

Попуњава ученик:

Назив школе

Седиште школе

Образовни профил

Име и презиме ученика

Датум одржавања испита

Електротехничар информационих технологија - оглед

МАТУРСКИ ИСПИТ

ИСПИТ ЗА ПРОВЕРУ СТРУЧНО-ТЕОРИЈСКИХ ЗНАЊА

- ТЕСТ -

Попуњава испитна комисија

СКАЛА ЗА ПРЕВОЂЕЊЕ БОДОВА У УСПЕХ

Постигнут број бодова

Успех

до 50

недовољан (1)

50,5 – 63

довољан (2)

63,5 – 75

добар (3)

75,5 – 87

врло добар (4)

87,5 - 100

одличан (5)

ПОСТИГНУТ
БРОЈ БОДОВА

/ 100

ОЦЕНА

_____ ()

Чланови испитне комисије:

1. _____
2. _____
3. _____

Датум прегледа теста: _____

УПУТСТВО ЗА РАД

- Тест који треба да решите има 50 задатака. За рад је предвиђено 120 минута.
- Сваки задатак пажљиво прочитајте, размислите о томе шта се у задатку тражи и пратите упутства о начину на који треба да га решите (заокруживање, повезивање, дописивање и друго).
- Задатке решавајте хемијском оловком. Током рада можете да користите графитну оловку и гумицу. Пре него што предате тест, проверите своје одговоре још једном, а затим оне одговоре који су написани графитном оловком подебљајте хемијском. Одговор који је написан само графитном оловком неће бити признат, као ни одговор који је прецртан или исправљен хемијском оловком. Уколико се у задатку тражи и приказ поступка рада, потребно је да и њега испишете хемијском оловком.
- Важно је да пажљиво одговарате на питања јер **сваки нетачан одговор (део одговора) доноси 0 бодова за задатак у целини**.
- Ако решавање задатака завршите раније, предајте тест и тихо изађите да се не би ометао рад других.

Желимо Вам много успеха у раду!

У следећим задацима заокружите број испред траженог одговора

1. Дата је декларација набројивог типа податка боја :

```
enum boja {crna, plava, zelena, crvena=4, bela=15};
```

Имајући у виду дефиницију набројивог типа податка одредити вредности које имају константе crna, plava, zelena, crvena i bela.

1. crna =0, plava = 1, zelena=2, crvena = 4, bela = 15
2. crna =1, plava = 2, zelena=3, crvena = 4, bela = 15
3. crna =255, plava = 254, zelena=253, crvena = 4, bela = 15

	/	1
--	---	---

2. Једна од функција система за управљање базом података је ауторизација.

Означити шта она регулише:

1. Права надокнаде за коришћење базе података.
2. Права приступа појединих корисника појединим објектима (подацима и другим ресурсима) као и права извршења разних операција над тим објектима.
3. Ауторска права над базом података и програмима који користе ту базу података.

	/	1
--	---	---

3. Одредити шта тачно резервисана реч DISTINCT проузрокује у SELECT наредби:

1. Не приказује NULL вредности.
2. Елиминира све јединствене вредности из табеле.
3. Елиминира све врсте које се више пута јављају у приказу.
4. Елиминира само јединствене врсте у приказу.

	/	1
--	---	---

4. Дата је табела ARTIKLI:

Id	Naziv	DatProiz	JM	Kol	Cena	Akcija
21	Hleb Sava	15.04.2015	kom	100	35	
32	Hleb Dunav	15.04.2015	kom	50	45	0.05
23	Baget beli	12.04.2015	kom	20	32	0.12
47	Baget crni	12.04.2015	kom	12	35	0.08
25	Tost tamni	10.04.2015	kom	6	75	0.15
36	Tost beli	10.04.2015	kom	8	70	
28	Kifla integralna	15.04.2015	kom	15	18	0
10	Brašno Graham	10.03.2015	kg	24	54	0

	/	1
--	---	---

Одредити упит који приказује називе за све артикле и за сваки одредити укупну цену за дату количину:

1. **SELECT Naziv, Kol * Cena as Ukupno FROM ARTIKLI;**
2. SELECT Naziv, count(Kol * Cena) FROM ARTIKLI;
3. SELECT Naziv, COUNT (Kol , Cena) as Ukupno FROM ARTIKLI;
4. SELECT Naziv, Kol + Cena FROM ARTIKLI;

5. Изабрати исказ о подупитима који је тачан:

1. Подупит се пише под наводницима.
2. Подупит не сме да садржи групне функције.
3. Подупит се често користи у WHERE клаузули да врати вредности које испуњавају одређени услов.
4. Подупит се најчешће извршава последњи, након што се изврши главни (спољни) упит.

	/	1
--	---	---

6. Одредити оператор поређења који се може користити у упиту који има подупит који враћа више врста:

1. IN, ANY и ALL
2. LIKE
3. BETWEEN... AND ...
4. =, < и >

	/	1
--	---	---

7. У програмском језику Ц дат је код програма у коме заглавље **main** има следећи изглед: **void main(int argc, char *argv[])**. Програм PRIMER.EXE, који има овакво заглавље main функције, стартовати из командне линије наредбом:

C:\TC\BIN>PRIMER A:\primer1.txt A:\primer2.txt

Одредити које су вредности улазних аргумената argc и argv функције main након стартовања програма PRIMER.EXE.

1. argc = 3 argv[0] = PRIMER argv[1] = A:\primer1.txt argv[2] = A:\primer2.txt
2. argc = 2 argv[0] = A:\primer1.txt argv[1] = A:\primer2.txt
3. argc = 2 argv[0] = primer1.txt argv[1] = primer2.txt
4. argc = 3 argv[0] = PRIMER argv[1] = primer1.txt argv[2] = primer2.txt

	/	1
--	---	---

8. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

	/	2
--	---	---

**SELECT imeod, avg(plata) AS ProsekPlata FROM Radnik, Odeljenje
WHERE Odeljenje.brod=Radnik.brod GROUP BY imeod HAVING avg(plata)>1000;**

Одредити резултат извршавања датог упита:

1. Групишу по одељењима радници са платом већом од просечне плате
2. Приказују називи одељења и висина просечне плате у њима само за одељења у којима је просечна плата већа од 1000
3. Приказују називи свих одељења и висина просечне плате у њима
4. Приказују бројеви свих одељења и висина просечне плате у њима

9. Дата је табела **RADNIK**, табела **ODELJENJE**:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	3900	10
5932	Mitar	Gavrilović	2600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	
6789	Janko	Nikolić	3900	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

	/	2
--	---	---

Одредити ком типу упита припада упит којим се радници из одељења **Plasman** прераспоређују у одељење **Nabavka**:

1. Угњеждени упит
2. **Акциони упит**
3. Параметарски упит
4. Агрегатни упит

10. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	3900	10
5932	Mitar	Gavrilović	2600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	
6789	Janko	Nikolić	3900	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

	/	2
--	---	---

SELECT odeljenje.imeod, radnik.ime
from odeljenje left join radnik on radnik.brod = odeljenje.brod ;

Одредити шта се види као резултат датог упита. Приказ чине:

1. Називи само одељења у којима има радника са именима радника у њима
2. **Називи одељења у којима има радника са именима радника у њима и називи одељења у којима нема радника**
3. Називи свих одељења и имена радника у њима
4. Називи само одељења у којима нема радника

11. Заокружити број испред упита који задовољава захтеве задатка.

Табела **ZAPOSLENI** има следећа поља (**zaposleni_id**, **ime**, **prezime**, **plata**, **odsek_id**). Потребно је повећати плату запосленима у ИТ сектору 12.5%. ИТ сектор има као шифру одсека – 15. Изабрати коју наредбу је потребно употребити да би се урадила тражена измена у табели:

	/	2
--	---	---

1. UPDATE zaposleni(plata)
SET plata=plata*1,125
2. UPDATE zaposleni
SET plata=plata*0.125
WHERE odsek_id=15
3. **UPDATE zaposleni
SET plata=plata*1.125
WHERE odsek_id=15**
4. UPDATE zaposleni (plata)
VALUES (plata*12.5)
WHERE odsek_id=15

12. Наведени су искази који се односе на дефиницију `do while` циклуса. Који од ових исказа су **тачни**:

1. Користи се када се незна 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 завршава када услов добија вредност логичке истине
4. Циклус се извршава барем једном.

	/	1
--	---	---

13. Наредбама програмског језика Ц, дата је декларација једне симболичке константе и једне константне променљиве:

```
#define k 50; ...
int m=100; ...
```

Одредити исправно написане наредбе декларације дводимензионалног низа целих бројева (матрице):

1. `int a [k][k];`
2. `int b [k][m];`
3. `int c [k][10];`
4. `int x [100][50];`
5. `int y [10, 10];`
6. `int z [m][10];`

	/	1,5
--	---	-----

14. 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 објеката класе `Figura`:

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

	/	1,5
--	---	-----

15. 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

	/	1,5
--	---	-----

16. Мајкрософт .NET Framework садржи базне класе које пружају широк спектар могућности.

Заокружити све елементе које су укључене у њу:

1. класе корисничког интерфејса
2. класе за приступ подацима и базама
3. класе корисника
4. веб сервер и примере базе података
5. класе за манипулацију XML докумената
6. Скрипт језик који се извршава на клијент страни

	/	1,5
--	---	-----

17. За упите са специфицираним редоследом приказа врста у резултујућој табели користи се клаузула ORDER BY после које се наводи назив колоне:

1. и службена реч ASCENDING за растући редослед
2. и службена реч DESCENDING за опадајући редослед
3. и службена реч ASCENDING за опадајући редослед
4. и службена реч DESCENDING за растући редослед
5. службена реч се може изоставити, добија се растући поредак
6. службена реч се може изоставити, добија се опадајући поредак

	/	1,5
--	---	-----

18. Дати су прототипови функција написани у програмском језику Ц. Одредити исправно написане прототипове функција:

1. int pp3(int a[], n; float b);
2. void pp4(int *a, int n);
3. int pp5(int a[][], int n);
4. int pp6(int a[], int n);
5. int pp7(int a, b, c);
6. float* pp1(int a, int b, int c);
7. int pp2(int a[][10], int n);
8. float[] pp8(float a[], int n);

	/	2
--	---	---

19. У програмском језику C је декларисана низовна променљива:

```
int niz[10];
```

Одредити исправно написане наредбе читања низа целих бројева дужине 10 из бинарног фајла на који показује показивач **in*:

1. `fread(niz, sizeof (int), 10, in);`
2. `fread(&niz, sizeof (int), 10, in);`
3. `fread(&niz, sizeof niz, 1, in);`
4. `fread(niz, sizeof niz, 1, in);`
5. `fread(niz, sizeof (niz), 1, *in);`
6. `fread(niz, sizeof (int)*10, in);`

	/	2
--	---	---

20. Који искази у програмском језику C# дефинишу конструктор:

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

	/	2
--	---	---

21. Дат је код програма у програмском језику C#. Код садржи објекте две класе у којима је дефинисан метод **toString()**.

```
namespace TestPrimer {  
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;}  
}
```

	/	2
--	---	---

Који од датих исказа су тачни:

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

22. Дат је код програма у програмском језику C# који декларише два објекта типа **Pravougaonik** и један објекат типа **Krug**. Класе **Pravougaonik** и **Krug** наслеђују апстрактну класу **Figura** и обе класе имају методу **ToString**. Код програма треба допунити са наредбом која ће исписивати низ **Figura[]** декларисан у коду програма.

	/	2
--	---	---

```
1. public abstract class Figura {
2.     protected static int brojFigura=0;
3.     public static int BrojFigura {get{return brojFigura;}}
4.     public Figura() {brojFigura++; }
5.     public abstract double površinaFigura();
6.     public override string ToString(){return "Figura ";}
7. }
...
1. Pravougaonik pr1 = new Pravougaonik();
2. Pravougaonik pr2 = new Pravougaonik(3, 2);
3. Krug kr1 = new Krug(2);
4.
5. Console.WriteLine("Broj figura="+Figura.BrojFigura+"\n");
6. Figura[] niz = new Figura[Figura.BrojFigura];
7. niz[0] = pr1; niz[1] = pr2; niz[2] = kr1;
8.
9. Console.WriteLine("Prikaz figura:");
10. _____
```

Заокружити бројеве испред одговора који представљају могуће начине приказа низа фигура:

1. **for(int i=0; i<niz.Length; i++) Console.WriteLine(niz[i]);**
2. **foreach (el in niz) Console.WriteLine(el);**
3. **foreach(Figura el in niz) Console.WriteLine(el);**
4. **foreach (int i in niz) Console.WriteLine(niz[i]);**

23. Од понуђених појмова шта дефинише језик WSDL (Web Services Description Language):

	/	2
--	---	---

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

24. Одредити ентитете који садрже одговарајуће атрибуте:

	/	2
--	---	---

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

25. Дата је табела **RADNIK**, табела **ODELJENJE**:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	3900	10
5932	Mitar	Gavrilović	2600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	
6789	Janko	Nikolić	3900	30

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

	/	2
--	---	---

Дат је упит:

UPDATE Radnik SET radnik.brod = 50

WHERE radnik.brod = (Select brod from odeljenje where odeljenje.imeod = 'Plasman');

Одредити којим типовима упита припада дати упит:

1. Параметарски упит
2. Агрегатни упит
3. Угњеждени упит
4. Акциони упит

26. Дат је код програма у програмском језику C# и састоји се од две класе у једној датотеци. Анализирати дати код и проценити његову тачност:

```
namespace TestPrimer {  
    class Program {  
        static void Main(string[] args){  
            A a = new A();  
            a.Print();  
        }  
    }  
    class A {  
        string s;  
        public A(string s) { this.s = s; }  
        public void Print(){  
            Console.WriteLine(s);  
        }  
    }  
}
```

	/	3
--	---	---

1. Програм има грешку, јер класа A нема подразумевани конструктор.
2. Програм нема грешака и нормално се извршава ништа не приказујући на екрану.
3. Програм има грешку, јер класа A није јавна класа.
4. Програм има грешку која се може исправити уколико се у четвртом реду наредба **A a = new A();** замени наредбом **A a = new A("poruka");**.

27. У програмском језику Ц написати функцију **xcalloc** која додељује меморијски простор вектору од **n** елемената и врши иницијализацију његових елемената. Сваки елемент вектора заузима меморијски простор величине **size**. У седму линију кода дописати наредбу која недостаје да би функција извршавала свој задатак.

	/	3
--	---	---

```
1. void *xcalloc(int n, int siz )
2. {
3.     int i;
4.     char *p,*q;
5.
6.     if((p = q = malloc(n*size))!= NULL){
7.         _____
8.     }
9.     return q;
10. }
```

Одредити наредбе које су могуће решење овог проблема:

1. for(i=0; i<n*size; i++) *p=0;
2. for(i=0; i<n*size; *p++=0, i++);
3. for(i=0; i<n*size; i++) *p++=0;
4. for(i=0; i<size; i++) p++=0;

28. Дата је дефиниција класе у програмском језику С# и састоји се од два конструктора, једне методе и поља x. У дефиницији се користи службена реч **this**. Анализирати дати код и проценити тачност следећих исказа:

	/	3
--	---	---

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

29. Дат је прототип функције написан у програмском језику Ц:

```
void Umetni(char *a, char k);
```

У main функцији дате су следеће декларације променљивих:

```
char s1[20], *s2, s3;
```

Одредити који су исправно написани позиви декларисане функције:

1. `Umetni(s2, s1[i]);`
2. `Umetni(s2, s1);`
3. `Umetni(s2, 'A');`
4. `Umetni(s1, s3);`
5. `Umetni(*s2, s3);`
6. `Umetni(s3, &s1);`

	/	3
--	---	---

Допуните следеће реченице и табеле

30. Дата је табела **RADNIK**, табела **ODELJENJE**:

IDBR	IME	PREZIME	PLATA	BROD	BROD	Naziv	Mesto
5900	Slobodan	Golubović	3900	10	50	Skladišta	Zemun
5932	Mitar	Gavrilović	2600	10	30	Marketing	Vračar
5953	Persida	Kosanović	1100	20	10	Plasman	Surčin
6234	Marko	Pavlović	1300		20	Direkcija	Grocka
6789	Janko	Nikolić	3900	30	40	Nabavka	Barajevo

	/	1
--	---	---

Извршава се упит:

```
UPDATE Radnik SET radnik.brod = 50
```

```
WHERE radnik.brod = ( Select brod from odeljenje where odeljenje.imeod = 'Plasman');
```

Допуните реченицу:

Извршењем упита ажурирају се подаци у записима за 2 радника.

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

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

catch

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

finally

Наредбе које могу изазвати грешку стављају се унутар блока

try

	/	1,5
--	---	-----

32. Дата је MVC стандардна рута (default route)

`http://localhost/Customер/Details/5`

која има 3 сегмента.

На основу дате руте вредности ових сегмената допуните реченицу:

Име контролера (Controller Name) је: Customer, назив методе (Action Method Name) је: Details а ID параметра методе је дат са: 5.

	/	1,5
--	---	-----

33. Дата је наредба форматираног уноса вредности у променљиве, написана на програмском језику Ц: `scanf("%o%c%x%e", &x, &y, &z, &w);`
За сваку променљиву, на основу употребљене конверзије, одредити ког је она типа и уписати на одговарајућу линију:

1. променљива **x** је типа: int
2. променљива **y** је типа: char
3. променљива **z** је типа: int
4. променљива **w** је типа: float

	/	2
--	---	---

34. На програмском језику Ц, декларисани су и иницијализовани стрингови:

```
char s1 []="crvena zvezda";  
char s2 []="zelengora";
```

Одредити и на предвиђену линију уписати садржај стрингова по извршењу следеће наредбе:

```
strncpy(s1, s2, 3);
```

s1 = "zelena zvezda"

s2 = "zelengora"

	/	2
--	---	---

35. Допунити реченицу:

Тригери (окидачи) се могу стартовати непосредно pre или posle извршавања SQL наредбе.

	/	2
--	---	---

36. Дате су следеће декларације: `int p[200], i, n, pom;`
 А затим и део кода који треба да врши циклично померање елемената низа `p`, дужине `n`, за једно место удесно. У коду недостају неки од елемената. Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

	/	3
--	---	---

```

pom = p[n-1];
for(i=n-2; i >=0; i--)
    p[i+1] = p[i];
p[0] = pom;
  
```

1. `p[0]`
2. `p[n-1]`
3. `p[n]`
4. `p[i+1]`
5. `p[i-1]`
6. `p[i]`
7. `i++`
8. `i--`
9. `>=0`
10. `>0`

37. Дат је део кода написан на програмском језику Ц:
- ```

int a[7]={81,12,35,97,40,52,17}, *pa, x, y;
pa=a+3;
x = *(pa-2)+1;
y = (*pa-2)+1;

```

|  |   |   |
|--|---|---|
|  | / | 4 |
|--|---|---|

Анализирати код и одредити вредности променљивих `x` и `y`, као и показивача `pa`, по извршењу све три извршне наредбе датог кода:

```

x = 13
y = 96
pa = a + 3

```

### У следећим задацима сажето написати одговор

38. Написати линију кода којом се укључује екстерна CSS датотека (`style.css`) у оквиру заглавља веб странице - `index.html` (датотеке `style.css` и `index.html` се налазе у истом директоријуму):

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

`<link rel="stylesheet" type="text/css" href="style.css" />`

### У следећим задацима повежите и уредите појмове према захтеву

39. Са леве стране наведене су врсте грешака које се могу десити у коду, а са десне описи тих грешака. На линију испред описа уписати редни број под којим је наведен одговарајући тип грешке:

|  |   |     |
|--|---|-----|
|  | / | 1,5 |
|--|---|-----|

- |                                             |          |                                                                                 |
|---------------------------------------------|----------|---------------------------------------------------------------------------------|
| 1. Синтаксне грешке (syntax error)          | <u>1</u> | Откривају се у фази компилације и најлакше су за кориговање                     |
| 2. Логичке грешке (logical error)           | <u>3</u> | Не морају, али се могу десити током извршења и тада доводе до „пуцања“ програма |
| 3. Грешке у време извршења (run-time error) | <u>2</u> | Програм функционише, нема „пуцања“, али се добијају погрешни резултати          |

40. Са леве стране су наведене су области видљивости појединих елемената класе, а са десне стране класификатори приступа којима се врши контрола области видљивости. На линију испред класификатора приступа унети редни број под којим је наведена одговарајућа област видљивости:

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

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

41. Декларисана је реална променљива `float w=123.456;`  
Са леве стране дати су различити прикази вредности променљиве добијени коришћењем наредби форматираног излаза које су приказане са десне стране. Поред сваке наредбе, на предвиђену линију уписати редни број приказа добијеног извршавањем те наредбе:

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

- |                  |          |                                 |
|------------------|----------|---------------------------------|
| 1. 123.456000    | <u>3</u> | <code>printf("%g", w);</code>   |
| 2. 1.234560e+002 | <u>1</u> | <code>printf("%f", w);</code>   |
| 3. 123.456       | <u>4</u> | <code>printf("%.2f", w);</code> |
| 4. 123.46        | <u>2</u> | <code>printf("%e", w);</code>   |

42. Са леве стране су наведене наредбе позиционирања у датотеци, а са десне описи ефекта датих наредби. На линију поред наредбе уписати редни број под којим је наведен опис ефекта наредбе:

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

- |                                      |          |                                                                          |
|--------------------------------------|----------|--------------------------------------------------------------------------|
| <code>ftell(dat)</code>              | <u>3</u> | 1. позиционирање на почетак датотеке                                     |
| <code>fseek(dat, 0, SEEK_END)</code> | <u>2</u> | 2. позиционирање на крај датотеке                                        |
| <code>fseek(dat, 0, SEEK_SET)</code> | <u>1</u> | 3. одређује позицију у датотеци у виду броја бајтова од почетка датотеке |
| <code>rewind(dat)</code>             | <u>1</u> | 4. ништа од понуђеног                                                    |

43. Са леве стране су дати HTML тагови, а са десне стране су наведени индикатори.

На линију испред уписати број, повезати наведене HTML тагове са индикаторима које ће се видети на HTML страници:

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

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

44. Са леве стране су дати HTML тагови, а са десне стране су наведени индикатори.

На линију испред уписати број, повезати наведене HTML тагове са индикаторима које ће се видети на HTML страници:

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

|  |   |   |
|--|---|---|
|  | / | 2 |
|--|---|---|

45. Са леве стране дате су врсте конверзије, а са десне типови података који се користе у функцији printf() у програмском језику Ц. На линију испред типа података унети број којим је означена одговарајућа конверзија:

- |        |          |                                                      |
|--------|----------|------------------------------------------------------|
| 1. %d  | <u>7</u> | Short                                                |
| 2. %i  | <u>1</u> | Signed int (u dekadnom obliku)                       |
| 3. %s  | <u>4</u> | Long                                                 |
| 4. %ld | <u>8</u> | Unsigned                                             |
| 5. %f  | <u>2</u> | Signed int (dekadni, heksadekadni ili oktalni oblik) |
| 6. %e  |          |                                                      |
| 7. %hd |          |                                                      |
| 8. %u  |          |                                                      |

|  |   |     |
|--|---|-----|
|  | / | 2,5 |
|--|---|-----|

46. Повезати резервисану реч са редним бројем који одговара појму на који се односи у терминологији база података..

- |                |          |             |
|----------------|----------|-------------|
| 1. Табела      | <u>3</u> | VIEW        |
| 2. Страни кључ | <u>5</u> | CONSTRAINT  |
| 3. Поглед      | <u>1</u> | TABLE       |
| 4. Окидач      | <u>2</u> | FOREIGN KEY |
| 5. Ограничење  | <u>4</u> | TRIGGER     |

|  |   |     |
|--|---|-----|
|  | / | 2,5 |
|--|---|-----|

47. Са десне стране дате су врсте конверзије, а са леве типови података који се користе у функцији за приказ података printf() у програмском језику Ц. На линију испред типа конверзије унети број којим је означен одговарајући тип података.

- |           |          |     |
|-----------|----------|-----|
| 1. char   | <u>3</u> | %f  |
| 2. sting  | <u>2</u> | %s  |
| 3. float  | <u>3</u> | %g  |
| 4. double | <u>4</u> | %lf |
| 5. long   | <u>3</u> | %e  |
| 6. short  | <u>1</u> | %c  |

|  |   |   |
|--|---|---|
|  | / | 3 |
|--|---|---|



48. Написати на цртама испред логичких операција редне бројеве њихових приоритета:

1. највиши приоритет 3 OR  
2. средњи приоритет 1 NOT  
3. најнижи приоритет 2 AND

|  |   |   |
|--|---|---|
|  | / | 3 |
|--|---|---|

49. Дата је табела **RADNIK** и табела **ODELJENJE**:

| IDBR | IME      | PREZIME    | PLATA | BROD |
|------|----------|------------|-------|------|
| 5900 | Slobodan | Golubović  | 3900  | 10   |
| 5932 | Mitar    | Gavrilović | 2600  | 10   |
| 5953 | Persida  | Kosanović  | 1100  | 20   |
| 6234 | Marko    | Pavlović   | 1300  |      |
| 6789 | Janko    | Nikolić    | 3900  | 30   |

| BROD | Naziv     | Mesto    |
|------|-----------|----------|
| 50   | Skladišta | Zemun    |
| 30   | Marketing | Vračar   |
| 10   | Plasman   | Surčin   |
| 20   | Direkcija | Grocka   |
| 40   | Nabavka   | Barajevo |

|  |   |   |
|--|---|---|
|  | / | 3 |
|--|---|---|

Повезати упите и њихова значења уписом броја упита на одговарајућу линију:

- |                                                                                                                                                       |          |                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------------------------------------------------------------------------------------------------------------|
| 1. SELECT odeljenje.imeod,<br>radnik.prezime<br>from odeljenje inner join radnik on<br>radnik.brod = odeljenje.brod ;                                 | <u>5</u> | Приказује само раднике који<br>нису распоређени у одељења                                                      |
| 2. SELECT odeljenje.imeod,<br>radnik.prezime<br>from odeljenje left join radnik on<br>radnik.brod = odeljenje.brod ;                                  | <u>3</u> | Приказује све раднике, и оне који<br>нису распоређени у одељења,<br>и само она одељења у којима<br>има радника |
| 3. SELECT odeljenje.imeod,<br>radnik.prezime<br>from odeljenje right join radnik on<br>radnik.brod = odeljenje.brod ;                                 | <u>1</u> | Приказује само одељења у<br>којима има радника и само<br>раднике распоређене у<br>одељењима                    |
| 4. SELECT odeljenje.imeod,<br>radnik.prezime<br>from odeljenje left join radnik on<br>radnik.brod = odeljenje.brod<br>where radnik.brod is null ;     | <u>2</u> | Приказује сва одељења, и она у<br>којима нема радника и само<br>оне раднике који су<br>распоређени у одељења   |
| 5. SELECT odeljenje.imeod,<br>radnik.prezime<br>from odeljenje right join radnik on<br>radnik.brod = odeljenje.brod<br>where odeljenje.brod is null ; | <u>4</u> | Приказује само одељења у<br>којима нема радника                                                                |

50. Направите редослед потребних корака за приказ једне статичке веб странице. Унесите редни број на линије испред описа корака.

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

|  |   |   |
|--|---|---|
|  | / | 4 |
|--|---|---|