног одговора. Дато је својство и додељена му је вредност: background-attachment: fixed;

Закључити који од понуђених исказа описује дефинисани стил елемента са позадинском сликом:

- 1) Позадинска слика се понавља унутар елемента
- 2) Позадинска слика је непомицна (фиксирана) у односу на остатак садржаја
- 3) Позадинска слика се понавља унутар елемента Позадинска слика се помера са остатком садржаја на страни
- 4) Позадинска слика се не понавља у оквиру елемента

42. Заокружити број испред траженог одговора. Дат је следећи код који стилизује три елемента странице различитим позадинским бојама:

```
<div style="background-color:green; z-index:3"> </div>
<div style="background-color:yellow; z-index:1;"> </div>
<div style="background-color:red; z-index:2;"> </div>
```

Проценити шта од понуђеног описује ефекат извршења горњег кода:

- 1) Слој са зеленом позадином је на врху и преклапа остале слојеве
- 2) Слој са жутом позадином је на врху и преклапа остале слојеве
- 3) Слој са црвеном позадином је на врху и преклапа остале слојеве
- 4) Слојеви се утапају у једну боју – нијансу смеђе

**РЕШЕЊЕ:**

Код слојева који се преклапају видљив ће бити онај са највећом вредности z-index.

43. Заокружите тачне исказе. Интернет је:

- 1) Подаци који „путују“ светском мрежом и скуп корисника заједно чине Интернет мрежу
- 2) Софтвер за преглед и приказ www страница се сматра Интернетом
- 3) Интернет је светски систем умрежених рачунарских мрежа
- 4) Интернет чини њена хардверска компонента као и систем софтверских слојева који контролишу различите аспекте њене комуникационе инфраструктуре

44. Међу понуђеним исказима заокружити оне које се односе на динамичке веб странице:

- 1) Могу приступити базама података
- 2) Странице се састоје искључиво од HTML кода
- 3) Динамичке странице се пишу у CSS језику
- 4) Могу слати персонализован садржај појединачним корисницима
- 5) Ажурирање података на сајту је компликованије и спорије него код статичког сајта
- 6) Странице се извршавају на веб серверу, а резултат овог извршавања представља HTML код

45. Заокружити могуће наћине укључивања CSS стила у оквиру HTML стране:

- 1) Увежена екстерна CSS датотека преко хипервезе у заглављу HTML веб странице - External Style Sheet
- 2) Дефинисан стил у оквиру елемента HTML стране - Inline style
- 3) Дефинисан екстерни CSS стил приказивања у прегледачу HTML страница – CSS plugin
- 4) Дефинисан интерни CSS стил у оквиру head секције
- 5) Дефинисан стил на крају HTML стране изван head и body секције – Outline style
- 6) Дефинисан CSS стил у оквиру секције body, унутар тага <css></css>

46. Повезати појмове према захтеву. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страници:

- |                           |                               |
|---------------------------|-------------------------------|
| 1. <font color="#ff0003"> | 3 приказ наслова трећег нивоа |
| 2. <b>                    | 4 приказ текста дате величине |
| 3. <h3>                   | 2 приказ подебљаног текста    |
| 4. <font size=„3">        | 1 приказ текста у боји        |

47. Повезати појмове према захтеву. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страници:

- |                                 |                               |
|---------------------------------|-------------------------------|
| 1. <li>                         | 4 постављање позадинске слике |
| 2. <ol>                         | 3 постављање позадинске боје  |
| 3. <body bgcolor="#ffff00">     | 1 дефинисање елемента листе   |
| 4. <body background="0001.jpg"> | 2 дефинисање нумерисане листе |

48. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страници:

- |                           |                         |
|---------------------------|-------------------------|
| 1. <tr>                   | 4 Дефинисање хиперлинка |
| 2. <td>                   | 2 Нови ћелија у табели  |
| 3.  | 1 Нови ред у табели     |
| 4. <a href="index.html" > | 3 Уметање слике         |

49. На линију испред дефинисане операције написати број њему одговарајућег сервиса.

- |           |          |                                                       |
|-----------|----------|-------------------------------------------------------|
| 1. E-mail | <u>2</u> | успостављање везе са удаљеним рачунаром и рад на њему |
| 2. Telnet | <u>4</u> | приказ HTML веб страница                              |
| 3. FTP    | <u>1</u> | слање електронске поште                               |
| 4. WWW    | <u>3</u> | пренос датотека са удаљеног сервера                   |

50. Направите редослед потребних корака за приказ статичке веб странице уносом редних бројева на линије испред описа корака, почев од броја 1 до броја 5.

- 3 Веб сервер проналази HTML страну коју је корисник захтевао
- 4 Веб сервер шаље пронађену страну клијенту - веб претраживачу.
- 2 Корисник захтева да види веб страну (кликом на линк, укуцавањем адресе у адресну линију веб претраживача и слично)
- 1 Аутор је креирао страну која се састоји од HTML кода и ставио је на веб сервер.
- 5 Веб претраживач обрађује добијени HTML код и приказује кориснику уредно форматирану страницу са свим елементима (сликама, линковима, табелама, ...)

51. Одредите редослед потребних корака за приказ динамичке веб странице уносом редних бројева (почев од 1) на линије испред описа корака.

- 6 Веб претраживач обрађује добијени HTML код и приказује кориснику страницу са свим елементима.
- 2 Корисник креира захтев за преглед динамичке веб стране. Захтев се прослеђује од клијента (веб претраживача) до веб сервера на коме се налази захтевана страна.
- 1 Аутор је креирао страну која се састоји од серверских контрола и инструкција у неком програмском језику и ставио је на веб сервер.
- 4 Веб сервер извршава програмски код који је саставни део стране и креира HTML код.
- 3 Веб сервер обрађује захтев корисника и проналази динамичку страну коју је корисник захтевао.
- 5 Веб сервер шаље преко Интернета генерисани HTML код веб претраживачу.

52. Уредите појмове према захтеву.

Дат је низ HTML наредби. Уписивањем редног броја на предвиђену линију (почев од броја 1 до броја 6) поређајте тачним редоследом HTML наредбе у складу са основном структуром HTML странице:

- 4 <body>
- 2 <head>
- 6 </html>
- 1 <html>
- 3 </head>
- 5 </body>

53. Међу понуђеним ентитетима, одредити ентитет са атрибутима који нису одговарајући.

Заокружити број испред траженог одговора:

- 1) Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек
- 2) Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина
- 3) Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача
- 4) Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта

#### РЕШЕЊЕ:

Нису одговарајући они ентитети од чијих појединих атрибута се може креирати посебан ентитет.

54. Заокружите тачан исказ:

- 1) Примарни кључ је атрибут који мора бити целобројног типа.
- 2) Примарни кључ је атрибут који указује на зависност од неке друге табеле.
- 3) Ако табела садржи више вредносни атрибут, њему се додељује функција примарног кључа.
- 4) Примарни кључ је атрибут који јединствено идентификује врсте у табели.

55. Заокружити тачан исказ:

- 1) Кардиналност неке везе одређује опционалност учешћа у вези.
- 2) Кардиналност неке везе представља апстракцију у којој се скуп сличних типова објеката представља општим генеричким типом (надтипом).
- 3) Кардиналност неке везе представља однос броја објеката који се повезују.
- 4) Кардиналност показује колико кандидата за примарни кључ има неки тип ентитета.

56. Заокружити бројеве испред тражених одговора. Одредити ентитете који садрже одговарајуће атрибуте:

- 1) Ентитет: КЊИГА – Атрибути: наслов, аутор, издавач, година издања
- 2) Ентитет: АУТОМОБИЛ – Атрибути: марка, година производње, боја, власник, година рођења власника
- 3) Ентитет: УЧЕНИК – Атрибути: име, презиме, разред, одељење, број оправданих, број неоправданих, просек
- 4) Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина, главни град, број становника главног града, површина главног града

57. Одредити ентитете који садрже одговарајуће атрибуте. Заокружити бројеве испред тражених одговора:

- 1) Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек
- 2) Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача
- 3) Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта
- 4) Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина
- 5) Ентитет: САЈАМ – Атрибути: назив, број излагача, покровитељ, адреса покровитеља, контакт особа покровитеља
- 6) Ентитет: ТУРИСТИЧКА АГЕНЦИЈА – Атрибути: назив, адреса, година оснивања, власник, стручна квалификација власника, запослени, стручна квалификација запослених

58. Обележити ентитете код којих је извршен адекватан избор јединственог идентификатора. Заокружи бројеве испред тражених одговора:

- 1) јединствени матични број грађанина (ЈМБГ) за ентитет ОСОБА
- 2) датум рођења за ентитет УЧЕНИК
- 3) ИСБН број за ентитет КЊИГА
- 4) режисер за ентитет СЕРИЈА
- 5) учионица за ентитет ШКОЛА
- 6) регистарска ознака за ентитет АУТОМОБИЛ

**59.** Исписати на цртама испред релација редни број под којим је наведена одговарајућа кардиналност везе:

1. 1 : 1

2 УТАКМИЦА – ГРАД ДОМАЋИН

2. 1 : M

3 НАСТАВНИК – ПРЕДМЕТ

3. M : M

1 ОСОБА – ПАСОШ

2 ВЛАСНИК – БРОЈ ТЕЛЕФОНА

3 КУПАЦ – МОДЕЛ АУТОМОБИЛА

**60.** Уписати редни број почев од 1 на линију испред резервисане речи тако да одговара редоследу навођења при формирању упита. За формирање упита за издвајање дела података из табеле која се налази у оквиру базе података користе се клаузуле у следећем редоследу:

3 WHERE

5 ORDER BY

1 SELECT

4 GROUP BY

2 FROM