

**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.

С++

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

int main() {
    int stefan, brat, sestra;
    cin >> stefan;
    brat = stefan + 2;
    sestra = stefan - 3;
    cout << stefan + brat + sestra << endl;
    return 0;
}
```

С

```
#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;
}
```

Python

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

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);
    }
}
```

Pascal

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

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 умањеоцу, потребно је да се изврше позајмице са претходне позиције. Тада се позајмљује један сат и претвара у 60 минута. Ако је број минута у умањеоцу једнак нула, онда се мора извршити враћање позајмице, тако што се резултујући број сати увећа за 1, а резултујући број минута постави на нула.

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;
}

```

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;
}

```

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))

```

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);
    }
}

```

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.

```

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 sat, 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 учитава 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 исписује 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 слетања. 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 слетања. Одузмемо та два броја и добијемо трајање лета у секундама. Потом испишемо трајање лета у целим сатима и целим минутима.

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;
}
```

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;
}
```

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)
```

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);
    }
}
```

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.
```



```

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.

C++

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

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

    return 0;
}
```

Python

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

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

Тест примери (за бодовање)

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

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 najvećiBroj = z*100 + y*10 + x; // najveći 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
```


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).

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 развојених 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.

$apresek = \max(\text{ПераДолазакНаПосао}, \text{АнаДолазакНаПосао})$

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

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

Пресек постоји ако и само ако је $bpresek > apresek$.

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.

Python

```
pera1s, pera1m = map(int, input().split()) #pera, pocetak radnog vremena  
pera2s, pera2m = map(int, input().split()) #pera, kraj radnog vremena
```

```
ana1s, ana1m = map(int, input().split()) #ana, pocetak radnog vremena  
ana2s, ana2m = map(int, input().split()) #ana, kraj radnog vremena
```

```
pera_dolazak = pera1s*60+pera1m #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 = ana1s*60+ana1m  
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)
```

C++

```
#include <iostream>  
using namespace std;  
int main ()  
{  
    int pera1sat, pera1minut, pera2sat, pera2minut, ana1sat, ana1minut, ana2sat, ana2minut;  
    int peradolazak,peraodlazak,anadolazak,anaodlazak,dolazak,odlazak;  
    cin >>pera1sat>>pera1minut; //Pera, dolazak na posao  
    cin >>pera2sat>>pera2minut; //Pera, odlazak sa posla  
    cin >>ana1sat>>ana1minut;  
    cin >>ana2sat>>ana2minut;  
    peradolazak=pera1sat*60+pera1minut;  
    peraodlazak=pera2sat*60+pera2minut;  
    anadolazak=ana1sat*60+ana1minut;  
    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;  
}
```

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 учитани 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	Пример 2	Пример 3
Улаз	Улаз	Улаз
23.02.2018	23.02.2020	30.1.2018
02.03.2018	1.3.2020	03.02.2018
Излаз	Излаз	Излаз
2	3	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

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 izracunajDaneUmeseceu(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)
        {
            // proverava 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 > izracunajDaneUmeseceu(godina, mesec))
        {
            mesec++;
            dan = 1;
        }
    if (mesec > 12)
        {
            godina++;
            mesec = 1;
        }
}
}

```

Stepeni

C++

```
#include <iostream>
#include <cassert>

using namespace std;

int main() {
    // učitavamo uglove u stepenima i minutima
    int ugao1, ugao2;
    cin >> ugao1 >> ugao2;
    int ugao3 = 180 - (ugao1 + ugao2);
    assert(0 < ugao3 && ugao3 < 180);

    // ako je bar jedan ugao tup, trougao je tupougli
    if (ugao1 > 90 || ugao2 > 90 || ugao3 > 90)
        cout << "tupougli" << endl;
    // u suprotnom, ako je bar jedan ugao prav, trougao je pravougli
    else if (ugao1 == 90 || ugao2 == 90 || ugao3 == 90)
        cout << "pravougli" << endl;
    // u suprotnom nema pravih ni tupih uglova, pa je trougao ostrougli
    else
        cout << "ostrougli" << endl;

    return 0;
}
```

Python

```
# učitavamo uglove
ugao1 = int(input())
ugao2 = int(input())
ugao3 = 180 - (ugao1 + ugao2)

# ako je bar jedan ugao tup, trougao je tupougli
if ugao1 > 90 or ugao2 > 90 or ugao3 > 90:
    print("tupougli")
# u suprotnom, ako je bar jedan ugao prav, trougao je pravougli
elif ugao1 == 90 or ugao2 == 90 or ugao3 == 90:
    print("pravougli")
# u suprotnom nema pravih ni tupih uglova, pa je trougao ostrougli
else:
    print("ostrougli")
```

Iks-oks

C++

```
#include <iostream>

using namespace std;

int main() {
    // dimenzija kvadrata
    const int a = 100;
    // koordinate piksela
    int x, y;
    cin >> x >> y;
    // redni broj vrste i kolone u kojoj se nalazi piksel
    int k = (x - 1) / a, v = (y - 1) / a;
    // redni broj kvadrata
    int kvadrat = 3 * v + k + 1;
    cout << kvadrat << endl;
    return 0;
}
```

```
#include <iostream>

using namespace std;

int main() {
    // koordinate piksela
    int x, y;
    cin >> x >> y;

    // redni broj kvadrata
    int kvadrat;

    // analiziramo sve slucajeve
    if (1 <= x && x <= 100 && 1 <= y && y <= 100)
        kvadrat = 1;
    if (101 <= x && x <= 200 && 1 <= y && y <= 100)
        kvadrat = 2;
    if (201 <= x && x <= 300 && 1 <= y && y <= 100)
        kvadrat = 3;
    if (1 <= x && x <= 100 && 101 <= y && y <= 200)
        kvadrat = 4;
    if (101 <= x && x <= 200 && 101 <= y && y <= 200)
        kvadrat = 5;
    if (201 <= x && x <= 300 && 101 <= y && y <= 200)
        kvadrat = 6;
    if (1 <= x && x <= 100 && 201 <= y && y <= 300)
        kvadrat = 7;
    if (101 <= x && x <= 200 && 201 <= y && y <= 300)
        kvadrat = 8;
    if (201 <= x && x <= 300 && 201 <= y && y <= 300)
        kvadrat = 9;
}
```

```
// ispisujemo resenje
cout << kvadrat << endl;
return 0;
}
```

Python

```
# dimenzija kvadrata
a = 100
# koordinate piksela
x = int(input())
y = int(input())
# redni broj vrste i kolone u kojoj se nalazi piksel
k = (x - 1) // a
v = (y - 1) // a
# redni broj kvadrata
kvadrat = 3 * v + k + 1
print(kvadrat)

# koordinate piksela
x = int(input())
y = int(input())

# analiziramo sve slucajeve
if 1 <= x and x <= 100 and 1 <= y and y <= 100:
    kvadrat = 1
if 101 <= x and x <= 200 and 1 <= y and y <= 100:
    kvadrat = 2;
if 201 <= x and x <= 300 and 1 <= y and y <= 100:
    kvadrat = 3;
if 1 <= x and x <= 100 and 101 <= y and y <= 200:
    kvadrat = 4;
if 101 <= x and x <= 200 and 101 <= y and y <= 200:
    kvadrat = 5;
if 201 <= x and x <= 300 and 101 <= y and y <= 200:
    kvadrat = 6;
if 1 <= x and x <= 100 and 201 <= y and y <= 300:
    kvadrat = 7;
if 101 <= x and x <= 200 and 201 <= y and y <= 300:
    kvadrat = 8;
if 201 <= x and x <= 300 and 201 <= y and y <= 300:
    kvadrat = 9;

# ispisujemo resenje
print(kvadrat)
```

Džudo

C++

```
#include <iostream>

using namespace std;

int main() {
    // broj dzudista u raznim kategorijama
    int broj_do_50 = 0;
    int broj_od_51_do_75 = 0;
    int broj_od_76 = 0;
    // ukupan broj dzudista
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        int tezina;
        cin >> tezina;
        if (tezina <= 50)
            broj_do_50++;
        else if (tezina <= 75)
            broj_od_51_do_75++;
        else
            broj_od_76++;
    }

    cout << broj_do_50 << endl;
    cout << broj_od_51_do_75 << endl;
    cout << broj_od_76 << endl;

    return 0;
}
```

Python

```
# broj dzudista u raznim kategorijama
broj_do_50 = 0
broj_od_51_do_75 = 0
broj_od_76 = 0
# ukupan broj dzudista
n = int(input())
for i in range(n):
    tezina = int(input())
    if tezina <= 50:
        broj_do_50 += 1
    elif tezina <= 75:
        broj_od_51_do_75 += 1
    else:
        broj_od_76 += 1
print(broj_do_50)
print(broj_od_51_do_75)
print(broj_od_76)
```


Loto

C++

```
#include <iostream>

using namespace std;

int main() {
    int n;
    cin >> n;
    for (int b1 = 1; b1 <= n-2; b1++)
        for (int b2 = b1 + 1; b2 <= n-1; b2++)
            for (int b3 = b2 + 1; b3 <= n; b3++)
                cout << b1 << " " << b2 << " " << b3 << endl;
    return 0;
}
```

Python

```
n = int(input())
for b1 in range(1, (n-2)+1):
    for b2 in range(b1+1, (n-1)+1):
        for b3 in range(b2+1, n+1):
            print(b1, b2, b3)
```

Uglovi

C++

```
#include <iostream>
#include <cassert>

using namespace std;

// pretvara ugao dat u stepenima (s) i minutima (m) u
// ugao samo u minutima
int ugao(int s, int m) {
    return s * 60 + m;
}

int main() {
    // ucitavamo uglove u stepenima i minutima
    int ugao1_s, ugao1_m, ugao2_s, ugao2_m, ugao3_s, ugao3_m;
    cin >> ugao1_s >> ugao1_m;
    cin >> ugao2_s >> ugao2_m;

    // pretvaramo ih u uglove date samo u minutima
    int ugao1 = ugao(ugao1_s, ugao1_m);
    int ugao2 = ugao(ugao2_s, ugao2_m);
    int ugao3 = ugao(180, 0) - (ugao1 + ugao2);

    ugao3_s = ugao3 / 60; ugao3_m = ugao3 % 60;

    // prav ugao u minutima
    int pravUgao = ugao(90, 0);
    // ako je bar jedan ugao tup, trougao je tupougli
    if (ugao1 > pravUgao || ugao2 > pravUgao || ugao3 > pravUgao)
        cout << "tupougli" << endl;
    // u suprotnom, ako je bar jedan ugao prav, trougao je pravougli
    else if (ugao1 == pravUgao || ugao2 == pravUgao || ugao3 == pravUgao)
        cout << "pravougli" << endl;
    // u suprotnom nema pravih ni tupih uglova, pa je trougao ostrougli
    else
        cout << "ostrougli" << endl;

    return 0;
}
```

Python

```
# pretvara ugao dat u stepenima (s) i minutima (m) u
# ugao samo u minutima
def ugao(s, m):
    return s * 60 + m

# ucitavamo uglove u stepenima i minutima
ugao1_s = int(input())
ugao1_m = int(input())
ugao2_s = int(input())
ugao2_m = int(input())
# pretvaramo ih u uglove date samo u minutima
```

```
ugao1 = ugao(ugao1_s, ugao1_m)
ugao2 = ugao(ugao2_s, ugao2_m)
ugao3 = ugao(180, 0) - (ugao1 + ugao2)

# prav ugao u minutima
pravUgao = ugao(90, 0)
# ako je bar jedan ugao tup, trougao je tupougli
if ugao1 > pravUgao or ugao2 > pravUgao or ugao3 > pravUgao:
    print("tupougli")
# u suprotnom, ako je bar jedan ugao prav, trougao je pravougli
elif ugao1 == pravUgao or ugao2 == pravUgao or ugao3 == pravUgao:
    print("pravougli")
# u suprotnom nema pravih ni tupih uglova, pa je trougao ostrougli
else:
    print("ostrougli")
```

Sudoku

C++

```
#include <iostream>

using namespace std;

int main() {
    // dimenzije kvadratica
    const int a = 30;
    // koordinate piksela
    int x, y;
    cin >> x >> y;
    // vrsta i kolona u kojoj se nalazi kvadratic (na polju 9x9), brojano od 0
    int kolona = (x - 1) / a;
    int vrsta = (y - 1) / a;
    // vrsta i kolona u kojoj se nalazi kvadrat (na polju 3x3), brojano od 0
    int K = kolona / 3;
    int V = vrsta / 3;
    // redni broj kvadrata, brojano od 0
    int kvadrat = V * 3 + K;
    // ispis resenja (brojano od 1)
    cout << vrsta + 1 << endl;
    cout << kolona + 1 << endl;
    cout << kvadrat + 1 << endl;
    return 0;
}
```

Python

```
# dimenzija kvadratica
a = 30
# koordinate piksela
x = int(input())
y = int(input())
# vrsta i kolona u kojoj se nalazi kvadratic (na polju 9x9), brojano od 0
kolona = (x - 1) // a
vrsta = (y - 1) // a
# vrsta i kolona u kojoj se nalazi kvadrat (na polju 3x3), brojano od 0
K = kolona // 3
V = vrsta // 3;
# redni broj kvadrata, brojano od 0
kvadrat = V * 3 + K
# ispis resenja (brojano od 1)
print(vrsta + 1)
print(kolona + 1)
print(kvadrat + 1)
```

Kartice

C++

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int brojLosih = 0;
    int n, k;
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        vector<int> karte(k);
        for (int j = 0; j < k; j++)
            cin >> karte[j];
        if (!is_sorted(begin(karte), end(karte)))
            brojLosih++;
    }
    cout << brojLosih << endl;
    return 0;
}
```

Python

```
brojLosih = 0
n = int(input())
k = int(input())
for i in range(n):
    karte = [int(input()) for j in range(k)]
    if (not(all(karte[j] <= karte[j+1] for j in range(k-1)))):
        brojLosih += 1
print(brojLosih)
```

Prag

C++

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> poeni(n);
    for (int i = n-1; i >= 0; i--)
        cin >> poeni[i];

    int m;
    cin >> m;
    for (int i = 0; i < m; i++) {
        int prag;
        cin >> prag;
        int broj = distance(lower_bound(begin(poeni), end(poeni), prag), end(poeni));
        cout << broj << endl;
    }

    return 0;
}
```

```
#include <iostream>
#include <vector>

using namespace std;

int prvi_veci_ili_jednak(const vector<int>& a, int x) {
    int l = 0, d = a.size()-1;
    while (l <= d) {
        int s = l + (d - l) / 2;
        if (a[s] < x)
            l = s + 1;
        else
            d = s - 1;
    }
    return d + 1;
}

int main() {
    int n;
    cin >> n;
    vector<int> poeni(n);
    for (int i = n-1; i >= 0; i--)
        cin >> poeni[i];

    int m;
    cin >> m;
```

```
for (int i = 0; i < m; i++) {
    int prag;
    cin >> prag;
    cout << n - prvi_veci_ili_jednak(poeni, prag) << endl;
}

return 0;
}
```

Python

```
import bisect

n = int(input())
poeni = list(map(int, input().split()))
poeni.reverse()
m = int(input())
pragovi = map(int, input().split())
for prag in pragovi:
    print(n - bisect.bisect_left(poeni, prag))
```

Par-nepar

C++

```
#include <iostream>

using namespace std;

int main() {
    int brojLosih = 0;
    int n, k;
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        bool uRedu = true;
        bool bilaNeparna = false;
        for (int j = 0; j < k; j++) {
            int karta;
            cin >> karta;
            if (karta % 2 == 1)
                bilaNeparna = true;
            else if (bilaNeparna)
                uRedu = false;
        }
        if (!uRedu)
            brojLosih++;
    }
    cout << brojLosih << endl;
    return 0;
}
```

Python

```
brojLosih = 0
n = int(input())
k = int(input())
for i in range(n):
    uRedu = True
    bilaNeparna = False
    for j in range(k):
        karta = int(input())
        if karta % 2 == 1:
            bilaNeparna = True
        elif bilaNeparna:
            uRedu = False
    if not(uRedu):
        brojLosih += 1
print(brojLosih)
```


Numeracija

C++

```
#include <iostream>

using namespace std;

int main() {
    int n;
    cin >> n;
    int broj = 0;
    // krecemo od jednocifrenih brojeva
    int s = 10; // interval [s/10, s-1] tj. [1, 9]
    int d = 1; // broj cifara je 1
    while (s - 1 < n) {
        // dodajemo cifre koriscene za zapis brojeva u intervalu [s/10, s-1]
        broj += ((s - 1) - s/10 + 1) * d; // u intervalu [a, b] ima (a - b + 1) brojeva
        // prelazimo na sledeci interval, tj. brojeve sa jednom cifrom vise
        s *= 10;
        d += 1;
    }
    // preostali su brojevi u intervalu [s/10, n]
    // dodajemo njihove cifre
    broj += (n - s/10 + 1) * d;
    cout << broj << endl;
    return 0;
}
```

Python

```
n = int(input())
broj = 0
# krecemo od jednocifrenih brojeva
s = 10 # interval [s/10, s-1] tj. [1, 9]
d = 1 # broj cifara je 1
while s - 1 < n:
    # dodajemo cifre koriscene za zapis brojeva u intervalu [s/10, s-1]
    broj += ((s - 1) - s//10 + 1) * d # u intervalu [a, b] ima (a - b + 1) brojeva
    # prelazimo na sledeci interval, tj. brojeve sa jednom cifrom vise
    s *= 10
    d += 1

# preostali su brojevi u intervalu [s/10, n]
# dodajemo njihove cifre
broj += (n - s//10 + 1) * d
print(broj)
```

Origami

C++

```
#include <iostream>

using namespace std;

int main() {
    int a, b;
    cin >> a >> b;
    int bk = 0;
    while (b != 0) {
        bk += a / b;
        int ost = a % b;
        a = b;
        b = ost;
    }
    cout << bk << endl;
}
```

Python

```
a = int(input())
b = int(input())
bk = 0
while b != 0:
    bk = bk + a // b
    ost = a % b
    a = b
    b = ost
print(bk)
```

Koferi

C++

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    // ubrzavamo učitavanje
    ios_base::sync_with_stdio(false);

    // učitavamo traženi zbir
    int traženiZbir;
    cin >> traženiZbir;

    // izračunavamo parcijalne sume elemenata niza
    int n;
    cin >> n;
    vector<int> S(n+1);
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        S[i+1] = S[i] + x;
    }

    // u sortiranom nizu parcijalnih suma tražimo da li postoje dva
    // elementa čija je razlika jednaka traženom zbiru
    int l = 0, d = 1;
    while (d <= n) {
        if (S[d] - S[l] < traženiZbir) {
            d++;
        } else if (S[d] - S[l] > traženiZbir) {
            l++;
        } else {
            cout << l << endl;
            l++;
        }
    }
    return 0;
}
```

```
#include <iostream>

using namespace std;

int main() {
    // ubrzavamo učitavanje
    ios_base::sync_with_stdio(false);

    // učitavamo traženi zbir
    int traženiZbir;
```

```

cin >> trazenizbir;

// učitavamo elemente niza
int n;
cin >> n;
int a[50000];
for (int i = 0; i < n; i++)
    cin >> a[i];

// granice segmenta
int i = 0, j = 0;
// zbir segmenta
int zbir = a[0];
while (true) {
    // na ovom mestu vazi da je zbir = sum(ai, ..., aj) i da
    // za svako i <= j' < j vazi da je sum(ai, ..., aj') < trazenizbir

    if (zbir < trazenizbir) {
        // prelazimo na interval [i, j+1]
        j++;
        // ako takav interval ne postoji, završili smo pretragu
        if (j >= n)
            break;
        // izračunavamo zbir intervala [i, j+1] na osnovu zbira intervala [i, j]
        zbir += a[j];
    } else {
        // ako je zbir jednak traženom, vazi da je sum(ai, ..., aj) = trazenizbir
        // pa prijavljujemo interval
        if (zbir == trazenizbir)
            cout << i << endl;
        // prelazimo na interval [i+1, j]
        // izračunavamo zbir intervala [i+1, j] na osnovu zbira intervala [i, j]
        zbir -= a[i];
        i++;
    }
}

return 0;
}

```

Python

```

trazenizbir = int(input())
n = int(input())
a = list(map(int, input().split()))
i = 0
j = 0
zbir = a[0]
while True:
    if zbir < trazenizbir:
        j += 1
        if j >= n:
            break;
        zbir += a[j]
    else:
        if zbir == trazenizbir:
            print(i)
        zbir -= a[i]
        i += 1

```

```
    }  
  
    // prelaz na sledeci dan sedmice  
    //npr. subota -> nedelja, nedelja -> ponedeljak  
    danSedmice = (danSedmice + 1) % 7;  
}  
  
cout << brojacVikenda << endl;  
  
return 0;  
}
```