

Општинско такмичење из програмирања за ученике основних школа
21. фебруар 2016.
I категорија (5. и 6. разред)

Израда задатака траје 120 минута

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

1. Сунђер Боб има тајни задатак: мора да направи супер плочицу у облику правоугаоника. Познато је колико износи обим плочице, и позната је дужина једне стране плочице. Напишите програм **PRAVOUGAONIK** који ће израчунати дужину друге стране правоугаоника. У једином реду стандардног улаза дата су два природна броја **O**, **A** (међусобно раздвојени једним бланко карактером) који представљају редом обим и дужину једне стране правоугаоника ($4 \leq O \leq 4000$, $1 \leq A \leq 1000$, **O** је паран број). На стандардном излазу исписати број који представља дужину друге стране правоугаоника.

ТЕСТ ПРИМЕР

УЛАЗ	ИЗЛАЗ
12 1	5

РЕШЕЊЕ:

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

```
int main()
{ int o, a, b;
  cin >> o >> a;
  b = (o-2*a)/2;
  cout << b << endl;
  return 0;
}
```

УЛАЗ	ИЗЛАЗ
2364 1000	182
2000 879	121
156 48	30
264 11	121
4000 36	1964
144 1	71

2. Госпођа Пуфна воли да игра игрице, те је на хард диску свог рачунара сместила доста компјутерских игрица. Али, госпођа Пуфна жели да на рачунару сачува и песму свог омиљеног бенда, те мора да израчуна колико има места на хард диску након што је сачувала песму која дугачка **n** минута и **m** секунди. Познато је да преостали слободни простор на хард диску је **k** MB (megabyte), као и да за запис 1 секунде звука је потребно 16 KB (kilobyte). Напишите програм **KONVERZIJA** који исписује (у једном или у више редова) два броја: колико је MB и колико KB простора преостало на хард диску. Сетите се да $1 \text{ MB} = 1024 \text{ KB}$ (1 megabyte има 1024 kilobyte). У једином реду стандардног улаза дата су цели бројеви **n**, **m**, **k** ($1 \leq n, m, k \leq 1000000$) међусобно раздвојени једним бланко карактером. Сматрати да ће крајњи резултат увек чинити ненегативни бројеви (бројеви који нису мањи од 0).

ТЕСТ ПРИМЕР

УЛАЗ	ИЗЛАЗ
5 28 32	26 896

Објашњење тест примера: $n=5$ минута $m=28$ секунди $k=32$ MB

5 минута 28 секунди = 328 секунде (и за 328 секунде потребно је 5248 KB меморије)

На хард диску је пре снимања песме било $k=32\text{MB}=32768 \text{ KB}$

Након што Пуфна сними песмицу преостаће $32768\text{KB}-5248\text{KB}=27520 \text{ KB}$

Када претворимо тај број у MB и KB биће: 26 MB и 896 KB

РЕШЕЊЕ:

Да би се решио задатак, потребно је искористити однос **јединица мере** $1 \text{ MB} = 1024 \text{ KB}$.
 Ако *vreme* је време трајања песме, и ако *memorija* је капацитет аудио датотеке (изражен у KB), онда важи да $vreme = 60 * n + m$; $memorija = 16 * vreme$;

```
#include <iostream>
using namespace std;
int main()
{ int n,m,k;
cin >> n >> m >> k;
long long vreme,memorija, rezultat;
long long rez01, rez02;

vreme=60*n+m;
memorija=16*vreme;
rezultat=k*1024-memorija;
rez01=rezultat/1024;
rez02=rezultat%1024;
cout << rez01 << " " << rez02 << endl;
return 0;
}
```

УЛАЗ	ИЗЛАЗ
1 2 3	2 32
100 34 4096	4001 736
8 56 12	3 640
1 1 1	0 48
14 56 14	0 0
1 26 3	1 672
4 0 4	0 256

3. Организује се аудиција за избор глумаца за нови серијал Ратова звезда. Пријавило се n глумаца. Познато је да је међу њима има a глумаца који су високи, b глумаца који су плавооки и c глумаца који су коврцави. За главну улогу у филму потребан је само један глумац - висок, плавих очију и коврцав. Да би испланирали време за појединачне интервјуе са глумцима, режисер мора да зна колики је максимални или минимални број пријављених глумаца који су погодни за ову улогу. Напишите програм AUDICIJA, који одређује максимални или минимални број глумаца са којима режисер мора да разговара. У првом реду стандардног улаза налази се један цео број који представља или максимални или минимални број глумаца који треба да се нађе. Овај број може имати следеће вредности:

- 1, ако је потребно да се утврди минимални број учесника;
- 2, ако је потребно да се утврди максималан број учесника.

У другом реду стандардног улаза налази се четири цела броја - n, a, b, c ($1 \leq n \leq 10000, 0 \leq a, b, c \leq n$) који су међусобно раздвојени једним бланко карактером.

У једином реду стандардног излаза програм треба да испише један цео број – минимални или максимални број глумаца (у зависност од датог улазног броја 1 или 2), који могу да освоје главну улогу.

Пример 1

УЛАЗ

2
5 3 4 5

ИЗЛАЗ

3

Пример 2

УЛАЗ

1
5 3 4 5

ИЗЛАЗ

2

Објашњење примера 1: Укупан број високих глумаца је 3, те на интервју не може ићи више од 3 глумца.

Објашњење примера 2: Сви глумци су коврцави и сви глумци осим једног су плавооки. Дакле међу три висока глумца су или 3 плавоока или 2 плавоока. Дакле имамо најмање два плавоока глумца међу високим глумцима (који су коврцави, наравно). Отуда, најмање двоје глумаца се интервјуишу за главну улогу.

РЕШЕЊЕ:

Овај задатак можемо поделити на два мања посебна задатка. Починимо са посебним задатком у ком се тражи максимални број глумаца. Тада је у питању случај да тражимо што је могуће више глумаца који су у истовреме високи, плавих очију и коврџави. Без ограничења општости, можемо да сматрамо да низ таквих глумаца почиње са глумцем број 1. Идеју можемо представити следећом шемом:

	1				b		a	c	n
Високи:									
Плавооки:									
Коврџави:									

Дакле, број максималних пресека кандидата ће бити једнак минимуму ова три скупа.

Погледајмо посебан задатак у ком се тражи минимални број глумаца. Најпре ћемо наћи пресек скупа високих и плавооких. Потом ћемо за тако добијени скуп наћи пресек са скупом коврџавих.

Пресек скупа високих и плавооких је приказан у табели. Опет, без смањења општости, почињемо са првим глумцем.

	1								n
Високи:									
Плавооки:									

	1				b		a	c	n
Високи									
плавооки:									
Коврџави:									

Пресек скупа високих и плавооких глумаца је пренесен у другу табелу и нумерација почиње са првим глумцем. Отуда можемо да додамо скуп коврџавих и поново задамо нумерацију глумаца и можемо да извршимо пренумерацију као у првој табели.

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

int main()
{int k, n, a, b, c,d,m,t;
cin>>k>>n;
cin>>a>>b>>c;
if(k==2)
    { if (b>c) d=c;    else d=b; //d je min (plavookih i kovrdzavih)
      if (d>a) m=a;    else m=d; // d je min (visokih, d) = min (visokih, plavookih, kovrdzavih)
    }
else
    { if ( a+b-n>0) t=a+b-n;    else t=0; // skup visokih+plavooki mora veci od n
      if (c-n+t>0) c=c-n+t;    else c=0;
      if (t>c) m=c;    else m=t;  m=min(c,t)
    }
cout<<m<<endl;
return 0;
```

}

УЛАЗ	ИЗЛАЗ
1 9 7 5 8	2
2 9 7 5 8	5
1 6 5 3 0	0
2 600 530 312 400	312
2 6000 5300 3142 470	470
1 6000 5300 3142 470	0
1 10000 10000 10000 10000	10000

**Општинско такмичење из програмирања за ученике основних школа
21. фебруар 2016.**

II категорија (7. и 8. разред)

1. исти као 2. задатак за 1. категорију

2. Освануо је и тај дан – чувени Концерт године на ком свирају напопуларнији бендови Србије. Концерт је почео у тачно h часова m минута и s секунди и бендови су свирали укупно k минута. Током концерта бендови су направили t пауза у трајању од по 5 минута. Напишите програм **KONCERT** који ће на стандардни излаз исписати када се завршио концерт. Са стандардног улаза се учитавају три цела броја h, m, s који означавају време почетка концерта ($8 \leq h \leq 23$, $0 \leq m \leq 59$, $0 \leq s \leq 59$), цео број k који означава колико минута је трајао концерт ($5 \leq k \leq 240$), цео број t који означава број пауза ($0 \leq t \leq 12$). На стандардном излазу исписати тачно време завршетка концерта у формату $hh:mm:ss$ где hh , mm , ss је редом време завршетка концерта. Водити рачуна о испису бројева са водећим нулама.

УЛАЗ	ИЗЛАЗ
20 00 00 60 4	21:20:00

Појашњење тест примера: Време почетка концерта је 20 ч 00 минута 00 секунди. Музичирање је трајало 60 минута, а паузе су трајале $4 * 5 = 20$ минута. Зато се концерт завршио у 21:20:00

РЕШЕЊЕ:

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

```
int main()
{ int h,m,s, k, t, cas, minut;
  long long rezultat;
  cin >> h >> m >> s >> k >> t;
  rezultat=h*3600+m*60+s+k*60+t*5*60;
  cas=rezultat/3600%24;
  minut=(rezultat%3600)/60;
  cout << setfill('0') << setw(2) << cas << ":" << setfill('0') << setw(2) << minut << ":" << setfill('0') << setw(2)
  << s << endl;
```

```
return 0;
}
```

УЛАЗ	ИЗЛАЗ
20 00 00 60 4	21:20:00
20 01 02 180 4	23:21:02
21 10 00 180 4	00:30:00
20 00 35 184 5	23:29:35
22 10 11 143 7	01:08:11
08 30 00 178 7	12:03:00
08 59 59 181 7	12:35:59
10 00 00 120 12	13:00:00

3.

Дата су два природна броја a и b ($1 < a < b < 1000$). Растављање броја на просте чиниоце је поступак којим се неки природан број представља у облику производа чинилаца који су сви прости бројеви. Написати програм **RASTAVI**, који ће пребројати укупан број простих чинилаца на које се могу раставити природни бројеви који су између a и b (укључујући a и b). У једином реду стандардног улаза дата су два природна броја a и b међусобно раздвојена бланко карактером. У једином реду стандардног излаза исписати резултат.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
7 10	8
17 17	1

Објашњење 1. тест примера. Растављањем на чиниоце, бројеви од 7 до 10 се могу написати у облику производа:

$$7 = 7$$

$$8 = 2 * 2 * 2$$

$$9 = 3 * 3$$

$$10 = 2 * 5$$

Укупан број свих простих чинилаца је 8.

РЕШЕЊЕ:

```
#include <iostream>
using namespace std;
int c;
void faktorisi(int n)
{ int d=2;
  while(n>1)
    if(n%d==0) {n=n/d; c++;}
    else d++;
}

int main()
{ int a,b;
  cin >> a >> b;
  for (int i=a;i<=b;i++) faktorisi(i);
  cout << c << endl;
  return 0;
}
```

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
2 3	2
10 11	3
13 133	308
2 201	514
11 999	2856
2 999	2871
900 999	301