

Окружно такмичење из програмирања за ученике основних школа

5. март 2016.

I категорија (5. и 6. разред)

1. Напишите програм **PROIZVOD**, који проналази највећи производ који се може добити ако се знак за множење постави између две цифре позитивног троцифреног броја N . На стандардном улазу је дат број N ($100 \leq N \leq 999$). У једином реду стандардног излаза потребно је исписати највећи производ.

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
795	665
100	0

ТЕСТ ПРИМЕРИ ЗА ОЦЕЊИВАЊЕ:

УЛАЗ	ИЗЛАЗ
105	50
553	265
901	90
888	704
800	0
320	60

РЕШЕЊЕ

Образујмо два производа:

- последње цифре броја N (цифра јединица $N \% 10$) и броја који образују прве две цифре броја N ($N / 10$);
- прве цифре броја N (цифра стотина $N / 100$) и броја који образују друга и трећа цифра броја N .

Решење задатка је већи од та два производа.

```
#include <iostream>
using namespace std;
int main()
{ int nbroj,m1,n1,m2,n2,s,p;
  cin >> nbroj;
  m1 = nbroj/10; n1 = nbroj%10; s = m1*n1; //1. proizvod
  m2 = nbroj/100; n2 = nbroj%100; p = m2*n2; //2. proizvod
  if (s > p) cout << s << endl;
  else cout << p << endl;
  return 0;
}
```

2. Три програмера – А, В и С (редом) пријавили су се на конкурс за избор победничке видео игрице. Жири ће свакој видео игрици дати две оцене – оцену за дизајн и оцену за функционалност. Напишите програм **KONKURS** који ће одредити који програмер је освојио прво место. Победник се бира на следећи начин: најпре се за сваку игрицу саберу све оцене. Победник је програмер који је креирао игрицу са највећим збиром оцена. Ако више програмера има једнак највећи збир оцена, онда победник је програмер који је добио већу оцену за дизајн (1. оцена). Ако се опет деси да више програмера има једнак број поена, онда победник је програмер који је први пред жиријем представио своју игрицу (поредак у ком су програмери представили игрице жирију је А, В, С). На стандардном улазу се у једном реду задају три пара природних бројева (који нису већи од 1000) међусобно раздвојени једним бланко карактером. Први пар бројева чине две оцене за првог програмера, други пар бројева су оцене за другог програмера, трећи пар бројева чине две оцене за трећег програмера.

На стандардном излазу се исписује једно од слова А, В или С, у зависности од тога који програмер је победник. Слово А се исписује ако је победио први програмер, слово В се исписује ако је победио други програмер, слово С се исписује ако је победио трећи програмер.

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
5 6 7 4 5 3	В
14 11 18 7 18 7	В

РЕШЕЊЕ:

Најпре за сваког програмера саберемо обе оцене. Потом посебно испитамо све услове под којим може да победи програмер А. Након тога испитамо услове под којим може да победи програмер В.

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

```
int main()
{ int a1, a2, b1, b2, c1, c2, sum1, sum2, sum3;
  cin >> a1 >> a2 >> b1 >> b2 >> c1 >> c2;
  sum1 = a1+a2;
  sum2 = b1+b2;
  sum3 = c1+c2;
  int o = 0; //o dobija vrednost 1 kada nadjemo pobednika

  //ispitajmo kada pobednik je programer A
  if ( (sum1 > sum2) || (sum1 == sum2 && a1 > b1) || (sum1 == sum2 && a1 == b1))
    if ( (sum1 > sum3) || (sum1 == sum3 && a1 > c1) || (sum1 == sum3 && a1 == c1))
      {cout << "A" << endl; o = 1;}

  //ako A nije pobednik, ispitajmo kada programer B moze biti pobednik
  if (!o)
    if (sum2 > sum1 || sum2 == sum1 && b1 > a1 || sum2 == sum1 && b1 == a1)
      if (sum2 > sum3 || sum2 == sum3 && b1 > c1 || sum2 == sum3 && b1 == c1)
        {cout << "B" << endl; o = 1;}

  if (!o)
    cout << "C" << endl;
  return 0;
}
```

ТЕСТ ПРИМЕРИ ЗА ОЦЕЊИВАЊЕ:

УЛАЗ	ИЗЛАЗ
8 15 2 7 3 7	А
4 4 4 4 7 1	С
6 3 5 4 5 4	А
100 100 100 100 100 100	А
1 2 1 3 1 4	С
1 2 1 4 1 3	В
1 2 1 3 1 3	В

3. Кажемо да је неки број прелеп ако су му све цифре различите. Напишите програм **ЛЕРОТА** који ће пронаћи колико има прелепих бројева у датом сегменту $[m, n]$ ($0 \leq m \leq n \leq 10\,000\,000$). У првој линији стандардног улаза дата су два цела броја m и n раздвојени бланко карактером. На стандардном излазу исписати један цео број који представља број прелепих бројева у датом сегменту.

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
90 91	2
80 90	10

Задатак се могао решити употребом низова, а могао се решити и без низова. Ево једног таквог решења.

```
#include<iostream>
using namespace std;
int main()
{
    long int m,n,broj,cifra,broj1,indikator,br=0;
    cin>>m>>n;
    for(int i=m;i<=n;i++)
    {
        indikator=1;
        broj=i;
        while(broj!=0)
        {
            cifra=broj%10;
            broj1=broj/10;
            while(broj1!=0)
            {
                if(broj1%10==cifra)
                {
                    indikator=0;
                    break;
                }
                broj1/=10;
            }
            if(indikator==0)break;
            broj/=10;
        }
        if(indikator!=0)br++;
    }
    cout<<br<<endl;
    return 0;
}
```

ТЕСТ ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
0 50	47
90 1024	659
1 32768	12233
9000001 10000000	60480
0 0	1
99999 100000	0
1 2	2

Израда задатака траје 120 минута

1. задатак – 30 поена
2. задатак – 35 поена
3. задатак – 35 поена

Општинско такмичење из програмирања за ученике основних школа
22. фебруар 2015.

II категорија (7. и 8. разред)

1. Написати програм **SUMA** који ће за дати цео број n ($1 \leq n \leq 10^9$) израчунати суму свих целих бројева од 1 до n , али тако да у суми испред бројева који су степен двојке стоји знак минус. На пример, за $n = 8$ сума је једнака вредности $-1 - 2 + 3 - 4 + 5 + 6 + 7 - 8 = 6$, где важи да бројеви 1, 2, 4, 8 су степени двојке, јер се редом могу представити као $2^0, 2^1, 2^2, 2^3$. Кажемо да је неки број степен двојке ако се може представити у облику 2^m ($m \geq 0$) односно написати као производ са m чинилаца, а сваки чинилац је једнак броју 2. У јединој линији стандардног улаза дат је цео број n . На стандардни излаз исписати цео број који представља вредност описане суме.

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
8	6
1000000000	499999998352516354

Решење

Сума свих природних бројева од 1 до n се рачуна по формули $1 + 2 + \dots + n = n \cdot (n + 1) / 2$. Али, израз у нашем задатку захтева да испред бројева који су степен двојке стоји знак минус. Дакле, ако би сабрали све бројеве од 1 до n по формули, онда тој формули морамо одузети двоструку вредност сваког садржаног степена двојке да би добили вредност траженог израза.

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

int main() {
    long long rez, n;
    cin >> n;
    rez = n * (n + 1) / 2;
    for (int i = 1; i <= n; i *= 2) {
        rez -= i * 2;
    }
    cout << rez << endl;
}
```

УЛАЗ	ИЗЛАЗ
1000000000	499999998352516354
10	25
100	4796
1	-1
901	404305
1000	498454
123456	7620491554

2.
исти као 3. задатак за 1. категорију

3. За дата два цела броја a и b ($0 < a, b < 3000$) уочимо све тачке са целобројним координатама (x, y) , за које важи $0 \leq x \leq a$ и $0 \leq y \leq b$. Напишите програм **PREBROJ**, који ће пребројати колико правих пролази кроз најмање две од ових тачака (са целобројним координатама (x, y) , за које важи $0 \leq x \leq a$ и $0 \leq y \leq b$). У једином реду стандардног улаза дати су цели бројеви a и b раздвојени једним бланко карактером. На стандардном излазу испишите број нађених различитих права.

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
1 1	6
4 4	140

УЛАЗ	ИЗЛАЗ
2 2	20
3 5	136
1 5	38
10 10	3296
1500 2000	2056539768890
2500 2000	5709581759996
2999 2999	18465826104258

Решење

Кроз све уочене тачке пролази $b+1$ хоризонталних и $a+1$ вертикалних правих што укупно даје $a + b + 2$ различитих права.

Колико има правих које пролазе кроз две од уочених тачака, али тако да те праве нису вертикалне или хоризонталне?

Идеја 1: Кроз сваке две тачке пролази једна права. Али, постоји опасност да нека права буде убројана више пута. Да би се то спречило, онда праву кроз тачке $A(x_1, y_1)$ и $B(x_2, y_2)$ бројимо само када изван дужи на правој AB нема других уочених тачки. У том случају је довољно наћи колико има правих AB , за које важи ако $x_1 < x_2$, онда $y_1 < y_2$. Аналогно се броје и остали случајеви.

Временска сложеност овог алгорита је $O(n^2)$, где n је број свих уочених тачака.

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

```
int nzd(int a, int b)
{ int r = a%b;
  while(r > 0)
    { a = b; b = r; r = a%b;}
  return b;
}
```

```
int main()
{ int a, b;
  cin >> a >> b;
  long long brojac = 0;
```

```
for(int x1=0; x1<a; x1++)
  for(int y1=0; y1<b; y1++)
    for(int x2=x1+1; x2<=a; x2++)
      for(int y2=y1+1; y2<=b; y2++)
        {
          int p = x2 - x1, q = y2 - y1;
          int d = nzd(p,q);
```

```

    int dx = p/d, dy = q/d;
    int x0 = x1 - dx, y0 = y1 - dy;
    int x3 = x2 + dx, y3 = y2 + dy;
    if((x0<0||y0<0) && (x3>ally3>b)) brojac++;
}

brojac = brojac * 2 + a + b + 2;
cout << brojac << endl;
return 0;
}

```

Идеја 2:

Нека m је максимум бројева a и b . Једна права може да прође кроз највише $m+1$ уочених тачака. Означимо низом $x[2], x[3], \dots, x[m+1]$ број правих, које пролазе кроз $2, 3, \dots, m+1$ тачки. Ове вредности се могу наћи на следећи начин:

За сваки сегмент који није хоризонтални ни вертикални (паралелан с координатним осама) и чији крајеви су две од датих тачака проверавамо колико уочених тачака лежи на сегменту. Ако крајеви тог сегмента су тачке $A(x_1, y_1)$ и $B(x_2, y_2)$, број тачки које леже на том сегменту је $\text{НЗД}(|x_1 - x_2|, |y_1 - y_2|) + 1$. Сегменти који нас занимају могу се посматрати као дијагонале свих могућих правоугаоника чије су стране паралелне са координатним осама и темена су четири од наведених тачки. При том пребројавању неке праве су бројане по неколико пута. На пример права, која пролази кроз три тачке је 2 пута бројана као права која пролази кроз две тачке. Слично, права која пролази кроз четири тачке је бројана 2 пута као права која пролази кроз три тачке и 3 пута као права која пролази кроз две тачке и тако даље.

Након одговарајућег подешавања добијају се праве вредности у низу

$x[2], x[3], \dots, x[m+1]$.

Коначно, број тражених правих је

$x[2] + x[3] + \dots + x[m+1] + a + b + 2$.

Временска сложеност овог алгоритма је $O(n)$, где n је број свих уочених тачака.

```

#include <iostream>
using namespace std;
const int Nmax=3100;
long long x[Nmax];

```

```

int nzd(int a, int b)
{ if (a == 0) return b;
  if (b == 0) return a;
  if (a >= b) return nzd(a-b,b);
  else return nzd(a,b-a);
}

```

```

int main()
{ int a, b;
  cin >> a >> b;

```

```

  int m;
  m = a;
  if (b > m) m = b;

```

```

for(int p=1; p<=a; p++)
for(int q=1; q<=b; q++)
{ int s;
  s = nzd(p,q);
  x[s+1] = x[s+1] + 2*(a-p+1)*(b-q+1);
}

```

```

}

long long br = x[m+1];
for(int i=m+1; i>=3; i--)
{ for(int j=i-1; j>=2; j--)
  x[j] = x[j] - (i-j+1)*x[i];
  br = br + x[i-1];
}

```

```
br = br+a+b+2;
```

```

cout << br << endl;
return 0;
}

```

Идеја 3: Додатна оптимизација тако да тражимо број нескративих разломака p/q , где p, q су редом позитивни делиоци редом a, b .

```

#include <iostream>
using namespace std;

```

```

int nzd(int a, int b)
{ int r = a%b;
  while(r > 0)
  { a = b; b = r; r = a%b; }
  return b;
}

```

```

int main()
{ int a, b;
  cin >> a >> b;
  if(a>b) { int pom = a; a = b; b = pom; } //razmena tako da vazi a<=b

```

```

long long brojac = 0;
for(int p=1; p <= a; p++)
for(int t=1; t <= p; t++)
  if(nzd(p,t)==1)
    for(int q = t; q<=b; q = q + p)
      brojac = brojac + (a-p+1)*(b-q+1) - max(a-2*p+1,0) * max(b-2*q+1,0);

```

```

brojac = brojac * 2 + a + b + 2;
cout << brojac << endl;

```

```

return 0;
}

```

Окружно такмичење из информатике 2017. – пети разред

1. Три другара, Алекса, Бане и Војин су одлучили да буду центлмени и да на матурској вечери отплету плес са својим другарицама, Аном, Биљаном и Вером. Да би били што складнији парови, одлучили су да при избору партнерки у обзир узму висину. Пошто је Алекса највиши од њих тројице, он ће плесати са највишом другарицом. Бане је средње висине и он ће плесати са другарицом која је средња по висини. Војин је најнижи и он ће плесати са најнижом другарицом. Напиши програм **PLES** који са стандардног улаза учитава редом Анину, Биљанину и Верину висину у сантиметрима (свака је задата у посебном реду) и исписује на стандардни излаз висине девојака са којима ће редом плесати Алекса, Бане и Војин (сваку у посебном реду).

ПРИМЕР 1

Улаз:	Излаз:
162	162
155	158
158	155

Пример 2

Улаз:	Излаз:
138	154
154	144
144	138

2. Љубица организује обуку својих другарица у прављењу веб-сајтова. Интересовање је јако велико и схватила је да ће морати да полазнице подели у две групе, у два рачунарска кабинета. Милица ће јој помоћи тако што ће одржати део обуке, али она има пуно обавеза и желела би да што краће остане у школи тог дана. Договориле су се да Милица држи обуку само док су обе групе истовремено у школи (када је само једна група у школи, обуку ће држати Љубица). Ако се са стандардног улаза учитавају сат почетка и сат завршетка обуке прве групе и сат почетка и сат завршетка обуке друге групе (сваки у посебном реду), напиши програм **ОБУКА** који одређује број сати колико ће Милица бити у школи (обрати пажњу да то може бити и нула).

Пример 1

Улаз:	Излаз:
8	1
12	
11	
14	

Пример 2

Улаз:	Излаз:
10	2
15	
12	
14	

Пример 3

Улаз:	Излаз:
9	0
10	
11	
12	

Објашњење:

Милица ће бити у школи од 11 до 12 Милица ће бити у школи од 12 до 14 Милица не мора да долази

3. Ћаци су за домаћи задатак из математике добили да израчунају збирове квадрата непарних цифара неколико бројева. Магдалена је добар програмер и одлучила је да себи помогне тако што ће написати програм који то ради. Замисли да си на њеном месту и напиши програм **ZBIR** који са стандардног улаза учитава један природан број мањи од милијарду и на стандардни излаз исписује збир квадрата његових непарних цифара (ако таквих цифара нема, рачунамо да је збир нула).

Пример 1

Улаз: 12345
Излаз: 35

Пример2

Улаз: 1111111
Излаз: 7

Пример 3

Улаз: 24680
Излаз: 0

4. Низ бројева се добија на следећи начин: почиње се од неког задатог броја и сваки наредни број се добија тако што се претходном броју дода број свих његових делилаца. На пример, ако је неки члан низа једнак 6, онда је следећи члан низа једнак $6 + 4$ тј. 10, јер постоје 4 делиоца броја 6 (то су редом бројеви 1, 2, 3, 6). Напиши програм **DELIOCI**, који чита са стандардног улаза у првом реду први члан низа ($0 < a_1 \leq 10000$) и у другом реду природан број N ($0 < N \leq 100$), а штампа на стандардни излаз N -ти члан низа a_N .

Пример 1

Улаз:	Излаз:
2	12
5	

Пример 2

Улаз:	Излаз:
1	744
100	

Објашњење:

$a_1=2, a_2=4, a_3=7, a_4=9, a_5=12$

Окружно такмичење из информатике 2017. – шести разред

1. Нађа је од оца сваки дан током једне радне недеље добијала џепарац. Напиши програм **DZEPARAC** који одређује ког дана је добила највећи џепарац и колико је тај џепарац износио. Програм са стандардног улаза учитава пет природних бројева, сваки у посебном реду (џепарац за понедељак, уторак, среду, четвртак и петак, редом), а на стандардни излаз исписује тражени дан означен текстом `pon`, `uto`, `sre`, `cet`, `pet` и након тога износ џепарца одвојен једним размаком. Може се претпоставити да је у свим данима џепарац био различит.

Пример 1

Улаз:

135
220
175
80
200

Излаз:

uto 220

Пример 2

Улаз:

110
180
105
240
225

Излаз:

cet 240

2. У програму за обраду слика, могуће је вршити правоугаону селекцију делова слике (при чему су странице селекције увек паралелне ивицама слике). Истовремено је могуће селектовати више правоугаоних области. Пера је селектовао две правоугаоне области и занима га број пиксела који се налазе у њиховом пресеку. Напиши програм **СЛИКА** који са стандардног улаза из прве линије учитава координате доњег левог темена прве области (број пиксела од леве ивице слике до леве ивице области и број пиксела од доње ивице слике до доње ивице области), њену ширину и висину (задате бројевима пиксела), затим из друге линије учитава исте податке за другу област, а на стандардни излаз исписује број пиксела који се налазе у њиховом пресеку.

Пример 1

Улаз:

100 100 50 50
125 75 75 100

Излаз:

1250

Пример 2

Улаз:

0 0 100 100
100 100 50 50

Излаз:

0

Пример 3

Улаз:

0 0 100 100
20 20 30 30

Излаз:

900

3. Осам другара се игра квиза. Напиши програм **KVIZ** који са стандардног улаза учитава редни број једног од другара (број од 1 до 8), након тога укупан број питања у квизу и затим за свако питање податак о другарима који су тачно одговорили на то питање дат у облику природног броја у коме нема поновљених цифара (на пример, број 512 означава да су на то питање тачно одговорили другари број 5, 1 и 2). Програм треба да на стандардни излаз испише укупан број питања на које је на почетку учитани другар тачно одговорио.

Пример 1

Улаз:

2
5
127
136
2
41
4568

Излаз:

2

Пример 2

Улаз:

1
4
135
412
6354
123456

Излаз:

3

Објашњење: означена питања су она на које је другар одговорио тачно.

4. Низ бројева се добија на следећи начин: почиње се од неког задатог броја и сваки наредни број се добија тако што се претходном броју дода број свих његових делилаца. На пример, ако је неки члан низа једнак 6, онда је следећи члан низа једнак $6 + 4$ тј. 10, јер постоје 4 делиоца броја 6 (то су редом бројеви 1, 2, 3, 6). Напиши програм **DELIOCI**, који чита са стандардног улаза у првом реду први члан низа ($0 < a_1 \leq 10000$) и у другом реду природан број N ($0 < N \leq 100$), а штампа на стандардни излаз N -ти члан низа a_N .

Пример 1

Улаз:

2
5

Излаз:

12

Пример 2

Улаз:

1
100

Излаз:

744

Објашњење:

$a_1=2, a_2=4, a_3=7, a_4=9, a_5=12$

Окружно такмичење из информатике 2017. – седми разред

1. Општинска такмичења из информатике су одржана у различитим општинама широм Србије. Централна комисија је за сваку општину добила податак о броју такмичара и просечан број поена такмичара у тој општини и на основу тога жели да израчуна просечан број поена у целој републици. Напиши програм **PROSEK** који учитава број општина n (природан број мањи од 100) и затим n линија таквих да се у свакој линији налазе подаци о појединачним општинама: број такмичара (природан број мањи од 100) и просечан број поена (ненегативан број заокружен на две децимале) раздвојени једним размаком, а на стандардни излаз исписује просечан број поена свих такмичара (број заокружен на једну децималу).

Пример 1

Улаз:
2
2 40.00
10 60.00

Излаз:
56.7

Пример 2

Улаз:
5
23 86.96
11 72.73
44 73.75
7 32.86
37 64.86

Излаз:
71.1

2. У програму за обраду слика, могуће је вршити правоугаону селекцију делова слике (при чему су странице селекције увек паралелне ивицама слике). Истовремено је могуће селекувати више правоугаоних области. Пера је селекувао две правоугаоне области и занима га број пиксела који се налазе у њиховом пресеку. Напиши програм **СЛИКА** који са стандардног улаза из прве линије учитава координате доњег левог темена прве области (број пиксела од леве ивице слике до леве ивице области и број пиксела од доње ивице слике до доње ивице области), њену ширину и висину (задате бројевима пиксела), затим из друге линије учитава исте податке за другу област, а на стандардни излаз исписује број пиксела који се налазе у њиховом пресеку.

Пример 1

Улаз:
100 100 50 50
125 75 75 100

Излаз:
1250

Пример 2

Улаз:
0 0 100 100
100 100 50 50

Излаз:
0

Пример 3

Улаз:
0 0 100 100
20 20 30 30

Излаз:
900

3. Марија је објавила пуно статуса на новој друштвеној мрежи, коју су испрограмирали њени школски другари, али никада није објавила више од једног статуса у току једног дана. Испоставило се, међутим, да је алгоритам сортирања који су њени другари имплементирали погрешан и статуси се не приказују у редоследу њиховог објављивања. Марију занима колико је највише статуса који су приказани један непосредно након другог, а који су објављени један након другог (не обавезно непосредно). Напиши програм **STATUSI** који са стандардног улаза учитава број приказаних статуса n (природан број мањи од сто), а затим n линија које садрже (исправне) датуме објављивања статуса (дан, месец и годину, раздвојене са по једним размаком), дате у редоследу њиховог приказивања (од првог до последњег). Програм на стандардни излаз исписује број који представља дужину најдуже растуће серије учитаних датума.

Пример 1

Улаз:
5
1 3 2016
1 5 2016
17 4 2016
18 12 2016
4 1 2017

Излаз:
3

Пример 2

Улаз:
6
11 4 2015
20 5 2015
11 7 2015
17 3 2015
11 2 2016
13 4 2015

Излаз:
3

Пример 3

Улаз:
4
1 1 2016
7 2 2016
3 5 2016
4 2 2017

Излаз:
4

Напомена: у примеру су обележене најдуже растуће серије датума.

4. Дата су два разломка $\frac{a}{b}$ и $\frac{c}{d}$. Њихов збир је такође разломак $\frac{e}{f}$ (нескратив облик разломка тј. бројеви e и f су узајамно прости). Збир збира свих делиоца за бројилац e и збира свих делиоца за именилац f означимо са S . На пример, збир разломака $\frac{1}{5}$ и $\frac{4}{6}$ је $\frac{13}{15}$, делиоци броја 13 су 1 и 13, а броја 15 су 1, 3, 5 и 15, па је $S = (1 + 13) + (1 + 3 + 5 + 15) = 38$. Напиши програм **RAZLOMCI**, који за дата два разломка налази тражени збир S . При сваком извршавању програм учитава различит број тест примера. У првом реду стандардног улаза дат је број тест примера T ($1 \leq T \leq 10$), а затим T линија тако да свака линија садржи по 2 исправна разломка $\frac{a}{b}$ и $\frac{c}{d}$ (описана редом са 4 природна броја a, b, c, d , мања од 1000, раздвојена са по једним размаком). За сваки тест, програм на стандардни излаз исписује тражени збир S .

Пример 1

Улаз:
1
1 2 3 4

Излаз:
13

Пример 2

Улаз:
3
2 5 3 6
3 7 10 3
12 123 56 567

Излаз:
31
112
6230

Пример 3

Улаз:
1
1 10 1 4

Излаз:
50

Окружно такмичење из информатике 2017. – осми разред

1. Општинска такмичења из информатике су одржана у различитим општинама широм Србије. Централна комисија је за сваку општину добила податак о броју такмичара и просечан број поена такмичара у тој општини и на основу тога жели да израчуна просечан број поена у целој републици. Напиши програм **PROSEK** који учитава број општина n (природан број мањи од 100) и затим n линија таквих да се у свакој линији налазе подаци о појединачним општинама: број такмичара (природан број мањи од 100) и просечан број поена (ненегативан број заокружен на две децимале) раздвојени једним размаком, а на стандардни излаз исписује просечан број поена свих такмичара (број заокружен на једну децималу).

Пример 1

Улаз:
2
2 40.00
10 60.00

Излаз:
56.7

Пример 2

Улаз:
5
23 86.96
11 72.73
44 73.75
7 32.86
37 64.86

Излаз:
71.1

2. Агенција ангажује глумце за различите улоге. За сваку улогу је допуштен одређен распон тежина и одређен распон висина (на пример, за неке улоге се траже ниски, дебљи глумци, а за неке високи, мршавији, а за неке улоге је распон прилично широк, јер физички изглед глумца није значајан). Агенција жели да сазна да ли постоји неки распон висина и тежина који би кандидату гарантовао да може да конкурише за све улоге које су тренутно у понуди. Напиши програм **GLUMCI** који са стандардног улаза учитава број улога, а затим за сваку улогу у посебном реду минималну и максималну тежину и минималну и максималну висину, раздвојене са по једним размаком, а на стандардни излаз исписује минималну и максималну тежину и минималну и максималну висину траженог пресека распона, односно текст `нема` ако такав распон не постоји.

Пример 1

Улаз:
2
65 75 160 175
70 80 150 190

Излаз:
70 75 160 175

Пример 2

Улаз:
3
55 70 170 190
60 85 150 175
50 65 140 160

Излаз:
`нема`

3. Марија је објавила пуно статуса на новој друштвеној мрежи, коју су испрограмирали њени школски другари, али никада није објавила више од једног статуса у току једног дана. Испоставило се, међутим, да је алгоритам сортирања који су њени другари имплементирали погрешан и статуси се не приказују у редоследу њиховог објављивања. Марију занима колико је највише статуса који су приказани један непосредно након другог, а који су објављени један након другог (не обавезно непосредно). Напиши програм **STATUSI** који са стандардног улаза учитава број приказаних статуса n (природан број мањи од сто), а затим n линија које садрже (исправне) датуме објављивања статуса (дан, месец и годину, раздвојене са по једним размаком), дате у редоследу њиховог приказивања (од првог до последњег). Програм на стандардни излаз исписује број који представља дужину најдуже растуће серије учитаних датума.

Пример 1

Улаз:
5
1 3 2016
1 5 2016
17 4 2016
18 12 2016
4 1 2017

Излаз:
3

Пример 2

Улаз: Излаз
6 3
11 4 2015
20 5 2015
11 7 2015
17 3 2015
11 2 2016
13 4 2015

Пример 3

Улаз: Излаз:
4 4
1 1 2016
7 2 2016
3 5 2016
4 2 2017

Напомена: у примеру су обележене најдуже растуће серије датума.

4. Дата су два разломка $\frac{a}{b}$ и $\frac{c}{d}$. Њихов збир је такође разломак $\frac{e}{f}$ (нескратив облик разломка тј. бројеви e и f су узајамно прости). Збир збира свих делиоца за бројилац e и збира свих делиоца за именилац f означимо са S . На пример, збир разломака $1/5$ и $4/6$ је $13/15$, делиоци броја 13 су 1 и 13 , а броја 15 су $1, 3, 5$ и 15 , па је $S = (1 + 13) + (1 + 3 + 5 + 15) = 38$. Напиши програм **RAZLOMCI**, који за дата два разломка налази тражени збир S . При сваком извршавању програм учитава различит број тест примера. У првом реду стандардног улаза дат је број тест примера T ($1 \leq T \leq 10$), а затим T линија тако да свака линија садржи по 2 исправна разломка $\frac{a}{b}$ и $\frac{c}{d}$ (описана редом са 4 природна броја a, b, c, d , мања од 1000, раздвојена са по једним размаком). За сваки тест, програм на стандардни излаз исписује тражени збир S .

Пример 1

Улаз: Излаз:
 1 13
 1 2 3 4

Пример 2

Улаз: Излаз:
 3 31
 2 5 3 6 112
 3 7 10 3 6230
 12 123 56 567

Пример 3

Улаз: Излаз:
 1 50
 1 10 1 4