



გამოცდის ფორმატი

* მონიშნეთ გამოცდის ფორმატი (მიუთითეთ ✓)

დახურული წიგნი	
ღია წიგნი	✓

* ღია წიგნის შემთხვევაში მონიშნეთ გამოცდაზე ნებადართული ელემენტები (მიუთითეთ ✓)

სალექციო მასალები (პრეზენტაცია და სხვა)	
ელექტრონული წიგნები	✓
წიგნები	
კონსპექტები	
ლექსიკონი	
კალკულატორი	
ლექტორი/პლანშეტი	

* გამოცდის ჩატარების წესი იხილეთ „დესკტოპზე“ საქალაქოში Exam materials

საგამოცდო საკითხების ფორმა ვარიანტი # 1

სკოლა/ საგანმანათ ლებლო პროგრამა	მათემატიკა და კომპიუტერული მეცნიერება	სტუდენტის მიერ მიღებული ქულა	
საგანი	პროგრამირების მეთოდოლოგია		
ლექტორი	შ. ღვინეუაძე		
კურსი	I		
ჯგუფი			
გამოცდის ფორმა	ღია წიგნი		
გამოცდის ხანგრძლივობა	3 საათი		
მაქსიმალური ქულა	180		
სტუდენტის სახელი და გვარი:			

ნაშრომის ჩაბარების წესი:



ჩამოგვირთეთ თქვენი ვარიანტის შესაბამისი problems ფოლდერი თქვენს კომპიუტერში. ფოლდერში ნახავთ 5 ფაილს. თითოეულ ფაილში დაწერეთ შესაბამისი კოდი. ფაილების გახსნა შეგიძლიათ პროგრამით notepad++. ამისათვის შეგიძლიათ ფაილზე დააკლიკოთ მაუსის მარჯვენა ღილაკით და მონიშნოთ open with notepad++ ან გახსნათ notepad++ და მაუსის ღრაგით ჩააგდოთ ფაილი პროგრამაში. ყურადღებით დააკვირდით და ნამდვილად თქვენს კომპიუტერზე მყოფ ფაილში შეიტანეთ ცვლილებები.

მუშაობის დასრულების შემდეგ:

1. დააარქივეთ თქვენს მიერ ჩამოგვირთული problems ფოლდერი
2. სახელად დაარქვით:
ვარიანტის_ნომერი-თქვენი_მეილის_პრეფიქსი.
მაგალითად თუკი თქვენი ვარიანტის ნომერია 3 მაშინ არქივის სახელი შეიძლება იყოს 3-gboch17.rar
3. დააკოპირეთ არქივი დესკტოპზე(თუკი უკვე დესკტოპზე არ არის).
4. ვებ ბრაუზერში გახსენით მისამართი <http://192.168.210.5> და ატვირთეთ არქივი (ატვირთვის დროს საიგი ფაილის სახელს არ დაგიწერთ, უბრალოდ ღილაკზე დაჭერის შემდეგ თუ გამწვანდა ესეიგი ფაილი წარმატებით აიტვირთა).

წარმატებები ;)

ამოცანა 1. სტუდენტების ქულები (60 ქულა)

თქვენი მიზანია დაწეროთ StudRecords კლასი, რომელიც დაგვეხმარება სტუდენტების ნიშნების ორგანიზებაში. კლასს უნდა ჰქონდეს კონსტრუქტორი და 4 public მეთოდი.



public StudRecords(String filename)

კონსტრუქტორს გადაეცემა ფაილის სახელი სტრინგის სახით. თქვენ ეს ფაილი უნდა წაიკითხოთ და მასში არსებული ინფორმაცია თქვენთვის სასურველი სტრუქტურის საშუალებით შეინახოთ. ფაილის თითოეულ ხაზში არის ინფორმაცია იმის შესახებ თუ რა ნიშანი მიიღო სტუდენტმა ამა თუ იმ კურსში. ეს ინფორმაცია ჩაწერილია შემდეგი ფორმატით:

student name: სტუდენტის სახელი, **course name:** კურსის სახელი, **mark:** ნიშანი

ინფორმაცია გამოყოფილია სიმბოლოებით ': ' და ', '. ყოველი ': ' და ', ' სიმბოლოს შემდეგ არის მუსტად ერთი სფეისი (პარი). სტუდენტის სახელის მაგივრად ცხადია შეიძლება იყოს ნებისმიერი სტრინგი (მათ შორის სფეისების შემცველიც), კურსის სახელის მაგივრად შეიძლება იყოს ნებისმიერი სტრინგი, ნიშანის მაგივრად შეიძლება იყოს ნებისმიერი მთელი რიცხვი 0-დან 100-მდე (ჩათვლით). ჩათვალით, რომ სტრინგები სტუდენტის სახელი და კურსის სახელი არ შეიცავენ სიმბოლოებს ', ' და ': '. ფაილი ერთს ან მეტ მსგავს ხაზს შეიძლება შეიცავდეს. ფაილის მაგალითი:

student name: Joseph Stalin, course name: Marxism, mark: 51

student name: Adolf Hitler, course name: Introduction to painting, mark: 88

student name: Benito Mussolini, course name: Journalism, mark: 45

შეგიძლიათ ჩათვალით, რომ ერთი სტუდენტი ერთ კურსში მხოლოდ ერთ ნიშანს მიიღებდა. ერთსა და იმავე სტუდენტს შეიძლება ბევრი ნიშანი ჰქაედეს სხვადასხვა კურსებში. ასევე, ერთი და იგივე კურსი შეიძლება ბევრ სხვადასხვა სტუდენტს გაეუღო.

public int getStudentMarkInCourse(String stud, String course)

getStudentMarkInCourse მეთოდს გადაეცემა ორი სტრინგი და აბრუნებს ინტს. პირველი სტრინგი არის სტუდენტის სახელი, მეორე სტრინგი კურსის სახელი. მეთოდმა უნდა დააბრუნოს სტუდენტმა კურსში რა ნიშანი მიიღო. თუკი ან მსგავსი სტუდენტი ვერ მოიძებნა ან მსგავს სტუდენტს ეს კონკრეტული კურსი არ აქვს გავლილი მეთოდმა უნდა დააბრუნოს -1.

public double getStudentAverage(String stud)

getStudentAverage მეთოდს გადაეცემა სტუდენტის სახელი და უნდა დააბრუნოს ამ სტუდენტის საშუალო ქულა. ანუ უნდა დაითვალოთ და დააბრუნოთ სტუდენტის მიერ გავლილი ყველა კურსის საშუალო ქულა.

public Iterator<String> getStudentsWithMark(String course, int mark)

getStudentsWithMark მეთოდს გადაეცემა კურსის სახელი და ნიშანი და უნდა დააბრუნოს ყველა სტუდენტის სია, რომლებმაც მოცემულ კურსში მოცემული ნიშანი მიიღეს. სტუდენტების სია უნდა დაბრუნდეს სტრინგების იტერატორის საშუალებით.

public Map<String, ArrayList<String>> getCoursesWithMark(int mark)

getCoursesWithMark მეთოდს გადაეცემა ნიშანი და უნდა დააბრუნოს ყველა სტუდენტისა და კურსის წყვილები სადაც ეს კონკრეტული ნიშანი დაიწერა. მეთოდი აბრუნებს მეფს რომლის key არის სტუდენტის სახელი, ხოლო value კი იმ კურსების ArrayList, რომლებშიც ამ სტუდენტმა mark ნიშანი მიიღო. გაითვალისწინეთ, რომ მეფში არსებული არც ერთი ArrayList არ უნდა იყოს ცარიელი. თუკი სტუდენტს არ მიუღია mark ნიშანი არც ერთ საგანში, მაშინ ის მეფში საერთოდ არ უნდა ფიგურირებდეს. თუკი არც ერთ სტუდენტს არ მიუღია ნიშანი mark, მაშინ უნდა დააბრუნოთ ცარიელი მეფი (და არა null).



კოდი დაწერეთ **StudRecords.java** ფაილში. თუკი დამატებითი კლასის შექმნა
დაგჭირდებათ ჩაწერეთ იმავე ფაილში.

ამოცანა 2. რას დაბეჭდავს? (20 ქულა)

თქვენი ამოცანაა დაწეროთ რას დაბეჭდავს **Print.java** ფაილში მოცემული პროგრამები. თუკი
შეცდომა მოხდება კომპილაციის ან გაშვების შემდეგ მიუთითეთ რა სახის შეცდომაა.
პასუხი დაწერეთ **Print.java** ფაილში.



ამოცანა 3. რიცხვების შეკრება (25 ქულა)

თქვენი ამოცანაა დაწეროთ მეთოდი რომლის საშუალებითაც შევძლებთ რაგინდ დიდი მთელი ღაღებითი რიცხვების შეკრებას. დიდი რიცხვების მარტივად წარმოდგენა მასივის საშუალებით არის შესაძლებელი. მასივის i -ურ ელემენტში რიცხვის i -ური ციფრი შეგვიძლია რომ ჩავწეროთ. მაგალითად რიცხვი 123 შეგვიძლია წარმოვადგინოთ 3 სიგრძის `int`-ების მასივის სახით სადაც ნულოვანი ელემენტი იქნება 1, პირველი ელემენტი იქნება 2 ხოლო მეორე ელემენტი იქნება 3. დაწერეთ მეთოდი `addBigInteger` რომელსაც გადაეცემა ორი `int`-ების მასივი(ანუ ორი რიცხვი) და რომელიც აბრუნებს ასევე `int`-ების მასივს - ამ ორი რიცხვის შეკრებით მიღებულ შედეგს. მაგალითად თუკი მეთოდს გადავეცით მასივები {1,2,3} და {9,9,9,0} მაშინ მეთოდმა უნდა დააბრუნოს 5 სიგრძის მასივი {1,0,1,1,3}

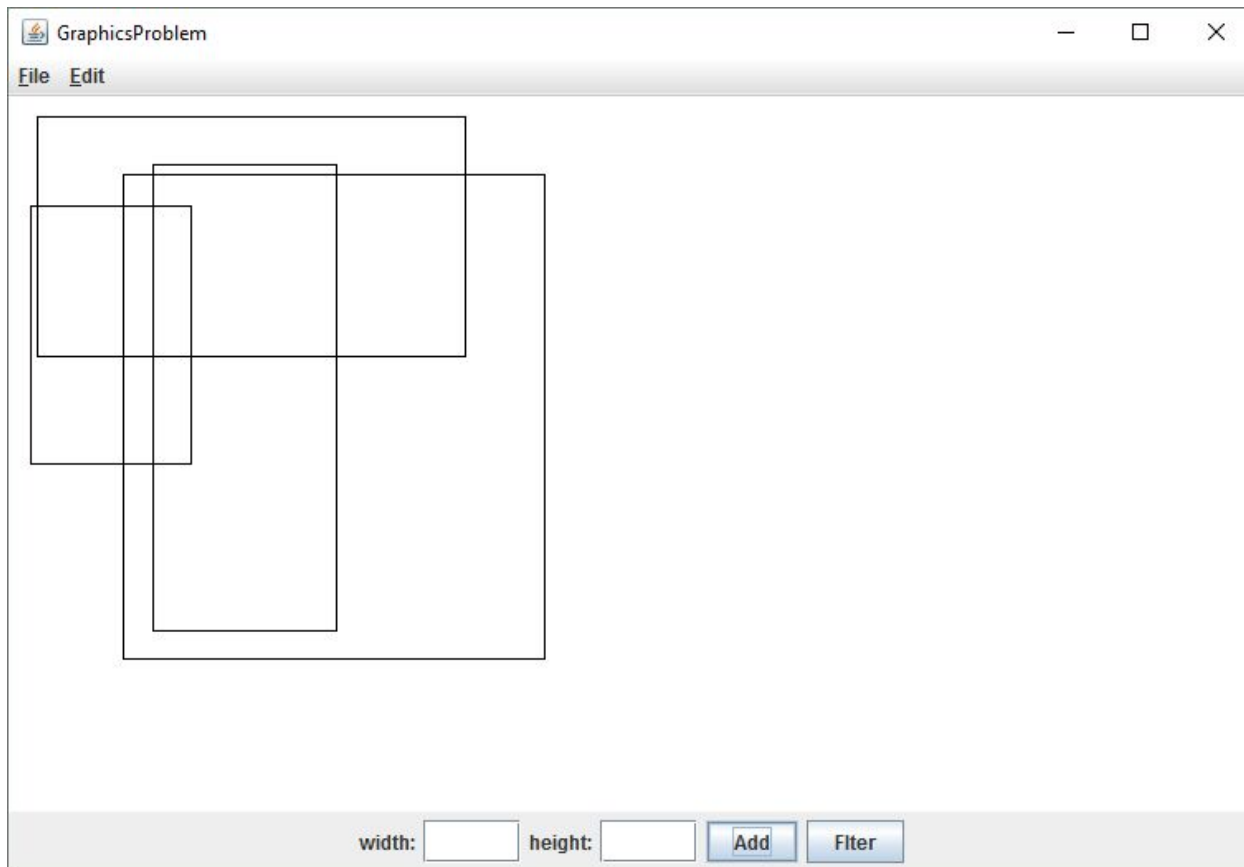
```
private int[] addBigInteger(int[] a, int[] b)
```



დაწერეთ შესაბამისი კოდი `addBigInteger.java` ფაილში.

ამოცანა 4. მართკუთხედების ფილგრი (50 ქულა)

თქვენი ამონცანაა შექმნათ სურათზე ნაჩვენები ვიზუალის მქონე გრაფიკული პროგრამა.



გრაფიკულ პროგრამას ქვედა ნაწილში აქვს ორი ტექსტური ველი **width** და **height**, რომლებიც განკუთვნილია მართკუთხედის სიგანის და სიმაღლის შესაყვანად. გარდა ამისა აქვს ორი ღილაკი **Add** და **Filter**.

Add ღილაკზე დაჭერით ფანჯრის შიგნით შემთხვევით წერტილში უნდა ემატებოდეს **width** და **height** ველებში შეყვანილი ზომების მქონე მართკუთხედი. თუ **Add** ღილაკზე დაჭერის მომენტში რომელიმე ველი ცარიელია, პროგრამამ არაფერი არ უნდა მოიმოქმედოს.

Filter ღილაკზე დაჭერით უნდა მოხდეს მართკუთხედების გაფილტვრა, ანუ გრაფიკულ პროგრამაზე უნდა დარჩეს მხოლოდ ისეთი **GRect**-ები რომელთა სიგანე ნაკლებია **width** ველში შეყვანილ მნიშვნელობაზე, ხოლო სიმაღლე ნაკლებია **height** ველში შეყვანილ მნიშვნელობაზე. თუ **Filter** ღილაკზე დაჭერის მომენტში რომელიმე (**width** ან **height**) ველი ცარიელია, მაშინ პროგრამამ ფილტრაცია უნდა გააუქმოს და ფანჯარაზე დაიხატოს ყველა მართკუთხედი რაც კი დამატებულია პროგრამის გაშვების შემდეგ.

ჩათვალით რომ მომხმარებელს **width** და **height** ველებში მხოლოდ რიხვითი მნიშვნელობები შეჰყავს. ფანჯრის ზომის ცვლილებაზე **GRect** ების ადგილმდებარეობა უცვლელი უნდა დარჩეს, ხოლო ზომები პროპორციულად უნდა შემცირდეს, ანუ თუ ფანჯრის ზომა 20%-ით შემცირდა მაშინ ყველა მართკუთხედის ზომაც 20%-ით უნდა შემცირდეს.

პროგრამის კოდი დაწერეთ **GRectFilter.java** ფაილში. თუკი დამატებითი კლასის შექმნა დაგჭირდებათ ჩაწერეთ იმავე ფაილში.

ამოცანა 5. გარდაქმნა (25 ქულა)

თქვენი ამოცანაა დაწეროთ **blowup** მეთოდის რეალიზაცია, რომელსაც გადაეცემა სკრინგი და აბრუნებს ამ სკრინგის გარდაქმნილ ვერსიას. გარდაქმნა უნდა იყოს შემდეგი სახის: სკრინგში ყოველი ციფრი (0-9) უნდა შეიცვალოს ამავე რაოდენობის მის მარჯვნივ მყოფი სიმბოლოთი. თუ



ციფრის მარჯვნივ არფერი არ არის (ეს ხდება როცა ციფრი სტრინგის ბოლოშია), მაშინ ეს ციფრი უბრალოდ უნდა წაიშალოს.

მაგალითად:

1. "a3tx2z" გარდაიქმნება "attttxzzz" - ში.

2. "12x" გარდაიქმნება "2xxx" - ში.

დაწერეთ Blowup.java ფაილში არსებული public String blowup (String s) ფუნქციის რეალიზაცია, რომელიც აბრუნებს გემოთ აღწერილი წესების მიხედვით გარდაქმნილ S სტრინგს.