



#	Problem Name	Time Limit	Memory Limit
1	rating	1 sec.	64 MB
2	forcycle	1 sec.	64 MB
3	warehouse	1 sec.	64 MB
4	darts	1 sec.	64 MB
5	anagramfree	3 sec.	64 MB

ამოცანა A. “პოსტების რეიტინგი”

ხშირად ფორუმებზე შეიძლება მომხმარებლის პოსტის სასარგებლოდ და საწინააღმდეგოდ ხმის მიცემა. პოსტის საწყისი რეიტინგი 0-ის ტოლია. ცხადია, რომ რაც უფრო მაღალი რეიტინგი აქვს პოსტს, იგი მით უფრო საინტერესო და სასარგებლოდ ითვლება. ყველაზე საზიზღარი პოსტები კი უარყოფითი რეიტინგით გამოირჩევა.

ჩვენ ვიმყოფებით ისეთ ფორუმზე, სადაც დადებითი ხმა პოსტის რეიტინგს 2-ით ზრდის, ხოლო უარყოფითი 1-ით ამცირებს. გვაქვს რაღაც პოსტი, რომლის მიმდინარე რეიტინგი არის R . დაადგინეთ, მინიმუმ რამდენ ადამიანს უნდა მიეცა ხმა ამ პოსტისთვის ასეთი რეიტინგის მისანიჭებლად. ერთ ადამიანს კონკრეტული პოსტისთვის ხმის მიცემა მხოლოდ ერთხელ შეუძლია.

შეზღუდვები

$$-100 \leq R \leq 100$$

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების rating.in ფაილის ერთადერთი ხაზი შეიცავს ერთ მთელ რიცხვს R .

გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების rating.out ფაილში დაბეჭდეთ მომხმარებლების ის მინიმალური რაოდენობა, რომელიც საკმარისია პოსტისთვის R რეიტინგის მისანიჭებლად.

შემომავალი ფაილის მაგალითი (rating.in)	გამომავალი ფაილის მაგალითი (rating.out)
7	5

განმარტება

თუ 4 მომხმარებელი მისცემს პოსტს დადებით ხმას, ხოლო 1 კი უარყოფითს, მისი რეიტინგი 7-ის ტოლი გახდება.

ამოცანა B. “for ციკლი”

ყველა პროგრამირების ენაში არსებობს კონსტრუქცია, რომლის მეშვეობით ხორციელდება ბრძანებების რაიმე ბლოკის მრავალჯერ განმეორება. ხშირად მას აქვს ასეთი სახე:

```
for i = first_value to second_value do
  some_action(i);
```

ეს პსევდოკოდი ასე შესრულდებოდა: i ცვლადს მიენიჭებოდა მნიშვნელობა `first_value` და შემდეგ `some_action`-ით აღნიშნული ბრძანებები შესრულდებოდა i -ს ამ მნიშვნელობის გათვალისწინებით. შემდეგ i გაიზრდებოდა 1-ით და კვლავ შესრულდებოდა `some_action`, ოღონდ i -ს ახალი მნიშვნელობის გათვალისწინებით. ასე გაგრძელდებოდა მანამ, სანამ i არ გადააჭარბებდა `second_value`-ს.

ზოგიერთ შემთხვევაში ეს უაღრესად სასარგებლო კონსტრუქცია სრულიად ზედმეტია. მაგალითად, განვიხილოთ ასეთი პსევდოკოდი:

```
ans = 0;
for i = 1 to N do
  ans = ans + i mod M;
print(ans);
```

აქ "`a mod b`" შეესაბამება a -ს b -ზე გაყოფისგან ნაშთის აღების ოპერაციას. მოცემული N და M -ისთვის დაადგინეთ, რას დაბეჭდავდა ეს პროგრამა.

შეზღუდვები

$$1 \leq N \leq 10^9$$

$$2 \leq M \leq 10^9$$

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების `forcycle.in` ფაილის ერთადერთი ხაზი შეიცავს ერთი ჰარით გამოყოფილ ორ მთელ რიცხვს N და M -ს.

გამომავალი ფაილის ფორმატი

`forcycle.out` ფაილში გამოიტანეთ `ans` ცვლადის მნიშვნელობა ციკლის დასრულების შემდეგ.

შემომავალი ფაილის მაგალითი (<code>forcycle.in</code>)	გამომავალი ფაილის მაგალითი (<code>forcycle.out</code>)
6 4	9

ამოცანა C. “საწყობი”

გვაქვს საწყობი, რომლის იატაკი N მეტრის სიგრძის და M მეტრის სიგანის მართკუთხედს წარმოადგენს. ამ საწყობში ინახავენ კუბური ფორმის ყუთებს. ყოველი ყუთის გვერდის სიგრძე 1 მეტრია. ყუთებს ყოველთვის საწყობის კედლების პარალელურად აწყობენ კედლებისგან მთელი რაოდენობის მეტრის მანძილზე. ყუთები შეიძლება იატაკზე ან ერთმანეთზე იდონ, ანუ საწყობის იატაკის ყოველ კვადრატულ მეტრზე ერთგვარი კომპი შეიძლება აშენდეს.

თქვენ გეძლევათ ინფორმაცია იმაზე, თუ რამდენი ყუთი დევს საწყობის იატაკის ყოველ კვადრატულ მეტრზე. ცნობილია, რომ მალე დამატებით K ყუთი უნდა მოიტანონ. ვინაიდან საწყობის სიმაღლე უსასრულო არაა, საჭიროა ეს ყუთები ისე დაეწყოს, რომ ყველაზე მაღალ კომპოში რაც შეიძლება ნაკლები ყუთი იყოს. საწყობში უკვე არსებული ყუთები არ უნდა გადალაგდეს, იმიტომ რომ შემდეგ მათში ძებნა გართულდება. დაადგინეთ უმაღლესი კომპის მინიმალური შესაძლო სიმაღლე.

შეზღუდვები

$$1 \leq N, M \leq 10$$

$$1 \leq K \leq 1,000,000,000$$

საწყობის ყოველ კვადრატულ მეტრზე დაწყობილი ყუთების რაოდენობა $[0, 10^6]$ შუალედიდანაა.

შემომავალი ფაილის ფორმატი

შესატან მონაცემთა warehouse.in ფაილის პირველი ხაზი შეიცავს სამ რიცხვს N, M, K . შემდეგი N ხაზიდან თითოეული M ცალ რიცხვს შეიცავს, რომლებიც გამოყოფილია თითო ჰარის ნიშნით. i -ური ხაზის j -ური რიცხვი საწყობის (i, j) კვადრატულ მეტრში დაწყობილი ყუთების რაოდენობაა, თუ დანომვრას საწყობის რომელიმე კუთხიდან დავიწყებთ.

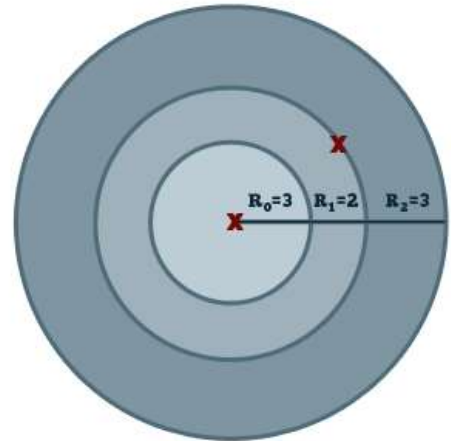
გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა warehouse.out ფაილში დაბეჭდეთ ერთი მთელი რიცხვი - ყველაზე მაღალი კომპის უმცირესი შესაძლო სიმაღლე ახალი ყუთების დამატების შემდეგ.

შემომავალი ფაილის მაგალითი (warehouse.in)	გამომავალი ფაილის მაგალითი (warehouse.out)
3 2 8 0 1 2 2 0 0	3

ამოცანა D. “დარტსი”

კედელზე დაკიდებულია დარტსის სათამაშო დაფა. დაფაზე K კონცენტრული წრეა დახაზული. ყველაზე პატარა წრის რადიუსი R_0 არის, ხოლო ყოველი შემდეგი i -ური წრის რადიუსი წინას აღმატება R_i ერთეულით. ყველაზე დიდი წრე დაფის ზომას ემთხვევა.



ყოველ წრეს თავისი შესაბამისი ქულა - S_i აქვს. განვიხილოთ საკოორდინატო სისტემა, რომლის სათავეც დაფის ცენტრში არის მოთავსებული. როდესაც მოთამაშე დაფის რომელიმე (x, y)

წერტილში ახვედრებს ხელშუბს, მას ეწერება ყველა იმ წრის ქულა, რომლებიც შეიცავს (x, y) წერტილს. თუ წერტილი რომელიმე წრეწირზე დევს, მოთამაშეს ამ წრეწირის ქულაც ეთვლება. დაფას აცდენილ ხელშუბში მოთამაშეს ქულა არ ეწერება.

მოთამაშემ ისროლა N ხელშუბი. მათ შორის i -ური მოხვდა (X_i, Y_i) წერტილში. გამოთვალეთ, სულ რამდენი ქულა დააგროვა მან.

შეზღუდვები

$$1 \leq K, N \leq 50,000$$

$$1 \leq R_i \leq 1,000 \text{ ყოველი } 0 \leq i < K\text{-სთვის}$$

$$1 \leq S_i \leq 1,000 \text{ ყოველი } 0 \leq i < K\text{-სთვის}$$

$$0 \leq |X_i|, |Y_i| \leq 10^9 \text{ ყოველი } 1 \leq i \leq N\text{-სთვის}$$

შემომავალი ფაილის ფორმატი

შესატან მონაცემთა darts.in ფაილის პირველ ხაზში ჩაწერილია ორი მთელი რიცხვი K და N . შემდეგი K ხაზიდან i -ური შეიცავს ორ რიცხვს R_i და S_i .

მომდევნო N სტრიქონიდან თითოეული შეიცავს იმ წერტილის კოორდინატებს X_i, Y_i , სადაც მოხვდა მორიგი ხელშუბი.

გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა darts.out ფაილში დაბეჭდეთ ერთი მთელი რიცხვი - მოთამაშის ჯამური ქულა ყველა სროლის შედეგად.

შემომავალი ფაილის მაგალითი (darts.in)	გამომავალი ფაილის მაგალითი (darts.out)
3 2 3 4 2 3 3 2 4 3 0 0	14

ამოცანა E. “უანაგრამო სიმრავლეები”

ორ სიტყვას ერთმანეთის ანაგრამა ეწოდება, თუ ისინი ტოლი სიგრძისაა და ერთი შეგვიძლია მეორისგან ასოების გადანაცვლებით მივიღოთ. მაგალითად, "abac", "caab" და "abca" სიტყვები ყველა ერთმანეთის ანაგრამებია, ხოლო "home" და "more" არა.

სიტყვების სიმრავლე არის უანაგრამო, თუ მასში არც ერთი ორი სიტყვა ერთმანეთის ანაგრამას არ წარმოადგენს.

განვიხილოთ N სიგრძის სიტყვები, რომლებიც შედგება მხოლოდ ინგლისური ანბანის პირველი A ასოსგან. დაადგინეთ ისეთი უანაგრამო სიმრავლეების რაოდენობა, რომლებიც შეიცავენ მხოლოდ მითითებულ სიტყვებს და რომელთა ელემენტების რაოდენობა არ აღემატება K -ს. ვინაიდან პასუხი შეიძლება ძალიან დიდი იყოს, გამოიტანეთ ნაშთი, რომელსაც იგი ტოვებს M -ზე გაყოფისას.

შეზღუდვები

$$1 \leq N \leq 100$$

$$1 \leq A \leq 4$$

$$1 \leq K \leq 100$$

$$2 \leq M \leq 1,000,000,000$$

შემომავალი ფაილის ფორმატი

შესატან მონაცემთა `anagramfree.in` ფაილის ერთადერთი ხაზი შეიცავს რიცხვებს N, A, K, M .

გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების `anagramfree.out` ფაილში ჩაწერეთ ერთადერთი რიცხვი - მოთხოვნილი უანაგრამო სიმრავლეების რაოდენობის ნაშთი M -ზე გაყოფისას.

შემომავალი ფაილის მაგალითი (<code>anagramfree.in</code>)	გამომავალი ფაილის მაგალითი (<code>anagramfree.out</code>)
2 2 2 100	10
4 3 5 81234567	10830640

განმარტება

პირველ მაგალითში ჩვენ გვჭირდება ისეთი უანაგრამო სიმრავლეების რაოდენობა, რომლებიც A და B ასოებისგან შემდგარ 2 სიგრძის არაუმეტეს 2 სიტყვას შეიცავენ. ესენია {}, {AA}, {AB}, {BA}, {BB}, {AA,AB}, {AA,BA}, {AA,BB}, {AB,BB}, {BA, BB}.