

**Општинско такмичење ученика основних школа из рачунарства - пети разред (10.03. 2019.)**

1. [БАНАНИЦЕ] Јована има  $m$  новчаница од 50 динара и двоструко више новчаница од 20 динара. Напиши програм који одређује колико јој новца остаје када купи  $n$  чоколадних бананица од којих свака кошта 15 динара (претпостави слободно да Јована има довољно новца да купи све бананице које жели). Са улаза се учитавају цели бројеви  $m$  и  $n$  (сваки у посебном реду,  $m < 100$ ,  $n < 100$ ).

Улаз:	Излаз:	Улаз:	Излаз:
3	120	2	90
10		6	

**Решење**

**Ц++**

```
#include <iostream>

using namespace std;

int main() {
    int m, n;
    cin >> m >> n;
    int novac = m*50 + (2*m)*20;
    int kusur = novac - n*15;
    cout << kusur << endl;
    return 0;
}
```

**Пајтон**

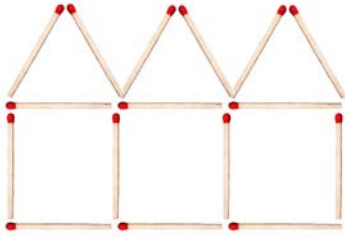
```
m = int(input())
n = int(input())
novac = m*50 + (2*m)*20
kusur = novac - n*15
print(kusur)
```

**Ц#**

```
using System;

class Program
{
    static void Main(string[] args)
    {
        int m = int.Parse(Console.ReadLine());
        int n = int.Parse(Console.ReadLine());
        int novac = m*50 + (2*m)*20;
        int kusur = novac - n*15;
        Console.WriteLine(kusur);
    }
}
```

2. [ПАЛИДРВЦА] Напиши програм који одређује колико је палидрваца шибица потребно да би се направило  $n$  кућица (број  $n$  се учитава и вредност му је између 1 и 1000).



Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:
2	11	3	16	83	416

### Решење

#### Ц++

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
    int n;
    cin >> n;
    cout << 6 + 5*(n-1) << endl;
    return 0;
}
```

#### Ц#

```
using System;
```

```
class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        Console.WriteLine(6 + 5 * (n-1));
    }
}
```

3. [УГАО] Са стандардног улаза се читава величина угла у степенима (цео број између 0 и 179) и минутима (цео број између 0 и 59). Напиши програм који испишује број степени угла, као и да ли је угао оштар, прав или туп.

Улаз:	Излаз:	Улаз:	Излаз:	Улаз:	Излаз:
130	130 tуп	90	90 прав	89	89 ostar
38		0		14	

### Решење

#### Ц++

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
    int stepeni, minuti;
```

```

cin >> stepeni >> minuti;
if (stepeni == 90 && minuti == 0)
    cout << stepeni << " " << "prav" << endl;
else if (stepeni >= 90)
    cout << stepeni << " " << "tup" << endl;
else
    cout << stepeni << " " << "ostar" << endl;
return 0;
}

```

### Пајтон

```

stepeni = int(input())
minuti = int(input())

```

```

if stepeni == 90 and minuti == 0:
    print(stepeni,"prav",sep=' ')
elif stepeni >= 90:
    print(stepeni,"tup", sep=' ')
else:
    print(stepeni,"ostar", sep=' ')

```

### Ц#

```

using System;

class Program
{
    static void Main(string[] args)
    {
        int stepeni = int.Parse(Console.ReadLine());
        int minuti = int.Parse(Console.ReadLine());
        Console.WriteLine(stepeni);
        if (stepeni == 90 && minuti == 0)
            Console.WriteLine(" prav");
        else if (stepeni >= 90)
            Console.WriteLine(" tup");
        else
            Console.WriteLine(" ostar");
    }
}

```

4. [БАКА] Напиши програм који читава број година баке, тате (њеног сина) и унуке (његове ћерке) и који одређује за колико година је бака старија од своје унуке. Бројеви се уносе у произвољном редоследу.

Улаз:	Изназ:	Улаз:	Изназ:
73	56	26	46
42		3	
17		49	

### Решење

#### Пајтон

```

a = int(input())
b = int(input())

```

```
c = int(input())
min = a
max = a
if b < min:
    min = b
elif b > max:
    max = b
if c < min:
    min = c
elif c > max:
    max = c
print(max - min)
```

## **Ц++**

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

```
using namespace std;
```

```
int main() {
    int a[3];
    cin >> a[0] >> a[1] >> a[2];
    sort(a, a + 3);
    cout << a[2] - a[0] << endl;
    return 0;
}
```

## **Ц#**

```
using System;
```

```
class Program
```

```
{
    static void Main(string[] args)
    {
        int[] godine = new int[3];
        godine[0] = int.Parse(Console.ReadLine());
        godine[1] = int.Parse(Console.ReadLine());
        godine[2] = int.Parse(Console.ReadLine());
        Array.Sort(godine);
        Console.WriteLine(godine[2] - godine[0]);
    }
}
```

## **Python**

```
godine = [0, 0, 0]
godine[0] = int(input())
godine[1] = int(input())
godine[2] = int(input())
godine = sorted(godine)
print(godine[2] - godine[0])
```

## Општинско такмичење ученика основних школа из рачунарства - шести разред (10.03. 2019.)

1. [УГАО] Са стандардног улаза се учитава величина угла у степенима, минутима и секундама (угао је мањи од 180 степени, али је задат тако да број минута и број секунди може бити и већи од 59). Напиши програм који исписује укупан број секунди угла, као и да ли је угао оштар, прав или туп.

Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:
23	84250	90	324000	89	325143
18	ostar	0	prav	79	tup
370		0		3	

### Решење

#### Ц++

```
#include <iostream>

using namespace std;

int uMinute(int stepeni, int minuti, int sekundi) {
    return (stepeni * 60 + minuti) * 60 + sekundi;
}

int main() {
    int stepeni, minuti, sekundi;
    cin >> stepeni >> minuti >> sekundi;
    cout << uMinute(stepeni, minuti, sekundi) << " ";
    if (uMinute(stepeni, minuti, sekundi) < uMinute(90, 0, 0))
        cout << "ostar" << endl;
    else if (uMinute(stepeni, minuti, sekundi) > uMinute(90, 0, 0))
        cout << "tup" << endl;
    else
        cout << "prav" << endl;
    return 0;
}
```

#### Пајтон

```
def uMinute(stepeni, minuti, sekundi):
    return (stepeni * 60 + minuti) * 60 + sekundi

stepeni = int(input())
minuti = int(input())
sekundi = int(input())

ugaoUMinute = uMinute(stepeni, minuti, sekundi)
if ugaoUMinute < uMinute(90, 0, 0):
    print(ugaoUMinute, "ostar", sep=' ')
elif ugaoUMinute > uMinute(90, 0, 0):
    print(ugaoUMinute, "tup", sep=' ')
else:
    print(ugaoUMinute, "prav", sep=' ')
```

2. [ОДЕЉЕЊА] Сви ученици у школи иду на драмску секцију или тренирају кошарку. Одељење је *супер* ако у њему бар 10 ученика иде на драмску секцију, бар 10 ученика тренира кошарку и бар 5 ученика иде на обе активности. Напиши програм који на основу укупног броја ученика у одељењу, броја ученика који иду на драмску секцију и броја ученика који тренирају кошарку одређује број ученика који иду на обе активности, а затим одређује и да ли је одељење једно супер одељење (ако јесте, исписује текст *super*).

Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:
27	4	25	7	30	6
14		15	super	8	
17		17		28	

### Решење

#### Ц++

```
#include <iostream>

using namespace std;

int main() {
    int broj_ucenika, dramska, kosarka;
    cin >> broj_ucenika >> dramska >> kosarka;
    int obe = (dramska + kosarka) - broj_ucenika;
    cout << obe << endl;
    if (dramska >= 10 && kosarka >= 10 && obe >= 5)
        cout << "super" << endl;
    return 0;
}
```

#### Ц#

```
using System;

class Program
{
    static void Main(string[] args)
    {
        int broj_ucenika = int.Parse(Console.ReadLine());
        int dramska = int.Parse(Console.ReadLine());
        int kosarka = int.Parse(Console.ReadLine());
        int obe = (dramska + kosarka) - broj_ucenika;
        Console.WriteLine(obe);
        if (dramska >= 10 && kosarka >= 10 && obe >= 5)
            Console.WriteLine("super");
    }
}
```

3. [БАКА] Напиши програм који учитава број година баке, тате (њеног сина) и унукe (његове ћерке) и који одређује за колико година је бака старија од своје унукe. Бројеви се уносе у произвољном редоследу.

Улаз:	Изназ:	Улаз:	Изназ:
73	56	26	46
42		3	
17		49	

4. [ДОМИНЕ] Зоран има две домине. На свакој домини су тачкицама представљене две цифре од 1 до 9. Напиши програм који одређује највећи број који Зоран може записати слагањем своје две домине једне поред друге (сваку од њих може окренути како жели и домине може поређати у произвољном редоследу).

Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:
2 3	6 1 3 2	0 9	9 0 4 3	6 2	6 3 6 2	6 6	8 8 6 6
1 6		3 4		3 6		8 8	

### Решење

#### Ц++

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

using namespace std;

int main() {
    int d11, d12, d21, d22;
    cin >> d11 >> d12 >> d21 >> d22;
    int m11 = max(d11, d12);
    int m12 = min(d11, d12);
    int m21 = max(d21, d22);
    int m22 = min(d21, d22);
    if (m11 > m21 || m11 == m21 && m12 > m22)
        cout << m11 << " " << m12 << " " << m21 << " " << m22 << endl;
    else
        cout << m21 << " " << m22 << " " << m11 << " " << m12 << endl;
    return 0;
}
```

#### Ц#

```
using System;

class Program
{
    static void Main(string[] args)
    {
        string[] str;
        str = Console.ReadLine().Split();
        int d11 = int.Parse(str[0]);
        int d12 = int.Parse(str[1]);
        str = Console.ReadLine().Split();
        int d21 = int.Parse(str[0]);
        int d22 = int.Parse(str[1]);
        int m11 = Math.Max(d11, d12);
        int m12 = Math.Min(d11, d12);
        int m21 = Math.Max(d21, d22);
        int m22 = Math.Min(d21, d22);
        if (m11 > m21 || m11 == m21 && m12 > m22)
            Console.WriteLine(m11 + " " + m12 + " " + m21 + " " + m22);
        else
            Console.WriteLine(m21 + " " + m22 + " " + m11 + " " + m12);
    }
}
```

```
}
```

### **Python**

```
(d11, d12) = map(int, input().split())
```

```
(d21, d22) = map(int, input().split())
```

```
if d11 > d12:
```

```
    d1 = 10*d11 + d12
```

```
else:
```

```
    d1 = 10*d12 + d11
```

```
if d21 > d22:
```

```
    d2 = 10*d21 + d22
```

```
else:
```

```
    d2 = 10*d22 + d21
```

```
if d1 > d2:
```

```
    d = 100*d1 + d2
```

```
else:
```

```
    d = 100*d2 + d1
```

```
print((d // 1000) % 10,
```

```
      (d // 100) % 10,
```

```
      (d // 10) % 10,
```

```
      (d // 1) % 10);
```



## Општинско такмичење ученика основних школа из рачунарства - седми разред (10.03. 2019.)

1. [ДОМИНЕ] Зоран има две домине. На свакој домини су тачкицама представљене две цифре од 1 до 9. Напиши програм који одређује највећи број који Зоран може записати слагањем своје две домине једне поред друге (сваку од њих може окренути како жели и домине може поређати у произвољном редоследу).

Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:
2 3	6 1 3 2	0 9	9 0 4 3	6 2	6 3 6 2	8 8	8 8 6 6
1 6		3 4		3 6		6 6	

2. [МЕЈЛОВИ] Пера ради у софтверској компанији у Нишу чија је централа у САД и послао је свом шефу  $n$  мејлова. За сваки мејл је познато време (сат и минут) слања, изражено у локалном, нишком времену. Ако његов шеф ради у Њујорку од 9 до 17h, по локалном, њујоршком времену које за нишким касни тачно 6 сати, напиши програм који одређује колико је мејлова шефу стигло ван његовог радног времена (мејлови у 9:00 се рачунају да су унутар, а у 17:00 да су ван радног времена). Сматрати да  $n < 100$ .

Улаз:	Изназ:
7	3
14 0	
22 59	
15 0	
9 0	
19 24	
17 0	
23 23	

### Решење

#### Ц++

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
    int broj_mejlova;
    cin >> broj_mejlova;
    int broj_van_radnog_vremena = 0;
    for (int i = 0; i < broj_mejlova; i++) {
        int sat, minut;
        cin >> sat >> minut;
        if (sat - 6 < 9 || sat - 6 >= 17)
            broj_van_radnog_vremena++;
    }
    cout << broj_van_radnog_vremena << endl;
    return 0;
}
```

#### Ц#

```
using System;
```

```
class Program
{
    static void Main(string[] args)
    {
```

```

int broj_mejlova = int.Parse(Console.ReadLine());
int broj_van_radnog_vremena = 0;
for (int i = 0; i < broj_mejlova; i++) {
    string[] str = Console.ReadLine().Split();
    int sat = int.Parse(str[0]);
    int minut = int.Parse(str[1]);
    if (sat - 6 < 9 || sat - 6 >= 17)
        broj_van_radnog_vremena++;
}
Console.WriteLine(broj_van_radnog_vremena);
}
}

```

### Пајтон

```

broj_mejlova = int(input())
broj_van_radnog_vremena = 0
for i in range(broj_mejlova):
    (sat, minut) = map(int, input().split())
    if sat - 6 < 9 or sat - 6 >= 17:
        broj_van_radnog_vremena += 1
print(broj_van_radnog_vremena)

```

3. [БРОЈЕВИ] Врсте троугла садрже наизменично непарне, па затим парне бројеве. Прва врста садржи само број 1, а свака наредна један број више од претходне. Напиши програм који за дати број врста  $n$  исписује описани троугао. Претпоставити да природан број  $n$  је мањи од 100.

Улаз:   Излаз:

```

4      1
        2 4
        1 3 5
        2 4 6 8

```

### Решење

#### Ц++

```

#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = i % 2 == 0 ? 2 : 1; j <= 2*i; j += 2)
            cout << j << " ";
        cout << endl;
    }
    return 0;
}

```

#### Ц#

```

using System;

class Program

```

```

{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        for (int i = 1; i <= n; i++) {
            for (int j = i % 2 == 0 ? 2 : 1; j <= 2*i; j += 2)
                Console.Write(j + " ");
            Console.WriteLine();
        }
    }
}

```

### Пајтон

```

n = int(input())
for i in range(1, n+1):
    for j in range(2 if i % 2 == 0 else 1, 2*i+1, 2):
        print(j, end=" ")
    print()

```

4. [ДАНИ] Лидија је за сваку петицу коју је добила током једног полугодишта записала име радног дан у коме је ту петицу добила. Напиши програм који одређује који јој је био најсрећнији дан током тог полугодишта (претпостави слободно да је добијала различит број оцена за сваки од 5 радних дана). Учитава се број петица  $n < 100$  које је добила и затим ознаке дана `pon`, `uto`, `sre`, `cet`, `pet`, у којима је те петице добијала свака у посебном реду.

Улаз: Излаз:

```

8      pon
pon
sre
cet
pon
pon
pet
cet
sre

```

### Решење

#### Ц++

```

#include <iostream>
#include <string>
#include <map>

using namespace std;

int main() {
    int n;
    cin >> n;

    map<string, int> brojPetica;
    for (int i = 0; i < n; i++) {
        string dan;
        cin >> dan;
        brojPetica[dan]++;
    }
}

```

```

}

int max = 0;
string max_dan;
for (auto it : brojPetica)
    if (it.second > max) {
        max = it.second;
        max_dan = it.first;
    }

cout << max_dan << endl;

return 0;
}

```

### Python

```

n = int(input())
brojPetica = {}
for i in range(n):
    dan = input()
    if dan in brojPetica:
        brojPetica[dan] += 1
    else:
        brojPetica[dan] = 1
max = 0
for (k, v) in brojPetica.items():
    if v > max:
        max = v
        max_dan = k

print(max_dan)

```

### C#

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        var brojPetica = new Dictionary<string, int>();

        for (int i = 0; i < n; i++) {
            string dan = Console.ReadLine();
            int trenutno;
            brojPetica.TryGetValue(dan, out trenutno);
            brojPetica[dan] = trenutno + 1;
        }

        int max = 0;
        string max_dan = "";
    }
}

```

```
foreach (var it in brojPetica)
    if (it.Value > max)
    {
        max = it.Value;
        max_dan = it.Key;
    }

Console.WriteLine(max_dan);
}
}
```

## Општинско такмичење ученика основних школа из рачунарства - осми разред (10.03. 2019.)

1. [ВРСТА ТРОУГЛА] Напиши програм који на основу дужине три странице троугла израчунава и исписује обим троугла. Исписати и поруку да ли је троугао једнакокраки, једнакостранични или неједнакостранични.

Улаз:	Изназ:	Улаз:	Изназ:	Улаз:	Изназ:
17	51 jednakostranicni	12	43 jednakokraki	13	42 nejednakostranicni
17		19		14	
17		12		15	

### Решење

Ц++

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int a, b, c;  
    cin >> a >> b >> c;  
    cout << a+b+c << " ";  
    if (a + b > c && a + c > b && b + c > a) {  
        if (a == b && b == c /* && a == c */)   
            cout << "jednakostranicni" << endl;  
        else if (a == b || b == c || a == c)  
            cout << "jednakokraki" << endl;  
        else  
            cout << "nejednakostranicni" << endl;  
    } else  
        cout << "trougao ne postoji" << endl;  
  
    return 0;  
}
```

2. [МЕЈЛОВИ] Пера ради у софтверској компанији у Нишу чија је централа у САД и послао је свом шефу  $n$  мејлова. За сваки мејл је познато време (сат и минут) слања (уједно и пријема), изражено у локалном, нишком времену. Ако његов шеф ради у Њујорку од 9 до 17h, по локалном, њујоршком времену које за нишким касни тачно 6 сати, напиши програм који одређује колико је мејлова шефу стигло ван његовог радног времена (мејлови стигли у 9:00 по њујоршком времену су унутар, а они у 17:00 су ван радног времена).

Улаз:	Изназ:
7	3
14 0	
22 59	
15 0	
9 0	
19 24	
17 0	
23 23	

3. [ДАНИ] Лидија је за сваку петицу коју је добила током једног полугодишта записала име дана у коме је ту петицу добила. Напиши програм који одређује који јој је био најсрећнији дан током полугодишта (претпостави слободно да је добијала различит број оцена за сваки од 5 радних дана). Учитава се број петица  $n < 100$  које је добила и затим ознаке дана `pon`, `uto`, `sre`, `set`, `pet`, у којима је те петице добијала свака у посебном реду.

Улаз:	Изназ:
8	pon

```
pon
sre
cet
pon
pon
pet
cet
sre
```

4. [ТРОУГАО] Троугао има  $n$  врста. Прва врста садржи само елемент 0. Свака следећа врста има два елемента више него претходна, први елемент сваке врсте је за  $d_1$  већи од првог елемента претходне врсте, док је у свакој врсти сваки наредни елемент за  $d_2$  већи од претходног. На пример, за  $d_1 = 5$ ,  $d_2 = 2$  и  $n=4$  добијамо следећи троугао.

```
      0
     5 7 9
    10 12 14 16 18
   15 17 19 21 23 25 27
```

Напиши програм који читава у посебним редовима природне бројеве  $d_1$ ,  $d_2$ ,  $n$  и испишује збир свих елемената на ивицама троугла. Ограничења:  $1 \leq d_1$ ,  $d_2 \leq 10$ ,  $2 \leq n \leq 1000$

Улаз: Излаз:

```
5      189
2
4
```

### Решење

#### Ц++

```
#include <iostream>

using namespace std;

// zbir aritmetickog niza a, a+d, a+2d, ..., a+(n-1)d
int aritmeticki(int a, int d, int n) {
    return (a + (a + (n-1)*d)) * n / 2;
}

int main() {
    int d1, d2, n;
    cin >> d1 >> d2 >> n;
    int zbir = aritmeticki(0, d1, n-1) + // leva ivica
               aritmeticki(0, d1 + 2*d2, n-1) + // desna ivica
               aritmeticki((n-1)*d1, d2, 2*n-1); // donja ivica
    cout << zbir << endl;
    return 0;
}
```

#### Ц#

```
using System;

class Program
{
    // zbir aritmetickog niza a, a+d, a+2d, ..., a+(n-1)d
    static int ZbirAritmetickogNiza(int a, int d, int n)
    {
        return (a + (a + (n-1)*d)) * n / 2;
    }
}
```

```

}

static void Main(string[] args)
{
    int d1 = int.Parse(Console.ReadLine());
    int d2 = int.Parse(Console.ReadLine());
    int n = int.Parse(Console.ReadLine());
    int zbir = ZbirAritmetickogNiza(0, d1, n-1) + // leva ivica
              ZbirAritmetickogNiza(0, d1 + 2*d2, n-1) + // desna ivica
              ZbirAritmetickogNiza((n-1)*d1, d2, 2*n-1); // donja ivica
    Console.WriteLine(zbir);
}
}

```

### **Пајтон**

# zbir aritmetickog niza a, a+d, a+2d, ..., a+(n-1)d

def aritmeticki(a, d, n):

return (a + (a + (n-1)\*d)) \* n // 2

d1 = int(input())

d2 = int(input())

n = int(input())

zbir = aritmeticki(0, d1, n-1) + \
 aritmeticki(0, d1 + 2\*d2, n-1) + \
 aritmeticki((n-1)\*d1, d2, 2\*n-1)

print(zbir)