

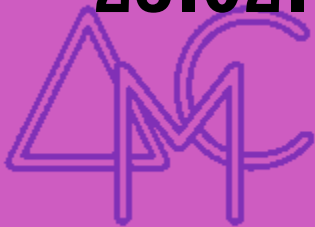
**ОПШТИНСКО  
ТАКМИЧЕЊЕ ИЗ  
РАЧУНАРСТВА ЗА  
УЧЕНИКЕ ОСНОВНИХ  
ШКОЛА**

**БИЛТЕН**

-----  
**ДРЖАВНА КОМИСИЈА  
ЗА ТАКМИЧЕЊЕ  
УЧЕНИКА ОСНОВНИХ  
ШКОЛА**

2017/18

**25.02. 2018.**



ДМС

## Садржај:

- **Допис организаторима**
- **Допустиви програмски језици и  
развојна окружења**
  - **Задаци и решења**
- **Анализа најчешћих грешака**



# ДРУШТВО МАТЕМАТИЧАРА СРБИЈЕ

**Кнез Михаилова 35/IV**  
**11000 Београд**  
**Србија**

## ОРГАНИЗАТОРИМА ОПШТИНСКОГ ТАКМИЧЕЊА ИЗ ИНФОРМАТИКЕ

### ЗА УЧЕНИКЕ ОСНОВНИХ ШКОЛА

Поштоване колегинице и колеге!

Шаљемо вам формулацију задатака из информатике за општинско такмичење ученика основних школа. Такмичење се одржава у недељу, **25. 02. 2018.** године, са почетком у **10 часова**. Израда задатака траје **120 минута**, рачунато од тренутка када су задаци подељени ученицима. Молимо вас да поштујете сатницу такмичења. Молимо вас да лице, које је одговорно за регуларност и реализацију такмичења, уз нужне мере предострожности, умножи задатке у потребном броју. Аутори задатака су одредили да се са по 5 поена вреднује сваки тест пример. Молимо комисију која прегледа ученичке радове да се придржава тих критеријума.

Извештај са резултатима такмичења, потребно је доставити Окружној комисији за такмичење. Поред имена ученика и броја освојених поена, нека спискови садрже и име наставника, назив школе и места, како би се формирао јединствен списак колега наставника који су важни учесници у спровођењу такмичења. Препоручујемо да резултате доставите Државној комисији на емаил [takmicenjeinf@gmail.com](mailto:takmicenjeinf@gmail.com) на формулару који можете превући са сајта <http://www.dms.rs/DMS/html/takmicenja/informatika/osnovne.html>

Молимо Вас да нам никако не шаљете скениране табеле или табеле које су различите од табела са препорученог формулара како би Окружна комисија могла што пре да обједини резултате.

На Веб страници <http://www.dms.rs/DMS/html/takmicenja/informatika/osnovne.html> пратите и упутство везано за подршку прегледању резултата такмичења путем платформе <https://petlja.org/> која је креирана уз помоћ Мајкрософт развојног центра Србија (МДЦС). Надамо се да сте креирали налоге за рад на овој платформи (потребно Вам је два минута да то урадите) !!!

Препоручујемо такмичарима да задатке решавају у форми конзолних програма. Свим такмичарима желимо успешан пласман, а комисији и организаторима такмичења желимо пријатан радни дан!

С поштовањем,  
Председник комисије за такмичења  
Јелена Хаџи-Пурић

**Општинско такмичење ученика основних школа из рачунарства - пети разред  
(25.02. 2018.)**

**1. Стефанов брат је две године старији од њега, док је сестра три године млађа од њега. Напишите програм STEFAN који на основу Стефанових година израчунава збир година њих троје.**

Улаз: Излаз:

7      20

**Тест примери (за бодовање)**

УЛАЗ	ИЗЛАЗ
8	23
12	35
4	11
22	65
30	89

**Анализа:**

Након уноса Стефанових година, потребно је израчунати колико година има Стефанов брат и колико година има Стефанова сестра. На крају се исписује збир Стефанових, братових и сестриних година.

**Програмски језик C**

```
#include <stdio.h>

int main()
{
    int stefan, brat, sestra;
    scanf("%d", &stefan);
    brat = stefan + 2;
    sestra = stefan - 3;
    printf("%d\n", stefan + brat + sestra);
    return 0;
}
```

**Програмски језик Python**

```
stefan=int(input())
brat=stefan+2
sestra=stefan-3
zbir=stefan + brat + sestra
print(zbir)
```

## Програмски језик Java

```
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner ulaz = new Scanner(System.in);

        int stefan = ulaz.nextInt();

        int brat = stefan + 2;
        int sestra = stefan - 3;

        System.out.println(stefan + brat + sestra);
    }
}
```

## Програмски језик Pascal

```
program zad1;
var stefan, brat, sestra:integer;
begin
    read(stefan);
    brat := stefan + 2;
    sestra := stefan - 3;
    writeln(stefan+brat+sestra);
end.
```

## Програмски језик C#

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int stefan, brat, sestra;
        stefan = int.Parse(Console.ReadLine());
        brat = stefan + 2;
        sestra = stefan - 3;
        Console.WriteLine(stefan + brat + sestra);
    }
}
```

### Најчешће грешке:

Такмичари који нису освојили поене на овом задатку су превидели чињеницу да је Стефанова сестра три године млађа од Стефана, а не од Стефановог брата. Неки од такмичара су штампали само Стефанов број година, док се у задатку прецизно захтевало да се штампа укупан број година за Стефана, његовог брата и његову сестру.

**2. Ленкин тата ради између 14 часова и поноћи (посао напушта тачно када откуца 00:00). Напиши програм LENKA који учитава тренутно време (у једном реду број сати између 14 и 23, а у наредном реду број минута између 0 и 59) и исписује колико сати и минута је остало до поноћи.**

Улаз:   Излаз:   Улаз:   Излаз:   Улаз:   Излаз:   Улаз:   Излаз:  
17    6 48    16    7 1    17    7 0    23    0 1  
12           59           0           59

### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
15 0	9 0
23 42	0 18
23 23	0 37
22 02	1 58
22 58	1 2

### Анализа:

Након уноса тренутног времена, наредна поноћ се може представити у облику времена 24:00. Један начин да се уради овај задатак је да се од броја 24:00 одузме учитани број сати и минута, тако што се примени поступак одузимања бројева записаних у бројевној основи 60. Када је број минута у умањенику мањи од броја минута у умањиоцу, потребно је да се изврше позајмице са претходне позиције. Тада се позајмљује један сат и претвара у 60 минута. Ако је број минута у умањиоцу једнак нула, онда се мора извршити враћање позајмице, тако што се резултујући број сати увећа за 1, а резултујући број минута постави на нула.

## Програмски језик C++

```
#include <iostream>
using namespace std;
int main()
{
    int sat, minut, preostaliSati, preostaliMinuti;
    cin >> sat >> minut;
    preostaliSati=24-sat-1;
    preostaliMinuti=60-minut;
    if(preostaliMinuti==60)
    {
        preostaliSati++;
        preostaliMinuti=0;
    }

    cout << preostaliSati<< " " << preostaliMinuti << endl;
    return 0;
}
```

## Програмски језик C

```
#include <stdio.h>
int main()
{
    int sat, minut, preostaliSati, preostaliMinuti;
    scanf("%d%d", &sat, &minut);
    preostaliSati=24-sat-1;
    preostaliMinuti=60-minut;
    if(preostaliMinuti==60)
    {
        preostaliSati++;
        preostaliMinuti=0;
    }
    printf ("%d %d", preostaliSati, preostaliMinuti);
    return 0;
}
```

## Програмски језик Python

```
sati=int(input())
minuti=int(input())
#preostalo vreme u minutima
ostalo = (60 - minuti) + (24 - sati - 1) * 60
print ('%d %d' % (ostalo // 60, ostalo % 60))
```

## Програмски језик Java

```
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner ulaz = new Scanner(System.in);
        int sat = ulaz.nextInt();
        int minut = ulaz.nextInt();

        int preostaliSati, preostaliMinuti;
        preostaliSati=24-sat-1;
        preostaliMinuti=60-minut;

        if(preostaliMinuti==60)
        {
            preostaliSati++;
            preostaliMinuti=0;
        }

        System.out.println(preostaliSati + " " +
preostaliMinuti);
    }
}
```

## Програмски језик Pascal

```
program zad2;
var sat, minut, preostaliSati, preostaliMinuti:integer;
begin
    readln(sat); readln(minut);
    preostaliSati:=24-sat-1;
    preostaliMinuti:=60-minut;
    if(preostaliMinuti=60) then
```



```

begin
    preostaliSati:=preostaliSati+1;
    preostaliMinuti:=0;
end;
writeln(preostaliSati, ' ', preostaliMinuti);
end.

```

## Програмски језик C#

```

using System;

class Program
{
    static void Main(string[] args)
    {
        int sati, minuti, ostalo;
        sati = int.Parse(Console.ReadLine());
        minuti = int.Parse(Console.ReadLine());
        //preostalo vreme do ponoci u minutima
        ostalo = (60 - minuti) + (24 - sati - 1) * 60;
        //ispis preostalih sati i preostalih minuta do ponoci
        Console.WriteLine(ostalo/60 + " " + ostalo%60);
    }
}

```

### Најчешће грешке:

Поједини такмичари су рачунали преостале часове до поноћи тако што од 23 часа одузму текући сат, али су забуном лоше прерачунали преостале минуте када је текући број минута једнак 0. Та идеја је могла да донесе поене на свим тест примерима, изузев првог тест примера.

**3. Написати програм AVION који са улаза читава у прва два реда времена полетања авиона (у првом реду број сати узмеђу 0 и 23, у другом реду број минута између 0 и 59), а у следећа два реда време слетања авиона (у трећем реду број сати узмеђу 0 и 23, у четвртом реду број минута између 0 и 59) и испишује трајање лета у часовима и минутима. Претпоставити да су полетање и слетање у истом дану, као и да су све вредности исправно унете.**

Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:
17	0 59	16	1 3	17	3 18	23	0 58
12		59		0		01	
18		18		20		23	
11		02		18		59	

### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
17 12 18 10	0 58
16 59 18 01	1 2
14 0 20 18	6 18
12 02 20 00	7 58
23 01 23 57	0 56

#### Анализа:

##### 1. начин

Израчунамо колико је секунди протекло од поноћи до почетка полетања и колико секунди је протекло од поноћи до почетка полетања. Одузмемо та два броја и добијемо трајање лета у секундама. Потом испишемо трајање лета у целим сатима и целим минутима.

##### 2. начин

Израчунамо колико је минута протекло од поноћи до почетка полетања и колико минута је протекло од поноћи до почетка полетања. Одузмемо та два броја и добијемо трајање лета у секундама. Потом испишемо трајање лета у целим сатима и целим минутима.

## Програмски језик C++

```
#include <iostream>
using namespace std;
int main()
{
    int poletanjeSat, poletanjeMinut, sletanjeSat,
sletanjeMinut, poletanje, sletanje, trajanjeLeta;
    cin >> poletanjeSat >> poletanjeMinut;
    cin >> sletanjeSat >> sletanjeMinut;
    //pretvoriti vreme poletanja u sekunde
    poletanje=poletanjeSat*3600+poletanjeMinut*60;
    //pretvoriti vreme sletanja u sekunde
    sletanje=sletanjeSat*3600+sletanjeMinut*60;
//izracunati trajanje leta u sekundama
    trajanjeLeta=sletanje-poletanje;
    //izdvajamo broj sati i broj minuta trajanja leta
    int brojSati = trajanjeLeta/3600;
    int brojMinuta = (trajanjeLeta %3600)/60;
    cout << brojSati << " " << brojMinuta << endl;
    return 0;
}
```

## Програмски језик C

```
#include <stdio.h>
int main()
{
    int poletanjeSat, poletanjeMinut, sletanjeSat,
sletanjeMinut, poletanje, sletanje, trajanjeLeta;
    scanf("%d%d", &poletanjeSat, &poletanjeMinut);
    scanf("%d%d", &sletanjeSat, &sletanjeMinut);
    //pretvoriti vreme poletanja u sekunde
    poletanje=poletanjeSat*3600+poletanjeMinut*60;
    //pretvoriti vreme sletanja u sekunde
    sletanje=sletanjeSat*3600+sletanjeMinut*60;
//izracunati trajanje leta u sekundama
    trajanjeLeta=sletanje-poletanje;
    //izdvajamo broj sati i broj minuta trajanja leta
    int brojSati = trajanjeLeta/3600;
    int brojMinuta = (trajanjeLeta %3600)/60;
    printf("%d %d\n", brojSati, brojMinuta);
    return 0;
}
```

## Програмски језик Python

```
poletanjeSat=int(input())
poletanjeMinut=int(input())
sletanjeSat=int(input())
sletanjeMinut=int(input())
minutiPoletanja=poletanjeSat*60+poletanjeMinut
minutiSletanja=sletanjeSat*60+sletanjeMinut
#II nacin - izracunati trajanje leta u minutama
trajanjeLeta=minutiSletanja-minutiPoletanja

sati=trajanjeLeta//60
minuti=trajanjeLeta %60
print(sati, minuti)
```

## Програмски језик Java

```
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner ulaz = new Scanner(System.in);

        int poletanjeSat = ulaz.nextInt();
        int poletanjeMinut = ulaz.nextInt();

        int sletanjeSat = ulaz.nextInt();
        int sletanjeMinut = ulaz.nextInt();

        int poletanjeUMinutama = (poletanjeSat * 60) +
poletanjeMinut;
        int sletanjeUMinutama = (sletanjeSat * 60) +
sletanjeMinut;

        int trajanjeLeta = sletanjeUMinutama -
poletanjeUMinutama;

        System.out.println(trajanjeLeta/60 + " " +
trajanjeLeta%60);
    }
}
```

## Програмски језик Pascal

```
program zad3;
var poletanjeSat, poletanjeMinut, sletanjeSat,
sletanjeMinut:integer;
    minutiPoletanja, minutiSletanja, trajanjeLeta, sati,
minuti :integer;
begin
    readln(poletanjeSat); readln(poletanjeMinut);
    readln(sletanjeSat); readln(sletanjeMinut);
    minutiPoletanja:=poletanjeSat*60+poletanjeMinut;
    minutiSletanja:=sletanjeSat*60+sletanjeMinut;
    trajanjeLeta:=minutiSletanja-minutiPoletanja;
    sati:=trajanjeLeta div 60;
    minuti:=trajanjeLeta mod 60;
    writeln(sati, ' ', minuti);
end.
```

## Програмски језик C#

```
using System;

class Program
{
    static void Main(string[] args)
    {
        int poletanjeSat = int.Parse(Console.ReadLine());
        int poletanjeMinut = int.Parse(Console.ReadLine());

        int sletanjeSat = int.Parse(Console.ReadLine());
        int sletanjeMinut = int.Parse(Console.ReadLine());

        int poletanjeUMinutama = (poletanjeSat * 60) +
poletanjeMinut;
        int sletanjeUMinutama = (sletanjeSat * 60) +
sletanjeMinut;

        int trajanjeLeta = sletanjeUMinutama -
poletanjeUMinutama;

        //ispis trajanje leta u celobrojnim satima i minutima
        Console.WriteLine(trajanjeLeta/60 + " " +
trajanjeLeta%60);
    }
}
```

## Најчешће грешке:

Поједини такмичари су грешили при рачунање у бројевном систему са основом 60. Наиме, трајање лета израчунава се тако што се од времена доласка одузме време поласка. Задатак, дакле, захтева израчунавање разлике између два временска тренутка за која знамо да су у оквиру једног дана. Како су оба временска тренутка записана у бројевној основи 60, могуће је применити алгоритам одузимања бројева датих у позиционом запису у датој основи.

Одузимање креће од позиција најмање тежине (у овом случају то су минути), при чему се врши позајмица са претходне позиције ако је то потребно (ако је број на датој позицији у умањеном већи од броја на датој позицији у умањенику). Позајмице је могуће вршити током одузимања. Али, могуће је позајмице извршити и одмах на почетку, у фази претварања оба временска тренутка у минуте, а тек онда кренути са одузимањем (које се онда врши у бројевној основи 10, јер више нема потребе за позајмицама).

**4. Напиши програм МАКSTROCIFREN који учитава троцифрен број, а на стандардни излаз исписује највећи троцифрен број који се састоји од истих цифара као и број на улазу.**

Улаз: Излаз:  
123 321

Улаз: Излаз:  
174 741

Улаз: Излаз:  
505 550

## Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
150	510
243	432
228	822
989	998
709	970

## Анализа:

За учитани број, израчунамо цифре стотина, десетица и јединица коришћењем количника и остатка при целобројном дељењу са 100 и са 10. Потом нађемо цифру која је највећа по вредности. Затим нађемо цифру која је најмања по вредности. Преостала цифра је средња по вредности. Потом формирамо број код кога највећа цифра представља цифру стотина, а најмања цифра представља цифру јединица. Испишемо формиран број.

## Програмски језик C++

```
#include <iostream>
#include <algorithm>
using namespace std;

int main()
{
    int broj, stotine, desetice, jedinice, najvecaCifra,
    najmanjaCifra, sredina;
    cin >> broj;
    stotine=broj/100;
    desetice=(broj%100)/10;
    jedinice=broj%10;
    najvecaCifra= max(max(stotine,desetice), jedinice);
    najmanjaCifra= min(min(stotine,desetice), jedinice);
    sredina= (stotine+desetice+jedinice) -
    (najvecaCifra+najmanjaCifra);
    cout << 100*najvecaCifra+10*sredina+najmanjaCifra;

    return 0;
}
```

## Програмски језик Python

```
broj=int(input())
stotine=broj//100
desetice=(broj%100)//10
jedinice=broj%10
najvecaCifra= max(stotine,desetice, jedinice)
najmanjaCifra= min(stotine,desetice, jedinice)
sredina= (stotine+desetice+jedinice) -
(najvecaCifra+najmanjaCifra)
print(100*najvecaCifra+10*sredina+najmanjaCifra)
```

### Најчешће грешке:

Поједини такмичари су тачно одредили највећу и најмању цифру, али су погрешно претпоставили да цифра која је средња по вредности је строго мања од најмање од највеће цифре и строго већа од најмање цифре. Видимо по тест примерима да такав програм не даје коректан резултат када број садржи више појава најмање или највеће цифре.

Неки такмичари нису успели да коректно одреде максимум три цифре.

Неки такмичари су погрешили при одређивању минимума три цифре.

## Општинско такмичење 2018. - шести разред

1. Напишите програм RAZLIKA, који учитава са улаза троцифрени број и израчунава разлику између највећег и најмањег троцифреног броја, који се записује истим цифрама као и учитани број.

Улаз: Излаз:  
123 198

Улаз: Излаз:  
174 594

Улаз: Излаз:  
505 45

### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
150	405
245	297
228	594
989	99
100	0

### Програмски језик C++

```
#include <iostream>
using namespace std;

int main()
{
    int n,x,y,z;
    cin >> n;        // unos trocifrenog broja

    x= n/100;        // stotine
    y = n/10%10;     // desetice
    z = n%10;        // jedinice

    if(x>y) swap(x,y); // sortiranje
    if(y>z) swap(y,z); // tri cifre
    if(x>y) swap(x,y); // z, y, x u nerastucem poretku

    int najveciBroj = z*100 + y*10 + x; // najveci trocifren
    broj zyx

    if(x==0) swap(x,y); // ako je broj sa jednom cifrom 0
    if(x==0) swap(x,z); // ako je broj sa 2 cifre nula
```



```

    int najmanjiBroj = x*100 + y*10 + z; // najmanji trocifren
    broj xyz

    cout << najvećiBroj - najmanjiBroj << endl; // razlika

    return 0;
}

```

**2. Наставница информатике жели да подели бомбоне својим ученицима. Напиши програм SLATKISI који учитава број бомбона које наставница има и број ученика и исписује колико најмање бомбона треба да купи да би сваки ученик добио исти број бомбона.**

```

Улаз:  Излаз:      Улаз:  Излаз:
234   6           100   0
4           25

```

#### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
235 35	10
100 10	0
2 10	8
989 50	11
100 99	98

#### Програмски језик C++

```

#include <iostream>
using namespace std;
int main() {
    int broj_bombona, broj_ucenika;
    cin >> broj_bombona >> broj_ucenika;
}

```

```

    cout << (broj_ucenika - broj_bombona % broj_ucenika) %
    broj_ucenika << endl;
    return 0;
}

```

**3. Напиши програм SEDMIDAN који за унети датум одређује који датум ће бити кроз 7 дана. Са стандардног улаза се уносе три позитивна цела броја (сваки у засебном реду) који представљају дан, месец и годину једног исправног датума након 31.12.1969. На стандардни излаз исписати три цела броја која представљају дан, месец и годину датума који ће наступити кроз 7 дана. Сви бројеви се исписују у једном реду, а иза сваког броја наводи се тачка.**

Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:
25	4.3.2018.	25	3.3.2016.	31	7.1.2018.
02		02		12	
2018		2016		2017	

#### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
1 1 2018	8.1.2018.
30 12 2017	6.1.2018.
29 7 2018	5.8.2018.
22 02 2018	1.3.2018.
24 02 2000	2.3.2000.

#### Анализа:

Да бисмо одредили дан након протекле седмисе, прво ћемо увећати број дана за 7. Потом је потребно одредити тачан број дана у унетом месецу (уз проверу и да ли је година преступна). Одређивање броја дана у месецу је пример вишеструког гранања у ком

вршимо провере да ли унети месец има тачно 31 дан или 30 дана или је унети месец фебруар.

И, ако се након увећања дана добије непостојећи датум тј. ако увећани дан прелази број дана у том месецу, тада се прелази на одговарајући број дана наредног месеца (тако што се месец увећа за један, а број дана умањи за највећи број дана учитаног месеца). Али, ако је учитани датум након 25. децембра, онда се увећањем броја месеца добије месец који је већи од 12. У том случају се прелази на одговарајући дан у јануару наредне године (тако што се месец постави на један, а година увећа за један).

## Програмски језик C++

```
#include <iostream>
using namespace std;
int main()
{
    int d,m,g; //dan, mesec, godina unetog datuma
    cin>>d>>m>>g;
    d=d+7; //uvecanje broja dana za 7
    //ako mesec ima tacno 31 dan
    if
    ((m==1)|| (m==3)|| (m==5)|| (m==7)|| (m==8)|| (m==10)|| (m==12))
        if (d>31) //narednih 7 dana je u sledecem
mesecu
        {
            d=d-31;
            m=m+1;
        }
        if ((m==4)|| (m==6)|| (m==9)|| (m==11)) //ako mesec ima
tacno 30 dana
        if (d>30) //narednih 7 dana je u sledecem mesecu
        {
            d=d-30;
            m=m+1;
        }
    if (m==2) //ako je mesec februar
        if (((g%4)==0)&&((g%100)!=0)) ||((g%400)==0)) //aoko je
godina prestupna
        {
            if (d>29) ////narednih 7 dana je u sledecem mesecu
            {
                d=d-29;
                m=m+1;
            }
        }
    }
```

```

else {

    if (d>28) //narednih 7 dana je u sledecem mesecu

    {

        d=d-28;
        m=m+1;
    }

    }

    if (m>12) //narednih 7 dana je u sledecoj godini

    {

        m=1;
        g=g+1;
    }

    cout<<d<<"."<<m<<"."<<g<<"."<<endl;
    return 0;
}

```

#### Најчешће грешке:

Осим решења у ком се мењају вредности променљивих које репрезентују данашњи дан на горе описани начин, такмичари су користили и решење у ком се користе посебне променљиве којима се представља будући датум након седам дана. Та идеја јесте коректна, али су поједини такмичари изостављали копирање вредности које се поклапају са унетим датумом (нпр. месец унетог датума и/или година унетог датума).

**4. Пера и Ана раде у истој програмерској фирми. Ако се зна радно време сваког од њих напиши програм INTERVAL који одређује колико су времена провели на послу заједно. Са улаза се учитава време када је Пера дошао на посао, време када је отишао, време када је Ана дошла на посао и време када је она отишла. Свако време задаје се у посебној линији, преко два броја одвојених једним размаком. Резултат исписати у облику броја сати и минута, опет раздвојених једним размаком.**

Улаз: Излаз:

8 50 5 5

16 40

9 20

14 25

#### Анализа:

Пресек два временска интервала. Први временски интервал представља Перин боравак на послу, а други временски интервал представља Анин боравак на послу.

Почетак евентуалног пресека је већи од два времена доласка на посао тј.  
 $a_{\text{presek}} = \max(\text{ПераДолозакНаПосао}, \text{АнаДолозакНаПосао})$

Крај евентуалног пресека је мањи од од два времена одласка са посла тј.

$b_{\text{presek}} = \min(\text{ПераОдлазакСаПосла}, \text{АнаОдлазакСаПосла})$ .

Пресек постоји ако и само ако је  $b_{\text{presek}} > a_{\text{presek}}$ .

### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
9 20 14 25 8 50 16 40	5 5
9 00 14 00 13 59 20 00	0 1
08 01 12 00 12 01 17 00	0 0
08 21 21 21 08 22 19 22	11 0
06 05 12 04 12 00 18 01	0 4

### Најчешће грешке:

Грешке у одређивању пресека два временска интервала. Неки ученици су уместо пресека два временска интервала заправо одређивали покривач два временска интервала и на тај начин уместо проблема пресека скупова решавали проблем уније скупова.

### Програмски језик C++

```
#include <iostream>
using namespace std;
int main ()
{
    int peralsat, peralminut, pera2sat, pera2minut, analsat,
    analminut, ana2sat, ana2minut;
```

```

int
peradolazak,peraodlazak,anadolazak,anaodlazak,dolazak,odlazak;
cin >>peralsat>>peralminut; //Pera, dolazak na posao
cin >>pera2sat>>pera2minut; //Pera, odlazak sa posla
cin >>analsat>>analminut;
cin >>ana2sat>>ana2minut;
peradolazak=peralsat*60+peralminut;
peraodlazak=pera2sat*60+pera2minut;
anadolazak=analsat*60+analminut;
anaodlazak=ana2sat*60+ana2minut;
dolazak=max(peradolazak, anadolazak);
odlazak=min(peraodlazak, anaodlazak);

if (dolazak>odlazak) cout<<0<<" "<<0;
else cout<<(odlazak-dolazak)/60<<" "<<(odlazak-dolazak)%60;
return 0;
}

```

## Програмски језик Python

```

perals, peralm = map (int, input().split()) #pera, pocetak
radnog vremena
pera2s, pera2m = map (int, input().split()) #pera, kraj radnog
vremena

anals, analm = map (int, input().split()) #ana, pocetak radnog
vremena
ana2s, ana2m = map (int, input().split()) #ana, kraj radnog
vremena

pera_dolazak = perals*60+peralm #pera, pocetak radnog vremena
(minuti u odnosu na ponoc)
pera_odlazak = pera2s*60+pera2m #pera, kraj radnog vremena
(minuti u odnosu na ponoc)
ana_dolazak = anals*60+analm
ana_odlazak = ana2s*60+ana2m

pocetak = max(pera_dolazak, ana_dolazak) #pocetak eventualnog
preseka dva intervala
kraj = min(pera_odlazak, ana_odlazak) #kraj eventualnog preseka
dva intervala

if pocetak > kraj:
    zajedno = 0
else:

```

```
zajedno = kraj - pocetak

presek_sati = zajedno // 60
presek_minuta = zajedno % 60

print (presek_sati,presek_minuta)
```

## Општинско такмичење 2018. - седми и осми разред

**1. Наставница информатике жели да подели бомбоне својим ученицима. Напиши програм SLATKISI који учитава број бомбона које наставница има и број ученика и исписује колико најмање бомбона треба да купи да би сваки ученик добио исти број бомбона.**

Улаз:	Излаз:	Улаз:	Излаз:
234	6	100	0
24		25	

Решење: Погледати задатак за 6. разред

**2. Напишите програм RAZLIKA, који учитава са улаза троцифрени број и израчунава разлику између највећег и најмањег троцифреног броја, који се записује истим цифрама као и читани број.**

Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:
123	198	174	594	505	45

Решење: Погледати задатак за 6. разред

**3. Пера и Ана раде у истој програмерској фирми. Ако се зна радно време сваког од њих напиши програм који одређује колико су времена провели на послу заједно. Са улаза се учитава време када је Пера дошао на посао, време када је отишао, време када је Ана дошла на посао и време када је она отишла. Свако време задаје се у посебној линији, преко два броја одвојених једним размаком. Резултат исписати у облику броја сати и минута, опет раздвојених једним размаком.**

Улаз:	Излаз:
8 50	5 5
16 40	
9 20	
14 25	

Решење: Погледати задатак за 6. разред

**4. Напишите програм VIKEND који ће израчунати колико има викенд дана између дата два датума. Викенд је свака субота или недеља која није пре првог датума, нити након другог датума. У прва два реда стандардног улаза су задата два датума у формату дд.мм.гггг (дан.месец.година). У једином реду стандардног излаза исписати укупан број викенд дана између дата два датума. Ако је први датум након другог, исписати 0. Оба датума су између 1.01.1900 (понедељак) и 31.12.2100 (петак) укључујући и та два датума.**



Пример 1

Улаз  
23.02.2018  
1.3.2020

Излаз  
2

Пример 2

Улаз  
02.03.2018  
30.1.2018

Излаз  
3

Пример 3

Улаз  
23.02.2020  
03.02.2018

Излаз  
1

### Тест примери (за бодовање)

УЛАЗ	ИЗЛАЗ
1.01.1900 31.12.2100	20974
29.02.1904 29.02.2096	20036
8.8.2008 11.11.2100	9628
29.02.2000 15.02.2002	204
1.1.2018 31.1.2018	8

### Програмски језик C++

```
#include <iostream>
using namespace std;
//vraca 0 ako su datumi isti, -1 ako je prvi datum pre drugog,
1 ako je prvi datum nakon drugog
int uporediDatume(int prviDan, int prviMesec, int prviGodina,
    int drugiDan, int drugiMesec, int drugiGodina)
{
    if (prviGodina < drugiGodina)
        return -1;
    else if (prviGodina > drugiGodina)
        return 1;
    else if (prviMesec < drugiMesec)
        return -1;
    else if (prviMesec > drugiMesec)
        return 1;
    else if (prviDan < drugiDan)
        return -1;
    else if (prviDan > drugiDan)
```

```

        return 1;
    else
        return 0;
}
//racuna tacan broj dana za dati mesec i datu godinu
int izracunajDaneUmeseću(int godina, int mesec)
{
    if (mesec == 1 || mesec == 3 || mesec == 5 || mesec == 7 ||
        mesec == 8 || mesec == 10 || mesec == 12)
        return 31;

    if (mesec == 4 || mesec == 6 || mesec == 9 || mesec == 11)
        return 30;

    // ako je mesec==2 tj. februar
    if ((godina % 4 == 0 && godina % 100 != 0) || (godina % 400
== 0))
        return 29; // prestupna godina
    else
        return 28;
}

int main()
{
    int pocetakDan, pocetakMesec, pocetakGodina; //dd, mm, gg za
pocetak intervala
    cin >> pocetakDan;
    cin.get(); // preskoci '.' sa ulaza
    cin >> pocetakMesec;
    cin.get(); // preskoci '.' sa ulaza
    cin >> pocetakGodina;

    int zavrsetakDan, zavrsetakMesec, zavrsetakGodina; //dd, mm,
gg za zavrsetak intervala
    cin >> zavrsetakDan;
    cin.get(); // preskoci '.' sa ulaza
    cin >> zavrsetakMesec ;
    cin.get(); // preskoci '.' sa ulaza
    cin >> zavrsetakGodina ;

    int brojacVikenda = 0;

    int dan = 1; // dan u opsegu [1..31]
    int mesec = 1; // mesec u opsegu [1..12]
    int godina = 1900; // opseg [1900..2100]

```

```

    int danSedmice = 0; // 1.01.1900 je ponedeljak, tj. 0-ti dan
    sedmice
    while (uporediDatume(dan, mesec, godina, zavrsetakDan,
        zavrsetakMesec, zavrsetakGodina) <= 0)
    {
        if (uporediDatume(dan, mesec, godina, pocetakDan,
            pocetakMesec, pocetakGodina) >= 0)
        {
            // provera da li danSedmice je subota ili nedelja
            if (danSedmice == 5 || danSedmice == 6)
                brojacVikenda++;
        }

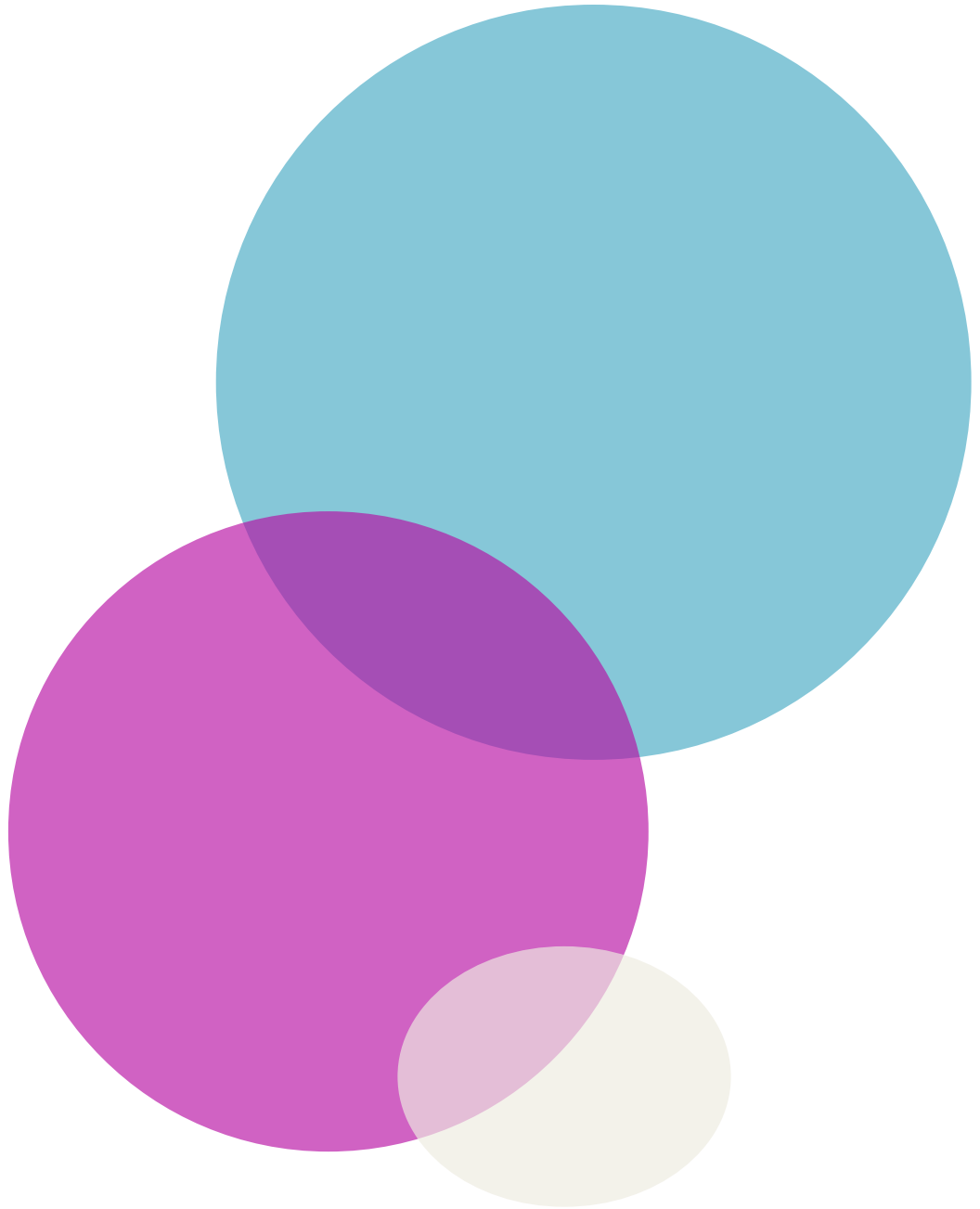
        // prelaz na sledeci dan, npr. 31.12.2017 -> 1.1.2018
        dan++;
        if (dan > izracunajDaneUmesecu(godina, mesec))
        {
            mesec++;
            dan = 1;
        }
        if (mesec > 12)
        {
            godina++;
            mesec = 1;
        }

        // prelaz na sledeci dan sedmice
        //npr. subota -> nedelja, nedelja -> ponedeljak
        danSedmice = (danSedmice + 1) % 7;
    }

    cout << brojacVikenda << endl;

    return 0;
}

```



## Препоручена развојна окружења за такмичења из информатике за основне школе за школску годину 2017/18

### Програмско окружење

За решавање задатака, такмичари могу да користе програмске језике C, C++, Pascal, C#, Python, Java. Организатор такмичења обезбеђују такмичарима развојне алате из табеле 1, а додатно могу обезбедити и поједине развојне алате из табеле 2. Уколико постоје проблеми да се обезбеде сви развојни алати из табеле 1, онда је довољно сваком појединачном такмичару обезбедити развојни алат из табеле 1 за који се претходно определио.

**Табела 1. Обавезно расположиви развојни алати**

Скраћени назив	Опис развојног алата (могу се користити наведене или новије верзије)
Dev-C++	Dev-C++ 5.11 са бар GCC 4.8.1
Code::Blocks	Code::Blocks 13.12 са GCC 4.8.1
Free Pascal	Free Pascal 2.6 са припадајућим Free Pascal IDE
Visual C#	Microsoft Visual Studio 2010 са подршком за C#
Java	Eclipse 4.6 (са Java 1.8, JDT plugins)
Python	IDLE (Python's Integrated Development and Learning Environment) са бар Python 3.6.1

**Табела 2. Опционо расположиви развојни алати**

Скраћени назив	Опис развојног алата (могу се користити наведене или новије верзије)
Visual Studio Code 1.11	Visual Studio Code 1.11 (cpptools, vscode-javac extension)

Ако организатори такмичења нису у могућности обезбедити наведене верзије званичних компјлера и окружења, могу се користити неке од приближних верзија које су у потпуности компатибилне са званичним верзијама. Приликом комисијског тестирања задатака, изворни кодови ће бити тестирани искључиво са препорученим развојним окружењима.

За све развојне алате из табеле 1 и 2 је могуће преузети бесплатне верзије са интернета (табела 3). Подаци у табели 3 су информативног карактера и подложни су променама, па се може догодити да у тренутку када ово читате неки податак из табеле 3. више није ажуран.

**Табела 3. Веб стране преко којих је могуће преузети бесплатну верзију развојног алата**

Скраћени назив алата	Веб страна
Code::Blocks	<a href="http://www.codeblocks.org/">http://www.codeblocks.org/</a>
Dev-C++	<a href="http://orwelldevcpp.blogspot.com/">http://orwelldevcpp.blogspot.com/</a>
Free Pascal	<a href="http://www.freepascal.org/">http://www.freepascal.org/</a>
Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>
MinGW (GCC) компјлер:	<a href="http://www.mingw.org/">http://www.mingw.org/</a>
Eclipse	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
IDLE	<a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a>

Такмичари који улажу напор да уђу у састав јуниорске репрезентације, морају да буду свесни да је на свим међународним програмерским такмичењима и турнирима пријављеним за јуниорску категорију у 2018. години предвиђено коришћење програмских језика C и C++ и развојних алата Code::Blocks и Dev-C++, а да је само на неким такмичењима предвиђено и коришћење програмског језика Java. На пријављеним међународним програмерским такмичењима задаци се решавају у форми конзолних апликација.