**Државно такмичење из рачунарства, 5.6.2021.**

**Задаци за 5. и 6. разред**

1. У прашуми је нађена чудесна таблица са 3 реда и 3 колоне испуњена цифрама. Поред таблице је нађен текст загонетке. Загонетка каже: Формирајте први троцифрени број на основу цифара са главне дијагонале таблице, почев од врха и идући ка дну таблице. Затим формирајте још један троцифрени број који се састоји од цифара у споредној дијагонали, такође почевши од врха до дна. Решење загонетке је производ ова два троцифрена броја.
Напишите програм који учитава цифре написане на таблици и исписује производ два троцифрена броја који се формирају на описани начин.
Појашњење: Главну дијагоналу таблице образују редом: прва цифра првог реда, средња цифра другог реда и последња цифра трећег реда. Споредну дијагоналу образују редом: последња цифра првог реда, средња цифра другог реда и прва цифра трећег реда.

Улаз

У сваком од три реда стандардног улаза дате су три цифре, записане без размака.

Излаз

На стандардном излазу, програм мора да испише један цео број - производ два троцифрена броја који се добија на горе описан начин.

Пример 1
Улаз
123
456
789
Излаз
56763
Појашњење примера: Број формиран од цифри на главној дијагонали је 159. Број формиран од цифри на споредној дијагонали је 357. Производ та два броја је 56763.

Пример 2
Улаз
082
020
423
Излаз
5152

Решење:

У програму се учита 9 цифри из таблице:
c11,c12,c13,c21,c22,c23,c31,c32,c33

Свега 5 цифри нам је довољно да формирамо два броја, описаних у формулацији задатка:

```
  int d1=100*int(c11-'0')+10*int(c22-'0')+c33-'0';
  int d2=100*int(c13-'0')+10*int(c22-'0')+c31-'0';
```

На крају се израчуна производ оба броја.

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

char c11,c12,c13,c21,c22,c23,c31,c32,c33;

int main()
{
  cin >> c11 >> c12 >> c13 >> c21 >> c22 >> c23 >> c31 >> c32 >> c33;

  int d1=100*int(c11-'0')+10*int(c22-'0')+c33-'0';
  int d2=100*int(c13-'0')+10*int(c22-'0')+c31-'0';

  cout << d1*d2 << endl;
}
```

2.
Ленка жели да израчуна збирове квадрата непарних цифара неколико бројева. Помози Ленки и напиши програм 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). Квадрат неке цифре је једнак производу те цифре са самом собом. На пример, квадрат цифре 3 је 3, јер 3×3=9

Пример 1
Улаз
1234560
Излаз
35
Пример 2
Улаз
11
Излаз
2
Пример 3
Улаз
888
Излаз
0

Решење:

У програму се учита број са стандардног улаза.
Поступак издвајања цифри броја се понавља све док не издвојимо сваку цифру броја (тражењем остатка при дељењу преосталог броја са 10).
Кад год је текућа цифра броја непарна, израчунамо у програму њен квадрат и додамо на збир претходно израчунатих квадрата непарних цифри.

3.
Лаза је одлучио да направи чаробни низ бројева на следећи начин: Лаза је почео да прави низ од неког задатог броја и сваки наредни члан низа 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 тачно 4 делиоца броја 6 (то су редом бројеви 1, 2, 3, 6).

Напишите програм који учитава са стандардног улаза у првом реду први члан низа ($0<a1\leq10000$) и у другом реду природан број N ($0<N\leq100$), а штампа на стандардни излаз N-ти члан Лазиног низа.

Пример 1
Улаз
2
4
Излаз
9
Појашњење: a1=2, a2=4, a3=7, a4=9

Пример 2
Улаз
3882
93
Излаз
4878

Решење:

У програму се учита број са стандардног улаза који представља први члан низа.
Поступак пребројавања делилаца сваког броја можемо издвојити у потпрограм који ће редом (почев од 1) тестирати кандидате за делиоце који нису већи од текућег броја. Потом се образује нови члан низа по правилу датом у формулацији задатка тј. нови члан низа је једнак збиру текућег члана низа и броју његових делилаца. Уочити да у програму није нужно креирати низ.

4.
Погледајмо следеће редове:
1
1 2 1

1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
...........................
Ред чији је редни број M (M=1, 2, 3, ...) се формира записивањем редом природних бројева од 1 до M, а затим записивањем бројева у обратном смеру од M-1 до 1. Сада формирамо бесконачни ред узимајући први од горе наведених редова и додајући му други ред, затим додајемо трећи ред и тако даље.

Почетак формиране бесконачне серије бројева изгледа овако: 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 1 2 3 4 5 4 3 2 1 ...
Чланови ове серије су нумерисани почев од један. Напишите програм који проналази на ком месту се дати број n јавља по k-ти пут.

Улаз

У једином реду стандардног улаза дата су два цела броја n и k, одвојена једним знаком размака (0<n≤1000000, 0<k≤1000000).

Излаз

На стандардном излазу, програм мора да испише цео број - позиција k-те појаве броја n у низу.
Пример 1
Улаз
1 1
Излаз
1
Пример 2
Улаз
2 2
Излаз
6
Пример 3
Улаз
3 3
Излаз
14

Решење:
Уочимо важно правило дато у формулацији задатка

| Број | Нови члан се додаје низу | Померај | Позиција |
|---|---|---|---|
| 1 | 1 | | 1 |
| 2 | 1 2 1 | 2 | 3 |
| 3 | 1 2 3 2 1 | 4 | 7 |
| 4 | 1 2 3 4 3 2 1 | 6 | 13 |
| 5 | 1 2 3 4 5 4 3 2 1 | 8 | 21 |
| 6 | 1 2 3 4 5 6 5 4 3 2 1 | 10 | 31 |
| 7 | 1 2 3 4 5 6 7 6 5 4 3 2 1 | 12 | 43 |

На почетку налазимо број прве појаве броја n у низу. Број 1 се први пут јавља на 1. месту, број 2 - на 3. месту, број 3 на 7. месту, 4 - на 13. месту итд. Уочимо да сваки пут померај се повећава за 2 додавањем новог члана у низу тј. додавањем новог реда.

**Државно такмичење из рачунарства, 5.6.2021.**

**Задаци за 7. и 8. разред**

1.
Напишите програм који проналази колико целих бројева који се налазе између две задате вредности a и b (укључујући a и b) има непаран број делилаца.
Улаз
У јединoм реду стандардног улаза дате су два цела броја a и b , одвојена једним знаком размака (0<a≤b<1015).
Излаз
На стандардном излазу, програм мора да испише цео број - број целих бројева између a и b (укључујући a и b) који имају непаран број делилаца.
Пример 1
Улаз
3 12
Излаз
2
Појашњење примера: Делиоци бројева од 3 до 12 су следећи:
3: 1, 3
4: 1, 2, 4
5: 1, 5
6: 1, 2, 3, 6
7: 1, 7
8: 1, 2, 4, 8
9: 1, 3, 9
10: 1, 2, 5, 10
11: 1, 11
12: 1,2,3,4,6,12
Видимо да укупно два броја (4 и 9) имају непаран број делилаца.

Решење:

Алгоритмом грубе силе израчунавамо одвојено за сваки цели број i у интервалу од a до b број делиоца за i и рачунамо колико постоји бројева i, који имају непаран број делилаца.

Да бисмо решили проблем у оквиру ограничења услова, морамо користити особину да непаран број делиоца имају само бројеви који су потпуни квадрати. То је зато што ако је цео број потпун квадрат, онда када се факторише као производ простих бројева, сваки од простих фактора који су укључени у тај производ улази у паран број пута. На пример, за потпун квадрат 1600 имамо 2*2 * 2 * 2 * 2 * 5 * 5 = 1600. Али

Сви делиоци броја x, који је потпун квадрат, добијају се формирањем производа у којима учествују прости делиоци броја x, узимајући сваки прост фактор или 0 пута, или 1 пут, или 2 пута, итд., или највећи број пута колико је учествовао у факторисању броја x, тј. за сваки прост фактор имамо непаран број могућности да га користимо.

Дакле, потребно је пребројати број потпуних квадрата у траженом интервалу.

2.
Сећате се "Баште сљезове боје", дивне књиге Бранка Ћопића и авантура малог Баје, самарџије Петрака, Бајиног дједа,... Мали Баја има неколико картица и на свакој картици је уписан цео број између 1 и 10. Баја и стари Петрак су кренули у поход на Мјесец. За то време дјед Раде који баш не види добро је поређао у картице у низ. Када се Баја вратио са неуспешног похода на Мјесец, започео је дељење картица. У сваком дељењу је желео да посложи добијене картице од најмање до највеће (нпр. ако има картица које се понављају. те картице се налазе једна уз другу). Помозите Баји тако што ћете написати програм који израчунава колико пута је Баја погрешио и није сложио карте како је желео.

Улаз
Из првог реда стандардног улаза читајте цео број N - ( 1<N<100) - број дељења које је Баја играо, затим број k (2<k<10) - број карата у сваком дељењу, а затим за свако дељење по k карата наведених онако како их је Баја сложио. Сваки број је наведен у засебном реду.
Излаз
У једном реду стандардног излаза, Ваш програм треба да прикаже број дељења у којима Баја није добро сложио картице.
Пример 1
Улаз
4
3
1
2
3
4
3
4
1
5
5
8
8

8
Излаз
1
Учитани су подаци за 4 дељења. Добро је посложено прво, треће, четврто дељење. Није добро посложено друго дељење са картицама 4,3,4.

Решење:
Потребно је за учитане поднизове проверити да ли су уређени у неопадајућем поретку. Када проверимо сваки подниз, бројач увећевамо само ако је текући подниз сортиран (уређен у неопадајућем поретку).

```
broj = 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)))):
      broj += 1
print(broj)
```

3. Диносаурус Дино ужива на турниру који је организовао познати Интернет прегледач. Сећате се игре Цхроме доносаурус, тј. Игре сакупљање кактуса. Сваки учесник у игри има право да сакупља редом узастопне кактуса, поређаних дуж праве линије.  Државна комисија треба да одреди праг за пролазак такмичара са првог нивоа игре на следећи ниво гледајући искључиво број скупљених кактуса. У овој игри учествује велики број такмичара. Дино одржава табелу са поенима и такмичари га стално питају који би број такмичара прошао даље када би доња граница пролаза била толико и толико поена (даље се пласирају сви учесници чији је број поена већи или једнак доњој граници). Дино је одлучио да напише програм који даје одговор на та питања. Са стандардног улаза учитава се број такмичара *n* (*1 ≤ n ≤ 50000*), а затим и поени такмичара (природни бројеви), задати у сортираном редоследу од највећег до најмањег и раздвојени размацима. Након тога се учитава број *m* (*1 ≤ m ≤ 50000*) који представља број питања на која Дино треба да одговори, а затим и m бројева раздвојених размацима за које је потребно дати одговор колико би се такмичара пласирало када би се тај број узео за доњу границу. На стандардни излаз исписати тражене бројеве такмичара који су се пласирали, у посебном реду за сваку доњу границу..
*Пример:*
*Улаз*
*5*
*79 63 63 46 13*
*4*
*85 40 60 5*

*Излаз*
*0*
*4*
*3*
*5*

*Појашњење*

ако је праг 85 поена, нико се није пласирао
ако је праг 50 поена, пласирали су се такмичари са освојених 79, 63, 63 и 46 поена
ако је праг 60 поена, пласирали су се такмичари са освојених 79, 63 и 63 поена
ако је праг 5 поена, сви су се такмичари пласирали

Решење:

За сваки од учитаних упита потребно је ефикасно у сортираном низу пребројати чланове који су већи од доње границе прага пролаза.

```
import bisect
n = int(input())
kaktusi = list(map(int, input().split()))
kaktusi.reverse()
m = int(input())
granica = map(int, input().split())
for x in granica:
   print(n - bisect.bisect_left(kaktusi, x))
```

4.
Дат је низ од n позитивних целих бројева $a_1, a_2, ..., a_n$. Напишите програм који која проналази позитиван цео број B са најмањом могућом вредношћу, тако да B је једнако квадрату неког целог броја c>0, тј. $B=c^2$, и важи да сваки од бројева $a_1, a_2, ..., a_n$ дели B.

Улаз

У једином реду стандардног улаза дат је број n ($0<n<10^4$). У следећем реду су дати бројеви $a_1, a_2, ..., a_n$ ($0<ai\leq42$ ) који су међусобно раздвојени са по једним знаком размака.

Излаз

На стандардном излазу, програм мора да испише један цео број, тј. број c.
Пример 1
Улаз
4 6 5 2 4
Излаз
30

**Решење:**

Израчунајмо најмањи заједнички делилац бројева $a_1, a_2, ..., a_n$ (нека је то број M). Тада наше програм представља M као производ простих бројева п [j], сваки уздигнут на снага с [j]. Основни бројеви који се користе нису већи од 41, због ограничења датог у услову задатка за вредности а1, а2, ..., ан.

Важи да можемо представити M као производ простих бројева p[j], и сваки прост фактор има експонент једнак s[j]. Користимо просте бројеве који нису већи од 41 за чланове низа a.
Када се у представи броја M, сваки степен s[j] је садржалац броја 2, онда сматрамо да решење је B = M.
Ако неки степен s[j] није садржалац од 2, онда повећавамо сваку такву вредност до најближег степена броја 2 и поставимо B да је једнако новој вредности за M. Морамо да водимо рачуна да не премашимо опсе за рад са целим бројевима.