

**Окружно такмичење из програмирања за ученике основних школа (22. март 2015.)**  
**I категорија (5. и 6. разред)**

*Напомена: Називи наредних задатака представљају називе градова из којих потичу такмичари (и њихови професори) који су прошле године као чланови српске репрезентације освојили 6 медаља на Јуниорској Балканијади из информатике.*

**1.** У околини Лознице налазе се прелепе шуме, ливаде и воћне плантаже. Мама коза обилази  $N$  ливада и прави букет разноврсног лишћа тако што са прве ливаде одабере и узме 1 лист, са друге ливаде узме 2 листа, са треће ливаде узме три листа, и тако даље наставља да са текуће ливаде узме 1 лист више него са претходне ливаде. На  $N$ -тој ливади, мама коза ће узети  $N$  листова. Мама ће лишће поделити на једнаке делове својој деци. У питању су 7 јарића из истоимене бајке и сваки од њих ће добити непаран број листова. Напишите програм LOZNICA који ће израчунати број листова које ће добити свако јаре, као и број листова који је остао мами кози имајући на уму да кози (као и свакој пожртвованој мајци) треба да припадне што је могуће мање листова (може и 0 листова).

**УЛАЗ:** У првом и једином реду стандардног улаза налази се природни број  $N$  ( $5 \leq N \leq 30$ ), број ливада које обилази мама коза.

**ИЗЛАЗ:** У једини ред стандардног излаза испишите два броја раздвојена размаком. Први број представља број листова које ће добити свако јаре. Други број представља број листова који је остао мами кози након поделе.

**ПРИМЕР**

УЛАЗ	ИЗЛАЗ
6	3 0
7	3 7

**2.** У Лесковцу се одржава такмичење за израду најбољих роштиљских кобасица. Мали Илија је направио  $N$  једнаких кобасица и потребно је да их (ножем) равномерно подели на  $K$  делова, тако да сваки од  $K$  чланова жирија добије једнаку количину кобасица за оцењивање. Да би подела била што квалитетнија, број резова кобасица мора бити што мањи. На пример, ако  $N=2$ ,  $K=6$ , довољно је да Илија сваку кобасицу са два реза подели на три једнака дела, што је укупно четири реза. На пример, ако  $N=3$ ,  $K=4$ , Илија може од сваке кобасице одрезати три четвртине. Три члана жирија оцениће те делове, а преостала три мања дела (од по једне четвртине) припашће четвртом члану жирија. Напишите програм LESKOVAC који ће израчунати најмањи укупан број резова потребан да се изврши тражена подела роштиљских кобасица.

**УЛАЗ:** У првом и једином реду стандардног улаза налазе се природни бројеви  $N$  и  $K$  ( $1 \leq N, K \leq 200$ ), број кобасица и број чланова жирија.

**ИЗЛАЗ:** У једини ред стандардног излаза испишите тражени минималан број резова.

**ПРИМЕР**

УЛАЗ	ИЗЛАЗ
2 6	4
3 4	3
6 2	0

**3.** Низ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55... се зове Фибоначијев низ. Прва два члана тог низа су једнаки 1, а сваки следећи члан се добија као збир претходна два члана низа. Ако се чланови Фибоначијевог низа запишу један поред другог без коришћења размака и других симбола, добија се следећи Фибоначијев низ цифара 11235813213455... Деца Београда се такмиче у погађању  $n$ -те цифре овог низа цифара. Написати програм БЕОГРАД који ће исписати  $n$ -ту цифру Фибоначијевог низа цифара.

**УЛАЗ:** У једином реду стандардног улаза се налази један природан број  $n$  ( $1 \leq n \leq 500$ )

**ИЗЛАЗ:** У једином реду стандардног излаза исписати  $n$ -ту цифру Фибоначијевог низа цифара.

**ПРИМЕР**

УЛАЗ	ИЗЛАЗ
12	4

Израда задатака траје 120 минута

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

# РЕШЕЊА ЗАДАТАКА ЗА 1. КАТЕГОРИЈУ

## 1. LOZNICA

Према пристиглој бодовној листи, овај задатак је најуспешније решен међу свим задацима са Окружног такмичења. На жалост, постоје региони у којима је и овај задатак решен са 0 поена, те ћемо зато уз имплементацију задатака дати и анализу и објашњења делова кода задатка. При објашњењу смо водили рачуна да образложимо најчешће грешке такмичара на овом задатку.

### Улазни подаци

`int n;` - укупан број ливада;

Променљива  $n$  добија своју вредност након учитавања са стандардног улаза.

### Изразни подаци

`int jare;` - број листова које ће добити свако јаре;

`int koza;` - број листова који је остао мами кози након поделе листова јарићима;

Вредности променљивих `jare`, `koza` се исписују на стандардном излазу у ЈЕДНОМ РЕДУ. Обе исписане вредности у сваком тест примеру морају бити тачне да би се тај тест пример вредновао са више од 0 поена.

### Потребне променљиве:

`int i;` - бројачка променљива у циклусу (`for` циклус);

`int s;` - укупан број убраних листова са ливада;

Променљива  $s$  се иницијализује на 0 (пре него се у њу смести резултат сабирања листова).

`int s=0;`

### Кораци алгоритма

#### 1. Израчунати укупан број листова које је мама коза сакупила:

```
for(i=1;i<=n;i++) s=s+i;
```

Такмичари који знају како је мали Гаус сабирао бројеве од 1 до 100 су користили и математичке формуле за збир бројева  $1+2+3+\dots+(n-1)+n$

#### 2. Израчунати број листова које треба поделити сваком јарету. Добија се целобројни резултат (користе се целобројни дељеник и делилац)!!!

```
int jare=s/7;
```

#### 3. Израчунати број листова који би остао мами кози након поделе листова јарићима.

```
int koza=s%7;
```

#### 4. На крају, потребно је испитати да ли је свако јаре добило непаран број листова. Ако није (тј. ако је број листова сваког јарета паран број),

```
if (jare%2==0)
```

потребно је да узмемо по један лист од сваког јарета и дамо мами кози. Зато одузимамо 1 од броја листова за свако јаре и мами кози додајемо укупно 7 листова.

```
{jare--; koza=koza+7;}
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ int n,i,s=0;
```

```
cin>>n;
```

```
for(i=1;i<=n;i++) s=s+i;
```

```
int jare=s/7;
```

```
int koza=s%7;
```

```
if (jare%2==0) {jare--; koza=koza+7;}
```

```
cout<<jare<<" "<<koza<<endl;
```

```
}
```

## 2. LESKOVAC

Замислимо да је мали Илија преспојио свих  $N$  роштиљских кобасица једну за другом и тако добио велику кобасицу.

Праведна подела из формулације задатка (*тако да сваки од  $K$  чланова жирија добије једнаку количину кобасица за оцењивање*) изискује да сваки члан комисије добије један од  $K$  једнаких делова велике кобасице. Дакле, прережимо велику кобасицу са  $K - 1$  резова.

Али, неки од тих  $K - 1$  резова се не морају учинити ножем, јер представљају крајеве малих  $N$  роштиљских кобасица које

смо спојили. На пример, ако  $N=2$ ,  $K=4$ , онда међу 3 реза само ће 2 бити учињена ножем (1. и 3. рез који ће пререзати 1. и 2. кобасицу на половине, док 2.рез ће пролазити између две мале кобасице).

Дакле, морамо пребројати колико има крајева малих роштиљских кобасица који представљају *већ изрезане крајеве*. Ако је  $i$ -ти рез по реду тзв. *већ изрезани крај*, онда важи да међу првих  $i$  делова (од  $K$  делова велике кобасице) можемо наћи неколико целих кобасица, нпр.  $M$  кобасица ( $M$  је цео број).

Дакле,  $(i / K)$  од  $N$  кобасица је једнако  $M$  кобасица тј.  $M = (i * N) / K$ .

Дакле, да би  $M$  био цео број, мора да важи да је  $i * N$  дељиво са  $K$ .

Зато креирамо бројачки циклус по сваком могућем  $i$  (из сегмента од 1 до  $K - 1$ ) којим преварамо да ли смо наишли на *већ изрезане крајеве*.

Укратко, наше решење је  $rez = K - NZD(N, K)$ .

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

int main () {
    int n, k; cin >> n; cin >> k;
    int rezani_kraj = 0;
    for (int i = 1; i < k; ++i) rezani_kraj += (i * n % k == 0);
    cout << k - 1 - rezani_kraj << endl;
    return 0;
}
```

### 3. BEOGRAD

```
#include<iostream>
using namespace std;
int brojCifara(long long a)
{
    int br=0;
    while(a!=0)
    {
        a=a/10;
        br++;
    }
    return br;
}

int main()
{
    long long f1=1, f2=1, f3; //clanovi Fibonacijevog niza
    int n,br=2,k,i;
    cin>>n;
    if(n==1||n==2) cout<<1<<endl;
    else
    {
        while (n>br)
        {
            n=n-br; //racunamo clanove niza
            f3=f1+f2; f1=f2; f2=f3;
            br=brojCifara(f3);
        }
        k=br-n;
        for(i=0;i<k;i++) f3=f3/10;
        cout<<f3%10<<endl;
    }
    return 0;
}
```

Сваки тест пример вреди 5 поена.

## 1. LOZNICA

УЛАЗ	ИЗЛАЗ
15	17 1
14	15 0

13	13 0
11	9 3
20	29 7
9	5 10

## 2. LESKOVAC

<b>УЈАЗ</b>	<b>ИЗЈАЗ</b>
20 20	0
49 7	0
35 36	35
90 54	36
56 98	84
56 48	40
51 34	17

## 3. BEOGRAD

<b>УЈАЗ</b>	<b>ИЗЈАЗ</b>
1	1
315	7
333	2
420	5
22	3
44	6
168	8

## II категорија (7. и 8. разред)

Напомена: Називи наредних задатака представљају називе градова из којих потичу такмичари (и њихови професори) који су прошле године као чланови српске репрезентације освојили 6 медаља на Јуниорској Балканијади из информатике.

1. У околини Чачка одржава се Кајсијада (фестивал Дани кајсије). Мали Стефан ће на Кајсијаду понети 7 корпица са кајсијама тако што ће убрати кајсије са **К** стабала на следећи начин: са првог стабла убере 1 кајсију, са другог стабла убере 2 кајсије, са трећег стабла убере три кајсије, и тако даље наставља да са текућег стабла убере једну кајсију више него са претходног стабла. На **К**-том стаблу, Стефан ће убрати **К** кајсија. У сваку корпицу Стефан ставља подједнак непаран број кајсија. Напишите програм САСАК који ће израчунати број кајсија које ће садржати свака корпица, као и број кајсија који је преостао малом Стефану имајући на уму да Стефану (као и сваком трудољубивом домаћину) треба да припадне што је могуће мање убраних кајсија (може и 0 кајсија).

**УЛАЗ:** У првом и једином реду стандардног улаза налази се природни број **К** ( $5 \leq K \leq 25$ ), број стабала са којих Стефан бере кајсије.

**ИЗЛАЗ:** У једини ред стандардног излаза испишите два броја раздвојена размаком. Први број представља број кајсија у свакој корпици. Други број представља број кајсија који је остао малом Стефану након смештања кајсија у корпице.

### ПРИМЕР

УЛАЗ	ИЗЛАЗ
5	1 8
8	5 1

2. Суботица је град познат по својим фестивалима. На једном музичком фестивалу, такмичар може решавати било који од 8 датих задатака за композиторе, али крајњи број освојених поена биће једнак суми поена 5 задатака који том такмичару укупно доносе највише поена. Написати програм SUBOTICA који за дат број поена који је такмичар добио на сваком задатку исписује крајњи број поена тог такмичара на овом фестивалу, а потом исписати ознаке оних 5 задатака који су у суми дали тај крајњи број поена. Сматрајте да такмичар никад неће на нека два задатка добити исти број поена.

**УЛАЗ:** У једином реду стандардног улаза се налази осам целих бројева **Р** ( $0 \leq P \leq 200$ ) тако да је вредност *i*-тог броја једнака броју поена које је такмичар добио за своје решење *i*-тог задатка. Бројеви су међусобно различити.

**ИЗЛАЗ:** У први ред стандардног излаза треба исписати крајњи број поена датог такмичара на овом фестивалу. У другом реду стандардног излаза исписати ознаке тражених пет задатака, поређаних од мање ознаке према већој и одвојене размаком. Ознаке задатака су природни бројеви од 1 до 8.

### ПРИМЕР

УЛАЗ	ИЗЛАЗ
21 28 52 45 33 66 0 65	261 3 4 5 6 8
21 0 52 80 77 108 55 45	372 3 4 5 6 7
21 30 52 80 108 11 0 87	357 2 3 4 5 8

3. Фабрика аутомобила из Крагујевца има неколико робота који помажу да се произведе што више квалитетних аутомобила. Један од таквих робота налази се на постољу правоугаоног облика које је подељено на мала поља. Поља су распоређена у правоугаону табелу са *m* редова и *n* колона. Свако поље табеле је обележено бројевима од 1 до *m\*n*. Робот се на почетку свог рада постави у поље које је обележено бројем *r*. Робот може у једном кораку да се креће до суседних поља у смеровима горе, доле, лево, десно. Напишите програм KRAGUJEVAC, који израчунава суму бројева у свим пољима до којих може стићи робот након тачно *k* корака.

**УЛАЗ:** У једином реду стандардног улаза налазе се цели бројеви *m*, *n*, *r*, *k* ( $1 < m < 100$ ,  $1 < n < 100$ ,  $1 < k < 200$ ).

**ИЗЛАЗ:** У једини ред стандардног излаза испишите један број – израчунату суму.

### ПРИМЕР

УЛАЗ	ИЗЛАЗ
3 4 6 1	24
4 3 6 1	17
4 3 6 2	32

Напомена уз пример 1: Излазни резултат се добија као сума  $24=2+5+7+10$

Напомена уз пример 2: Излазни резултат се добија као сума  $17=3+5+9$

Израда задатака траје 120 минута

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

## РЕШЕЊА ЗАДАТАКА ЗА 2. КАТЕГОРИЈУ

### 1. САСАК

Погледати анализу решења 1. задатка у I категорији.

```
#include<iostream>
using namespace std;
int main()
{ int k,i,s=0;
  cin>>k;
  for(i=1;i<=k;i++) s=s+i;
  int korpa=s/7; int stefan=s%7;
  if (korpa%2==0) {korpa--; stefan=stefan+7;}
  cout<<korpa<<" "<<stefan<<endl;
}
```

### 2. SUBOTICA

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
vector< pair<int,int> > poeni;
bool poredi1( pair<int,int> a, pair<int,int> b ) { return a.second > b.second; }
bool poredi2( pair<int,int> a, pair<int,int> b ) { return a.first < b.first; }
int main()
{
  for(int i=1; i<=8; i++)
  { int p; cin >> p;
    poeni.push_back(make_pair(i, p));
  }
  sort(poeni.begin(), poeni.end(), poredi1);
  int s = 0;
  for(int i=0; i<5; i++) s += poeni[i].second;
  cout << s << endl;
  sort(poeni.begin(), poeni.begin()+5, poredi2);
  for(int i=0; i<5; i++) cout << poeni[i].first << " ";  cout << endl;
  return 0;
}
```

Задатак је могуће решити на неколико начина. Поред 1. начина (тражења пет задатака који у суми поена дају максималну вредност), може се, заправо, пронаћи три задатка који у својој суми дају најмању вредност и памтити ознаке тих задатака.

```
min=ukupno; //suma vrednosti svih zadataka
for i=1 to 6
for j=i+1 to 7
for k=j+1 to 8
if poeni[i]+poeni[j]+poeni[k] < min
{ min=poeni[i]+poeni[j]+poeni[k];
  prvi=i; drugi=j; treci=k;
}
piši (ukupno-min);
for i=1 to 8
if (i != prvi) && (i != drugi) && (i != treci) piši (i);
```

### 3. KRAGUJEVAC

Замислимо да су поља правоугаоне табле обојена као поља шаховске табле (наизменична црно-бела поља) и да робот креће са белог поља (поље  $r$  је бело). При сваком кораку, робот стаје на поље чија боја се разликује од боје његовог текућег поља. Дакле, ако је број корака  $k$  непаран, достижна поља робота су црне боје. Ако број  $k$  је паран, боја достижних поља робота се поклапа са бојом полазног поља.

Робот се креће по блоковима наизменично по белим и црним пољима. Блок растојање између поља на позицијама  $(i,j)$  и  $(p,q)$  јесте број  $d$

$$d = |p-i| + |q-j|$$

Запишимо блок растојања у следећој правоугаоној табlici почевши од поља обележеног бројем  $r$  до одговарајућег поља.

						5	4	3	4	5		
					5	4	3	2	3	4	5	
			5	4	3	2	1	2	3	4	5	
		5	4	3	2	1	$r$	1	2	3	4	5
			5	4	3	2	1	2	3	4	5	
				5	4	3	2	3	4	5		
					5	4	3	4	5			
						5	4	5				

Дакле, за кретање од поља  $(p,q)$  до поља  $(i,j)$  робот мора да направи бар  $d$ -корачни ход, где се растојање  $d$  између два поља описује на горе описани начин.

Отуда, ако робот крене из поља  $(p,q)$  и обави тачно  $k$  корака, може да стигне до сваког поља  $(i,j)$  чије блок растојање  $d \leq k$  и  $d$  и  $k$  су исте парности и боје. Поља табlice која задовољавају ове услове могу да се обиђу на различите начине.

### Први начин

Нека  $(p,q)$  је почетно поље. Приступимо сваком пољу  $(i,j)$  у табlici и проверимо да ли блок растојање  $d$  од поља  $(p,q)$  до  $(i,j)$  задовољава горе наведене услове.

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

```
int main()
{ int m,n,r,k, s = 0;
  cin >> m >> n >> r >> k;
```

```
int p = 1 + (r-1)/n;
int q = 1 + (r-1)%n;
int parnost = k % 2;
```

```
for(int i=1; i<=m; i++)
  for(int j=1; j<=n; j++)
  { int d = abs(p-i) + abs(q-j);
    if(d<=k && d%2==parnost) s = s + (n*(i-1)+j);
  }
```

```
cout << s << endl;
```

```
return 0;
}
```

### Други начин.

Нека  $(p,q)$  је почетно поље. Приступимо сваком пољу  $(i,j)$  чије блок растојање  $d \leq k$  од поља  $(p,q)$  и  $d$  и  $k$  су исте парности и проверимо да ли су то координате поља у табlici.

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

```
int m,n,r,k;
```

```

int provera(int i, int j)
{ if(1<=i && i<=m && 1<=j && j<=n) return n*(i-1)+j;
  return 0;
}

int main()
{ cin >> m >> n >> r >> k;

  int p = 1 + (r-1)/n;
  int q = 1 + (r-1)%n;

  int s = 0;
  if(k%2==0) s = r;
  for(int d=k; d>0; d=d-2)
  { s = s + provera(p-d,q)+provera(p+d,q)+provera(p,q-d)+provera(p,q+d);
    for(int dp = 1; dp<d; dp++)
    { int dq = d - dp;
      s = s + provera(p+dp,q+dq)+provera(p+dp,q-dq)+provera(p-dp,q+dq)+provera(p-dp,q-dq);
    }
  }

  cout << s << endl;

  return 0;
}

```

Сваки тест пример вреди 5 поена.

### 1. САСАК

УЛАЗ	ИЗЛАЗ
15	17 1
14	15 0
12	11 1
11	9 3
20	29 7
9	5 10

### 2. SUBOTICA

УЛАЗ	ИЗЛАЗ
80 35 90 52 40 88 57 3	367 1 3 4 6 7
95 45 50 90 2 105 1 111	451 1 3 4 6 8
35 132 31 94 12 53 22 37	351 1 2 4 6 8
86 6 25 13 5 91 2 64	279 1 3 4 6 8
135 145 90 132 0 50 136 141	689 1 2 4 7 8
35 55 70 68 89 160 157 90	566 3 5 6 7 8
74 5 100 90 40 130 110 78	508 3 4 6 7 8

### 3. KRAGUJEVAC



<b>УЛАЗ</b>	<b>ИЗЛАЗ</b>
8 10 40 2	236
15 25 1 10	3286
30 50 1500 10	47625
90 90 4000 40	6720040
40 80 1500 50	1953711
80 80 3000 20	1323000
60 50 1234 25	777785