



#	Problem Name	Time Limit	Memory Limit
1	building	1 sec.	64 MB
2	bowling	1 sec.	64 MB
3	contacts	1 sec.	64 MB
4	split	1 sec.	64 MB
5	grendizer	5 sec.	64 MB

## ამოცანა A. “მე-13 კორპუსი”

ერთ-ერთ პატარა, მაგრამ სწრაფად განვითარებად ქალაქში ააშენეს ცათამბჯენების კომპლექსი და შენობები გადანომრეს ნატურალური რიცხვებით. იმ შენობის მაცხოვრებლები, რომელსაც მე-13 ნომერი ხვდა წილად, ისედაც არ არიან ბედნიერები და წარმოიდგინეთ მე-13 სართულზე ხალხი რა დღეშია. უფრო მეტად, ამ კორპუსის 13-ის ყოველ ჯერად (13, 26, 39, 52, ...) სართულზე ხალხი ძალზედ უკმაყოფილოა შექმნილი ვითარებით. ამიტომ იყო მიღებული გადაწყვეტილება, სართულების დანომვრაში არ გამოეყენათ 13-ის ჯერადი რიცხვები. იმ სართულს, რომელიც მიწიდან გადათვლით მე-13 არის, მე-14 დაარქვეს, მაგის შემდეგ - მე-15 და ასე შემდეგ. მიწიდან 25-ე სართული უკვე ნომერ 27-ს ატარებს.

თქვენ მოცემული გაქვთ სართულის მიწიდან გადათვლილი ნომერი  $N$ . დაადგინეთ მისი ნომერი მე-13 კორპუსის მაცხოვრებელთა სისტემაში.

### შეზღუდვები

$$1 \leq N \leq 200$$

### შემომავალი ფაილის ფორმატი

შესატანი მონაცემების ფაილი `building.in` შეიცავს ერთადერთ მთელ რიცხვს - მიწიდან გადათვლილ სართულის ნომერს  $N$ .

### გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების `building.out` ფაილში დაბეჭდეთ ერთი მთელი რიცხვი - სართულის ნომერი მე-13 კორპუსის მაცხოვრებელთა სისტემაში.

შემომავალი ფაილის მაგალითი ( <code>building.in</code> )	გამომავალი ფაილის მაგალითი ( <code>building.out</code> )
15	16

## ამოცანა B. “ბოულინგი”

ბოულინგში რამდენიმე ადამიანი მორიგეობით აგორებს ბურთებს კეგლების წაქცევის მიზნით. ყოველ რაუნდში ყოველ მოთამაშეს 10 კეგლის წასაქცევად 2 ცდას აძლევენ. თუ მოთამაშე 10-ივეს პირველივე ცდით აქცევს, ამბობენ რომ მან სტრაიკი შეასრულა. ამ შემთხვევაში იგი მეორე ცდას აღარ იყენებს. წინააღმდეგ შემთხვევაში იგი მეორედ აგორებს ბურთს. შემდეგ სვლა გადადის მომდევნო მოთამაშესთან, რომელიც ისევ 2 ცდაში ცდილობს 10 კეგლის წაქცევას. როდესაც ბოლო მოთამაშე გამოიყენებს თავის ცდებს, რაუნდი მთავრდება და იწყება ახალი, ისევ პირველი მოთამაშიდან.

საზოგადოდ, ბოულინგის თამაშში 10 რაუნდია. ჩვენ ასეთ შეზღუდვას არ განვიხილავთ. ვთქვათ, შედგა თამაში, რომელშიც მონაწილეობა  $K$  მოთამაშემ მიიღო. თქვენ გეძლევათ ამ თამაშის საწყისი  $N$  ცდის ჩანაწერი, ანუ ყოველ გაგორებაზე წაქცეული კეგლების რაოდენობები იმ მიმდევრობით, რომლითაც ეს თამაშის დროს ხდებოდა. ასევე ცნობილია, რომ საბოულინგე მოთამაშეს სამ მიყოლებით შესრულებულ სტრაიკში 1 ბოთლ ლუდს აძლევს უფასოდ. თუ მოთამაშე რიგში ოთხ ან მეტ სტრაიკს აკეთებს, ყოველზე მას კიდევ ერთი ბოთლი ეძლევა. დაადგინეთ, რამდენი ბოთლი იკისრა საბოულინგემ იმ თამაშში, რომელიც აღწერილია შესატან მონაცემებში.

### შეზღუდვები

$$1 \leq N \leq 100$$

$$1 \leq K \leq 10$$

თითოეულ ცდაზე წაქცეული კეგლების რაოდენობა  $[0, 10]$  შუალედშია.

თამაშის ჩანაწერი სწორია, ანუ არც ერთი მოთამაშე არ წააქცევს ერთ რაუნდში 10-ზე მეტ კეგლს.

### შემომავალი ფაილის ფორმატი

შესატან მონაცემთა `bowling.in` ფაილის პირველი ხაზი შეიცავს ორ მთელ რიცხვს  $K$  და  $N$ . შემდეგი ხაზი შეიცავს თითო ჰარით გამოყოფილ  $N$  ცალ რიცხვს, რომელთაგან  $i$ -ური თამაშის  $i$ -ურ ცდაზე წაქცეული კეგლების რაოდენობას აღნიშნავს.

### გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა `bowling.out` ფაილის ერთადერთი ხაზი უნდა შეიცავდეს 1 მთელ რიცხვს - ლუდის ბოთლების რაოდენობას, რომელიც აჩუქა საბოულინგემ მოთამაშეებს.

შემომავალი ფაილის მაგალითი (bowling.in)	გამომავალი ფაილის მაგალითი (bowling.out)
2 11 5 5 10 5 5 10 5 5 10 10 10	2
1 7 10 10 10 10 10 10 9	4

**განმარტება.**

პირველი მაგალითი: მე-9 ბურთის გაგორებაზე მეორე მოთამაშემ რიგში სამი სტრაიკი შეასრულა, რისთვისაც ლუდის ბოთლი მიიღო. ამის შემდეგ პირველმა მოთამაშემ თავისი პირველი სტრაიკი გააკეთა, შემდეგ კი მეორე მოთამაშემ რიგში მეოთხე სტრაიკი მოახდინა და მეორე ბოთლი მიიღო.

მეორე მაგალითი: მესამე სტრაიკის შემდეგ მოთამაშე ყოველ სტრაიკზე ერთ ბოთლს იგებდა ბოლო ცდამდე.

## ამოცანა C. “კონტაქტები”

მანაოს საკმაოდ ბევრი კონტაქტი დაუგროვდა მობილურ ტელეფონში. რადგანაც მას ხშირად უწევს ზარების განხორციელება, მანაო ცდილობს, რომ კონტაქტების სიაში პიროვნებები რაც შეიძლება სწრაფად მოძებნოს.

ყოველი კონტაქტის სახელი შედგება დაბალი რეგისტრის ლათინური ასოებისაგან. სიაში კონტაქტები დალაგებულია ლექსიკოგრაფიულად. სტრიქონი  $s$  ნაკლებია სტრიქონ  $t$ -ზე, თუ ის  $t$ -ს პრეფიქსს წარმოადგენს ან თუ არსებობს ისეთი პოზიცია  $i$ , რომ  $i$ -მდე ყოველ პოზიციაზე  $s$  და  $t$  ერთხვევა ერთმანეთს, ხოლო  $i$ -ურ პოზიციაზე მათ განსხვავებული ასო აქვთ და  $s$ -ის ასო ანბანში უფრო ადრე გვხვდება.

როდესაც მანაო კონტაქტის მოძებნას ცდილობს, მის წინაშე კონტაქტების სრული სია იხსნება და კურსორი პირველ კონტაქტზე უთითებს. როდესაც კურსორი საჭირო კონტაქტზე გადავა, კონტაქტი ნაპოვნად ითვლება. ამისთვის მანაოს შეუძლია ორი ხერხის გამოყენება. პირველია კონტაქტის სახელის რაღაც პრეფიქსის აკრეფა. ყოველი ახალი ასოს აკრეფისას სიიდან იშლება ყველა ის კონტაქტი, რომლის სახელის პრეფიქსს არ წარმოადგენს უკვე შეყვანილი ტექსტი. ამგვარად, ჩნდება ახალი სია (რომელიც ისევ ლექსიკოგრაფიულადაა დალაგებული) და კურსორი კვლავ მის პირველ კონტაქტზე მიუთითებს. ასევე მანაოს შეუძლია კონტაქტების სიაში (როგორც საწყისში, ისევე ტექსტის შეყვანის შედეგად მიღებულში) ზევით და ქვევით მოძრაობა. კურსორის ერთი პოზიციით ზევით ან ქვევით გადაადგილებას 1 დაჭერა სჭირდება. სიები ციკლურია, ანუ პირველ პოზიციაზე ყოფნისას კურსორის ერთი პოზიციით ზევით გადაადგილებას იგი სიის ბოლოში გადააქვს.

მანაოს მობილურ ტელეფონს აქვს სტანდარტული კლავიატურა, რომელიც შეგიძლიათ სურათზე იხილოთ. კონკრეტული ასოს დასაწერად მან იმდენჯერ უნდა დააჭიროს შესაბამის ლილაკს, მერამდენედაც წერია ეს ასო ამ ლილაკზე. მაგალითად,  $d$ -ს დასაწერად მას 1 დაჭერა, ხოლო  $r$ -ის დასაწერად კი 3 დაჭერა დასჭირდება. ერთ დაჭერას მანაო 1 წამს ანდომებს. ასევე, თუ ტექსტის შეყვანისას მანაოს მოუწევს მიყოლებით ორი ერთ ლილაკზე მოთავსებული ასოს დაწერა, მაშინ პირველის შეყვანის შემდეგ მან ერთი წამი უნდა დაიცადოს ასოს დაფიქსირებისთვის. მაგალითად  $bc$  სიტყვის ასაკრეფად დასჭირდება 6 წამი. სიაში კურსორის ზევით ან ქვევით გადასაადგილებლად დაჭერას ასევე 1 წამი სჭირდება.



თქვენ მოცემული გაქვთ მანაოს კონტაქტების სია, რომელშიც  $N$  კონტაქტია. ყოველ კონტაქტს მანაო ოპტიმალური სტრატეგიით ეძებს. გამოთვალეთ, კონკრეტული კონტაქტის მოსაძებნად მას მაქსიმუმ რამდენი წამი დასჭირდება.

### შეზღუდვები

$$1 \leq N \leq 100$$

ყოველი კონტაქტის სახელი შედგება 1-დან 100-მდე ჩათვლით ქვედა რეგისტრის ლათინური ასოსგან.

ყველა კონტაქტის სახელი განსხვავებულია.

მოცემულ სიაში კონტაქტების სახელები ზრდადობით იქნება დალაგებული.

### შემომავალი ფაილის ფორმატი

შესატანი მონაცემების contacts.in ფაილის პირველ ხაზში წერია ერთი მთელი რიცხვი  $N$ . შემდეგი  $N$  ხაზი შეიცავს მანაოს კონტაქტების სიას, თითოეულს თითო ხაზზე. ყოველი ასეთი ხაზი შეიცავს მხოლოდ კონტაქტის სახელს, მის თავში ან ბოლოში არ არის ჰარები. ყოველი სტრიქონი (მათ შორის მე- $N$ -ც) მთავრდება ახალ ხაზზე გადასვლის სიმბოლოთი.

### გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების contacts.out ფაილში დაბეჭდეთ ერთი მთელი რიცხვი - ამოცანის პასუხი.

შემომავალი ფაილის მაგალითი (contacts.in)	გამომავალი ფაილის მაგალითი (contacts.out)
7 aka akaki giorgi givi nino tamari tamuna	2

### განმარტება.

givi-ს მისაგნებად მანაომ ჯერ უნდა აკრიფოს ასო g და შემდეგ სიაში ერთი პოზიციით ზევით ან ქვევით გადავიდეს - ამ შემთხვევაში ორივე ქმედება ტოლფას შედეგს გამოიღებს. ამას მანაო ჯამში 2 წამს მოანდომებს. nino-ს მოსაძებნად საკმარისია ასო n-ის აკრეფა, რაც ასევე 2 წამს წაიღებს. aka-ს მოძებნად დრო არ სჭირდება, ვინაიდან ის სიის თავში იმყოფება. ყველა სხვა კონტაქტის მოძებნა 1 წამში ხერხდება.

## ამოცანა D. “გაყავი და იბატონე”

მანო ახალ თამაშს თამაშობს სახელად "გაყავი და იბატონე". მას აქვს  $N$  რიცხვისგან შემდგარი მიმდევრობა. მანომ უნდა შეარჩიოს რომელიმე ორი მეზობელი ინდექსი  $i$  და  $i+1$  და მათ შორის მიმდევრობა გაწყვიტოს, რის შედეგადაც მიიღებს ორ მიმდევრობას: ერთს ძველი მიმდევრობის  $1..i$  ინდექსებზე და მეორეს  $(i+1)..N$  ინდექსებზე მყოფი რიცხვებისგან. შემდეგ მან ორივე მიმდევრობაზე ანალოგიური გაყოფა უნდა ჩაატაროს და ასე გააგრძელოს მანამ, სანამ ყველა მიღებული მიმდევრობის სიგრძე არ იქნება  $K$  ან ნაკლები. არ შეიძლება მიმდევრობის გაყოფა ისე, რომ შედეგად მიღებულ ერთ-ერთი მიმდევრობაში მხოლოდ 1 ელემენტი დარჩეს.

ყოველ გაყოფას გააჩნია თავისი ფასი. მიმდევრობის ვარიაცია დავარქვათ მისი უდიდესი და უმცირესი ელემენტის სხვაობას. დავუშვათ, გაყავით რაღაც მიმდევრობა და გაყოფის შედეგად მივიღეთ  $A$  და  $B$  მიმდევრობები. ასეთი გაყოფის ფასი იქნება  $A$ -ს და  $B$ -ს ვარიაციების ნამრავლი.

მოცემული მიმდევრობისთვის დაადგინეთ მისი ბოლომდე გაყოფის უმცირესი ფასი.

### შეზღუდვები

$$4 \leq N \leq 100$$

$$3 \leq K \leq N - 1$$

საწყისი მიმდევრობის თითოეული წევრი  $[0, 999]$  შუალედშია მოთავსებული.

### შემომავალი ფაილის ფორმატი

შესატან მონაცემთა `split.in` პირველი ხაზი შეიცავს ორ მთელ რიცხვს  $N$  და  $K$ . შემდეგი  $N$  ხაზიდან  $i$ -ური შეიცავს მიმდევრობის  $i$ -ურ წევრს.

### გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა `split.out` ფაილში დაბეჭდეთ მოცემული მიმდევრობის ბოლომდე გაყოფის მინიმალური შესაძლო ფასი.

შემომავალი ფაილის მაგალითი (split.in)	გამომავალი ფაილის მაგალითი (split.out)
7 3 2 10 6 5 7 10 5	45

**განმარტება.**

მაგალითის ტესტისთვის, ოპტიმალურ პასუხს მივიღებთ მაშინ, თუ მიმდევრობას გავყოფთ მეორე ინდექსის შემდეგ. მივიღებთ ორ მიმდევრობას (2, 10) და (6, 5, 7, 10, 5). გაყოფის ფასი იქნება  $8 * 5 = 40$ . (2, 10) მიმდევრობის გაყოფა აღარ გვჭირდება, რადგანაც მისი სიგრძე არ აღემატება 3-ს, ხოლო მეორე მიმდევრობას დავყოფთ (6, 5) და (7, 10, 5) ნაწილებად. ამ დაყოფის ფასი იქნება 5. მივიღეთ სამი მიმდევრობა და დაყოფის ჯამური ფასი - 45.



## ამოცანა E. “გრენდაიზერი”

90-იან წლებში "რუსთავი 2"-ზე რამდენჯერმე აჩვენეს მულტსერიალ "გრენდაიზერის" პირველი 26 ეპიზოდი. გრენდაიზერი გიგანტური რობოტია, რომელსაც შორეული პლანეტიდან ჩამოსული დიუკ ფლიდი მართავს და მისი საშუალებით ებრძვის ბოროტ მეფე ვეგას, რათა ამ უკანასკნელის დედამიწის დაპყრობის გეგმები არ ახდეს.

განვიხილოთ დიუკ ფლიდის ერთ-ერთი მისია. იგი იმყოფება დედამიწაზე და უნდა მიაღწიოს მეფე ვეგას ფარულ თავშესაფრამდე, რომელიც შორეულ პლანეტაზეა განლაგებული. ვინაიდან გასავლელი მანძილი ძალიან დიდია, გრენდაიზერი ზოგჯერ უნდა გაჩერდეს რომელიმე პლანეტაზე და რესურსების შევსება მოახდინოს. სულ არის  $N$  პლანეტა, რომელიც დანომრილია 1-დან  $N$ -მდე რიცხვებით. დედამიწას ნომერი 1 აქვს მინიჭებული და მეფე ვეგას თავშესაფარს ნომერი  $N$ .

სხვადასხვა მიზეზების გამო (კოსმოსში მეფე ვეგას უზარმაზარი ჯარია განლაგებული, ზოგ ადგილას გაუვალი ასტეროიდების სარტყლებია მოთავსებული, შავი ხვრელების გრავიტაციას უნდა მოერიდოს და ასე შემდეგ) გრენდაიზერი მხოლოდ პლანეტების გარკვეულ წყვილებს შორის მოახერხებს გადაფრენას. ასეთი წყვილი სულ არის  $M$  ცალი და მათ შორის გრენდაიზერს ორივე მიმართულებით შეუძლია გადაადგილება. ჩვენთვის ცნობილია ეს წყვილები და მათ შორის მანძილი ასტრონომიულ ერთეულებში.

დედამიწის გარდა ყოველ პლანეტაზე გრენდაიზერს შეიძლება ელოდოს მოწინააღმდეგის მფრინავი თევზების რაზმი. დროის საწყის მომენტში პლანეტა  $i$ -ზე  $A_i$  ცალი მფრინავი თევზია. ყოველი საათის ბოლოს მათ ემატება  $B_i$  ცალი თევზი. თუ გრენდაიზერი აღწევს პლანეტაზე, მას უწევს ყველა მფრინავ თევზთან ერთდროულად ბრძოლა, რის შემდეგაც იგი ჩერდება ამ პლანეტაზე მიმდინარე საათის დასასრულამდე რესურსების შესავსებად. თუ გრენდაიზერი პლანეტაზე საწყისი მომენტიდან მთელი რაოდენობის საათის შემდეგ მოხვდება, მაშინ იგი შემდეგი საათის ბოლომდე დარჩება იქ. მე- $N$  პლანეტაზე მოხვედრისას გრენდაიზერმა უნდა დაამარცხოს ყველა იქ იმ მომენტისთვის მყოფი მფრინავი თევზი, რის შემდეგაც გამარჯვებას იზეიმებს.

გრენდაიზერს აქვს ორი მთელრიცხოვანი პარამეტრი - სიჩქარე და სიმძლავრე. მას შეუძლია ეს პარამეტრები შეარჩიოს თავისი მისიის დაწყებამდე, მაგრამ მათი ნამრავლი აუცილებლად  $S$ -ის ტოლი უნდა იყოს. სიჩქარე განსაზღვრავს, რამდენ ასტრონომიულ ერთეულს გაივლის გრენდაიზერი საათში. სიმძლავრე განსაზღვრავს, თუ რამდენი მფრინავი თევზის ერთდროულად დამარცხება შეუძლია რობოტს. თუ პლანეტაზე მას ამაზე მეტი თევზი დახვდება, იგი წააგებს. სიჩქარის და სიმძლავრის კონკრეტული მნიშვნელობების ერთობლიობას კონფიგურაცია ეწოდება. დაადგინეთ, რამდენი კონფიგურაცია იძლევა დედამიწიდან მეფე ვეგას თავშესაფრამდე მიღწევის იმედს.

### შეზღუდვები

$$2 \leq N \leq 5,000$$

$$1 \leq M \leq 10,000$$

$$0 \leq A_i \leq 1,000,000, 2 \leq i \leq N. A_i \text{ რიცხვები მთელია.}$$

$$0 \leq B_i \leq 1,000,000, 2 \leq i \leq N. B_i \text{ რიცხვები მთელია.}$$

$1 \leq L_j \leq 1,000,000$ ,  $1 \leq j \leq M$ .  $L_j$  რიცხვები მთელია.

$1 \leq S \leq 1,000,000,000$ .  $S$  მთელია.

შემოსატან მონაცემებში არ იქნება ორჯერ მითითებული ერთმანეთისგან მიღწევადი პლანეტების ერთიდაიგივე წყვილი. ასევე არ იქნება წყვილები, სადაც საწყისი და საბოლოო პლანეტა ერთმანეთს ემთხვევა.

### შემომავალი ფაილის ფორმატი

შესატან მონაცემთა `grendizer.in` ფაილის პირველი ხაზი შეიცავს სამ მთელ რიცხვს  $N$ ,  $M$  და  $S$ . შემდეგ ხაზში წერია  $(N-1)$  თითო ჰარით გამოყოფილი რიცხვი, მათგან  $i$ -ური არის  $A_{i+1}$ . მესამე ხაზში ანალოგიურად ჩაწერილია  $B_i$  რიცხვები. შემდეგ მოდის  $M$  ხაზი,  $j$ -ურ მათგანზე წერია სამი რიცხვი  $X_j$ ,  $Y_j$ ,  $L_j$  - ერთმანეთისგან მიღწევადი პლანეტების  $j$ -ური წყვილის პირველი და მეორე პლანეტა და მათ შორის მანძილი.

### გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა `grendizer.out` ფაილში ჩაწერეთ ერთი მთელი რიცხვი - გრენდაიზერის ისეთი კონფიგურაციების რაოდენობა, რომელიც იძლევიან მეფე ვეგას თავშესაფრამდე მიღწევის საშუალებას.

შემომავალი ფაილის მაგალითი ( <code>grendizer.in</code> )	გამომავალი ფაილის მაგალითი ( <code>grendizer.out</code> )
<pre>3 3 36 36 0 0 2 2 1 2 2 3 3 1 3 35</pre>	<pre>3</pre>

### განმარტება.

პირველ მაგალითში გრენდაიზერის სამი კონფიგურაცია გვაწყობს:  $\{1,36\}$ ,  $\{18,2\}$ ,  $\{36,1\}$ . პირველის შემთხვევაში მას შეუძლია, მეორე პლანეტაზე გადაფრინდეს, დაამარცხოს იქ მყოფი 36 თეფში და შემდეგ მესამე პლანეტაზე გადაფრინდეს, სადაც მას 12 თეფში დახვდება და უშანსოდ იქნებიან. მეორე შემთხვევაში მას შეუძლია მეორე საათის მიწურულს აღმოჩნდეს პირდაპირ მესამე პლანეტაზე და იქ იმ დროისთვის მყოფი 2 თეფში გაანადგუროს. მესამე შემთხვევაში იგი პირდაპირ მესამე პლანეტაზე მოასწრებს გადაფრენას ისე, რომ არც ერთი თეფში არ დახვდეს.