1. Дата је декларација низа:

```
int k;
int[] brojevi = { 5, 12, 37, 7, 27, 33, 36 };
```

На основу дате декларације одредити шта је резултат позива: `k = Array.BinarySearch(brojevi, 37);`

1) k=0, јер метод BinarySearch прво изврши сортирање у опадајућем редоследу, па тражи задату вредност
2) метод BinarySearch баца изузетак увек када је низ неуређен и програм „пуца”
3) k=2, јер се тражени елемент налази на позицији 2
4) k добија неочекивану вредност јер низ мора бити сортиран у растућем поретку пре позива BinarySearch
5) k=6, јер метод BinarySearch прво изврши сортирање у растућем редоследу па тражи задату вредност

2. Заокружити број испред одговора којим треба комплетирати дати код:

```
public static bool palindrom(String s)
{
    if (s.Length <= 1) return true; //bazni slučaj
    else if (____________________) return false;
    else return palindrom(s.Substring(1, s.Length - 2));
}
```

1) `s[0] != s[s.Length - 1]`
2) `s[0] != s[s.Length]`
3) `s[1] != s.[s.Length - 1]`
4) `s[1] != s[s.length]`

РЕШЕЊЕ:
Уколико је нека реч палиндром значи да се чита исто и са леве и са десне стране. Да би проверили да ли је реч палиндром прво проверавамо да ли су прво (s[0]) и последње (s[s.Length-1] јер је индекс последњег слова за један мањи од укупне дужине речи) слово.

3. Да би код био комплетиран потребно га је допунити.

```
public static bool Palindrom(String s)
{ return Palindrom(s, 0, s.Length - 1); }
public static bool Palindrom(String s, int levi, int desni)
{
    if (desni <= levi)
        return true; // bazni slucaj
    else if (s[levi] != s[desni])
        return false;
    else return _____________________; }
```

Заокружити број испред тачног одговора:

1) Palindrom(s)
2) Palindrom(s, levi, desni)
3) Palindrom(s, levi +1, desni - 1)
4) Palindrom(s, levi + 1, desni)
5) Palindrom(s, levi, desni - 1)

РЕШЕЊЕ:
Након што смо проверили прво и последње слово речи и закључили да су иста проверавамо даље, са леве стране се померамо у десно (+1) а са десне стране се померамо улево (-1) и тако проверавамо друго и претпоследње слово, па треће с лева и треће с десна и тако даље до краја речи.

4. Заокружити број испред одговора којим треба комплетирати дати код:

```
public static int TraziBroj(int[] niz, int broj)
{
    return TraziBroj(niz, broj, 0, niz.Length - 1);
}
public static int TraziBroj(int[] niz, int broj, int levi, int desni)
{
    if (levi > desni) return -1; // broj nije nadjen u nizu
    int sredina = (levi + desni) / 2;
    if (broj < niz[sredina]) return TraziBroj(niz, broj, levi, sredina - 1);
    else if (broj > niz[sredina]) return _____________________________;
    else return sredina;
}
```

1) TraziBroj(niz, broj, sredina + 1, levi)
2) TraziBroj(niz, broj, sredina - 1, levi)
3) TraziBroj(niz, broj, desni, sredina + 1)
4) TraziBroj(niz, broj, sredina + 1, desni)

РЕШЕЊЕ:
С обзиром да се дати код односи на претраживање сортираног низа значи да ако је тражени број већи од броја у средини низа ми треба да претражимо десну половину. То радимо тако што нам лева граница постаје sredina + 1, а десна граница нам је крај низа, односно desni.

5. Заокружити број испред тачно написаног конструктора копије објекта класе Point:

```
public class Point
{
    private double x, y;
    public Point() { x = 0; y = 0; }
    public void Set(double xx, double yy) { x = xx; y = yy; }
    public Point(Point p)
    {
        _____________//Odgovor
    }
}
```

1) this(p.x, p.y); 2) this(p); 3) Set(p); 4) Set(p.x, p.y);

РЕШЕЊЕ:
Потребно је обратити пажњу на број параметара у методи Set (две double вредности). Одговори са this кључном речју немају смисла и могу одмах бити искључени из разматрања.

6. Анализирати дати код и заокружити број испред тачног исказа:

```
class Program
{
    public static void Main(string[] args)
    {
        KlasaA a1 = new KlasaA();
        KlasaA a2 = new KlasaA();
        Console.WriteLine(a1.Equals(a2)); } }

class KlasaA
{
    int x;
    public bool Equals(KlasaA a)
    {
        return this.x == a.x; } }
```

1) Програм има грешку, јер се изразом a1.Equals(a2) проверава једнакост објеката a1 и a2 различитог типа од Object.
2) Програм има грешку јер се једнакост објеката a1 и a2 типа KlasaA проверава изразом a1 == a2.
3) Програм се извршава без грешке и приказује се true на екрану.
4) Програм се извршава без грешке и приказује се false на екрану.

РЕШЕЊЕ:
Разлика између шестог и седмог питања јесте у томе што су у шестом објекти a1 и a2 типа податка KlasaA што значи да ће се приликом позивања методе Equals позвати метода Equals коју смо сами креирали и проверити да ли је вредност поља x иста из чега следи да ће одговор бити true јер нисмо декларисали вредност поља x па оно самим тим има вредност по дифолту.
У седмом задатку тип података објеката је Object што значи да ће се позвати метода Equals из класе Object која проверава да ли су два објекта иста из чега следи да ће одговор бити false.

7. Анализирати дати код и заокружити број испред тачног исказа:

```
class Program
{
    public static void Main(string[] args)
    {
        Object a1 = new KlasaA();
        Object a2 = new KlasaA();
        Console.WriteLine(a1.Equals(a2)); } }

class KlasaA
{
    int x;
    public bool Equals(KlasaA a)
    {
        return this.x == a.x; } }
```

1) Програм има грешку, јер се изразом a1.Equals(a2) проверава једнакост објеката a1 и a2 различитог типа од Object.
2) Програм има грешку, јер се једнакост објеката a1 и a2 типа KlasaA проверава изразом a1 == a2.
3) Програм се извршава без грешке и приказује се true на екрану.
4) Програм се извршава без грешке и приказује се false на екрану.

8. Заокружити бројеве испред кључних речи које омогућавају редефинисање дефинисаног метода кроз ланац наслеђивања:

1) new
2) virtual
3) sealed
4) override
5) abstract
6) base
7) довољно је да буде public или protected

9. Дати су делови кода који треба да рачунају збир елемената матрице int[,] a = new int[10, 10]. Заокружити бројеве испред тачно написаних делова кода:

1)
```
int sum = 0;
for (int i = 0; i < a.Length; i++)
for (int j = 0; j < a[i].Length; j++)
sum += a[i][j];
```

2)
```
int sum = 0;
foreach (int x in a) sum += x;
```

3)
```
int sum = 0;
for (int i = 0; i < a.GetLength(0); i++)
for (int j = 0; j < a.GetLength(1); j++)
sum += a[i, j];
```

4)
```
int sum = 0;
foreach (int[] vrsta in a)
foreach (int el in vrsta)
sum += el;
```

РЕШЕЊЕ:
Проблем у првом одговору је са тим што се тип низа не слаже са нашим – a[i][j] је низ низова, дакле потребно је обратити пажњу да уколико одговор има двоје угластих заграда не слаже се са нашом матрицом која има само један.
Проблем у четвртом одговору је што унутар петље елементе нашег низа означавамо као низове int[] vrsta in a што је грешка јер су чланови нашег низа обични бројеви а не низови.

10. Заокружити бројеве испред тачних исказа:

```
class TestPrimer
{
    public double x;
    public TestPrimer(double x)
    {
        this.fun();
        this.x = x; }
    public TestPrimer()
    {
        Console.WriteLine("Podrazumevani konstruktor");
        this(23); }
    public void fun()
    {
        Console.WriteLine("Poziv metoda fun()"); } }
```

1) this.fun() у конструктору TestPrimer(double x) може се поједноставити и заменити само са fun().
2) this.x у конструктору TestPrimer(double x) може се поједноставити и заменити само са x.
3) позив конструктора this(23) унутар другог конструктора TestPrimer() је прво шта се извршава и мора се писати одмах после декларације public TestPrimer():this(23)
4) this(23) у конструктору Test() мора се заменити са прецизнијим изразом this(23.0).

РЕШЕЊЕ:
Кључну реч this можемо изоставити тамо где постоји само једно поље или метода са тим називом. С обзиром да имамо само једну методу са називом fun можемо израз поједноставити и склонити this.
Поље x је и поље методе и параметар конструктора, из тог разлога нам је потребно да користимо кључну реч this како би разликовали те две променљиве.
Уколико желимо да позовемо други конструктор приликом извршавања једног то радимо на начин из одговора под редним бројем три.

11. Дат је код програма у програмском језику C#. Код садржи објекте две класе у којима је дефинисан метод ToString(). Анализирати код датог програма и одредити који од датих исказа су тачни.

```
class Program
{
    static void Main(string[] args)
    {
        Object a = new Klasa();
        Object obj = new Object();
        Console.WriteLine(a);
        Console.WriteLine(obj); } }


class Klasa
{
    int x;
    public override string ToString() { return "x u A je " + x; }
```

1) Програм има грешку, јер наредбу Console.WriteLine(a) треба заменити наредбом Console.WriteLine(a.ToString()).
2) Приликом извршавања наредбе Console.WriteLine(a), програм позива се метод ToString() наслеђен из класе Object.
3) Приликом извршавања наредбе Console.WriteLine(a), програм позива метод ToString() из класе Klasa.
4) Приликом извршавања наредбе Console.WriteLine(obj), програм позива метод ToString() из класе Object.

РЕШЕЊЕ:
Посматрамо начине на који су креирани објекти, уколико је Object и тип података и конструктор (са обе стране креирања као код објекта obj) онда се позива метода ToString() из класе Object. Уколико користимо конструктор класе Klasa онда се позива override метода ToString(), односно метода из класе Klasa().

12. У програмском језику C# дата је декларација две класе: KlasaA и KlasaB која наслеђује класу KlasaA. Анализирати дате класе и проценити који од понуђених исказа су тачни.

```
class Program
{
    static void Main(string[] args)
    {
        KlasaB b = new KlasaB();
        b.Print();
    }
}
class KlasaA
{
    string s;
    public KlasaA(string s) { this.s = s; }
    public void Print() { Console.WriteLine(this.s); }
}
class KlasaB : KlasaA { } }
```

1) Програм има грешку, јер KlasaB нема подразумевани конструктор KlasaB().
2) Програм има грешку јер KlasaB има подразумевани конструктор, док родитељска KlasaA нема такав конструктор. Програм би радио без грешке уколико би се уклонио конструктор са параметрима из KlasaA.
3) Програм има грешку која се може отклонити уколико би се у KlasaA експлицитно додао конструктор без параметара KlasaA().
4) Програм нема грешку, извршава се, али се на конзоли ништа не исписује јер је поље s добило подразумевану вредност String.Empty

РЕШЕЊЕ:
С обзиром да има више тачних одговора четврти може одмах да се избаци, јер програм не може и да има и да нема грешку.
Након тога се први и други одговор потиру, јер први каже да KlasaB нема подразумевани конструктор, а други да KlasaB има подразумевани конструктор. Подразумевани конструктор се додаје класи уколико сами не направимо неки конструктор, што значи да га KlasaB има.

13. ASP.NET MVC 3.0 долази са новом техником за дефинисање погледа (View Engine). Заокружити назив те технике:

1) ASP.NET View Engine
2) Salome
3) Razor
4) Default

14. Заокружити од понуђених опција који симбол се користи за коментаре у ASP.NET MVC Razor синтакси:

1) //
2) /* ... */
3) <!-- ... -->
4) @* ... *@

15. SOAP протокол (Simple Object Access protocol) који се користи за размену података између рачунара коришћењем веб сервиса, у основи користи један скрипт језик. Заокружити број испред тог језика:

1) JavaScript
2) XML
3) HTML
4) CSS

16. Дата је ASP.NET MVC апликација у којој је креирана нова мастер страница (master layout page) која се зове _Layout.WindowsPhone.cshtml. Ако желимо да укључимо нову мастер страницу на новом погледу (View) који сегмент кода треба да искористимо. Заокружити тачан одговор:

1) @Html.ActionLink("_Layout.WindowsPhone.cshtml");
2) Layout = "~/views/Shared/_layout.WindowsPhone.cshtml";
3) Layout = "Layout.WindowsPhone.cshtml";
4) @Html.Partial("_Layout.WindowsPhone.cshtml");

17. Заокружити тип сервиса који може бити хостован у конзолној или десктоп апликацији:

1) ASMX
2) WCF
3) RESTful
4) XML

18. Заокружити бројеве испред назива скрипт језика за израду динамичких веб страница:

1) PHP 2) jQuery 3) ASP.NET 4) JSP 5) VBScript 6) CSS 7) HTML

19. Обележити шта од наведеног је дефинисано WSDL језиком:

1) Комуникациони интерфејс за веб сервис
2) Начин имплементације метода веб сервиса
3) Списак метода веб сервиса
4) Комуникациони протокол за веб сервис

20. Повратне вредности акције MVC контролера (controller action method) могу бити различитих типова. Заокружити бројеве испред могућих повратних вредности.

1) ViewResult
2) MVCResult
3) ModelResult
4) JsonResult
5) RedirectResult
6) ASPResult

21. Обележити шта од наведеног дефинише XML Schema:

1) Типове података XML елемената и атрибута
2) Вредности XML елемената и атрибута
3) XML елементе који представљају child (деца) елементе
4) Уређење child елемената
5) Начин сортирања атрибута у оквиру елемената
6) Међусобни однос корених (root) елемената документа

22. Дата је MVC стандардна рута (default route) http://localhost/Customer/Details/5 која има 3 сегмента. Препознати на основу дате руте вредности ових сегмената и допунити реченицу: Име контролера (Controller Name) је: Customer, назив методе (Action Method Name) је: Details а ID параметра методе је дат са: 5.

23. Табела RADIONICA садржи следеће колоне: radionica_id, naziv, zanat, lokacija_id. Потребно је да се прикаже колико на свакој локацији има различитих заната. Заокружити број испред упита који даје тражени извештај:

1)
```
SELECT DISTINCT location_id, COUNT(zanat)
FROM radionica
GROUP BY lokacija_id
```

2)
```
SELECT location_id, COUNT(zanat)
FROM radionica
GROUP BY lokacija_id
```

3)
```
SELECT location_id, COUNT(DISTINCT zanat)
FROM radionica
GROUP BY lokacija_id
```

4)
```
SELECT location_id, COUNT(DISTINCT zanat)
FROM radionica
GROUP BY zanat
```

РЕШЕЊЕ:
С обзиром да је задатак да проверимо колико има различитих заната користимо кључну реч distinct која неће дозволити да се исти занат понавља у пребројавању. Резутате групишемо по локацији.

24. Уколико поглед (view) треба користити за измену података у табели, поглед не сме садржати:

1) WHERE клаузулу
2) Спој више табела
3) Алијас колоне
4) GROUP BY клаузулу

25. Међу понуђеним алатима обележити Case алате:

1) Rational Rose
2) Oracle Designer
3) .NET
4) Microsoft Visio
5) Java
6) SQL Express

26. Означити операторе који се не могу користити за поређење са подупитом који враћа више вредности:

1) ALL
2) ANY
3) BETWEEN
4) SOME
5) LIKE
6) IN

РЕШЕЊЕ:
Between се односи на проверу одређеног опсега вредности, од – до. Самим тим потребна нам је само по једна вредност за од и једна за до део опсега.
Like се односи на проверу вредности, нпр. да ли податак почиње са словом А, и не може истовремено проверавати да ли податак почиње са словом А, и словом Б итд.

27. Дата је табела Kupci са структуром:

```
( Id int primary key, Prezime varchar(50), Adresa varchar(50), Mesto varchar(20), Telefon varchar(5),
Status varchar(8) )
```

И табела NoviKupci са структуром:

```
Id int primary key, Prezime varchar(50), Telefon varchar(20), Status varchar(8) )
```

Извршава се упит:

```
INSERT INTO NoviKupci
SELECT * FROM Kupci WHERE Status <> 'Aktivan'
```

Одредити шта је резултат извршења датог упита. Заокружити број испред траженог одговора:

1) Како табела NoviKupci има све колоне које постоје и у табели Kupci, упит се извршава без грешке и у табелу NoviKupci се уписују записи из табеле Kupci са статусом који није Aktivan
2) Упит се не извршава, пријављује грешку јер се број колона у табели Kupci разликује од броја колона у табели NoviKupci
3) Упит би се извршио без грешке да су у SELECT клаузули подупита, уместо * наведене све колоне табеле Kupci које имају своју одговарајућу колону у табели NoviKupci
4) Упит јавља грешку због покушаја уписа вредности у поље примарног кључа који је аутоматски, тј креира га сама база

28. Креиране су и попуњене подацима табеле Korisnik и Prijatelji. Њихова структура и садржај приказани су на слици:

Korisnik

ID	IME	POL
1	Ana	NULL
2	Steva	m
3	Marta	z
4	Petra	z

Prijatelji

Korisnik1	Korisnik2
1	2
1	3
2	3

Извршавањем упита добија се табела са подацима.

```
select k.ime, COUNT(*) as [broj prijatelja]
from Korisnik as k
left join Prijatelji as p on p.korisnik1=k.id or p.korisnik2=k.id
where k.pol='z'
group by k.id, k.ime
```

1) Ana, 1
2) Ana, 2
3) Steva, 1
4) Steva, 2
5) Marta, 1
6) Marta, 2
7) Petra, 0
8) Petra, 1

РЕШЕЊЕ:
С обзиром да се користи left join ка табели Korisnik и да је услов да поље pol има вредност z, подаци који ће се исписати јесу Марта и Петра. Left join означава да ће се из леве табеле сви подаци приказати без обзира да ли се налазе и у десној табели, тако ће Петра бити набројана једном, док ће Марта која се појављује у десној табели двапут бити набројана два пута.

29. Допунити реченицу наводећи назив нормалне форме:
Уколико ни један атрибут релације није вишевредносни, нити композитни, тј. не може се раставити, кажемо да је релација у _____првој_____ нормалној форми.

30. Допунити реченицу наводећи назив нормалне форме:
Уколико сви атрибути релације који нису део кључа зависе од сваког атрибута који је део кључа кажемо да је релација у ___другој____ нормалној форми.

31. Допунити реченицу наводећи назив нормалне форме:
Уколико сви некључни (споредни) атрибути релације не зависе од неког другог некључног атрибута, тј. ако не постоји транзитивна зависност било ког споредног атрибута од било ког кључа те релације, кажемо да је релација у ___трећој_____ нормалној форми.

32. Табела ZAPOSLENI је креирана и попуњена извршавањем следећих наредби:

```
create TABLE zaposleni(
 id INTEGER NOT NULL PRIMARY KEY, rukovodilacId INTEGER, ime VARCHAR(30)
NOT NULL,
 FOREIGN KEY (rukovodilacId) REFERENCES zaposleni(id))
INSERT INTO zaposleni VALUES(1, NULL, 'Petar');
INSERT INTO zaposleni VALUES(2, 1, 'Mihajlo');
INSERT INTO zaposleni VALUES(3, 2, 'Milica');
INSERT INTO zaposleni VALUES(4, 3, 'Lazar');
INSERT INTO zaposleni VALUES(5, NULL, 'Sofija');
```

```
_5_ select * from zaposleni
     where id not in (select distinct rukovodilacId from zaposleni)

_2_ select * from zaposleni
    where id not in
   (select rukovodilacId from zaposleni where rukovodilacId is not null)
_3_ select * from zaposleni where rukovodilacId is null
```

1) Сви радници који су руководиоци неком другом раднику
2) Сви радници који нису руководиоци ником
3) Сви радници који немају надређене руководиоце
4) Сви радници који имају надређеног руководиоца
5) Празна табела

РЕШЕЊЕ:
Празну табелу, односно изостанак било каквих података, добијамо кад год користимо distinct над табелом која има null вредности.