



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 „დესკოპზე“ საქალაქოში Exam materials

საგამოცდო საკითხების ფორმა ვარიანტი # 2

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:			



1. დესკოპზე შექმენით *.java გაფართოების ფაილი, დაარქვით სახელად თქვენი უნივერსიტეტის მეილის პრეფიქსი – ვარიანტის ნომერი.

მაგალითად: nbarb14-2.java სადაც nbarb14 არის მეილის პრეფიქსი, ხოლო 2 ვარიანტის ნომერი.

2. გახსენით ფაილი რომელიმე ტექსტური რედაქტორის საშუალებით, მაგალითად Notepad++, პირველ ხაზზე დაწერეთ თქვენი სახელი და გვარი, შემდეგ კი ამოცანების ამოხსნები.
არ გამოგრჩეთ ფაილის დასეივება.

ამ წესების არ დაცვის შემთხვევაში შესაძლოა თქვენი ნაშრომი დაიკარგოს და შეფასება ვერ მიიღოთ.

ამოცანა 1. რას დაბეჭდავს (10 ქულა).



მოცემული გაქვთ შემდეგი პროგრამა თქვენი ამოცანაა დაადგინოთ თუ რას დაბეჭდავს პროგრამა. თუკი პროგრამის შესრულებისას მოხდება შეცდომა მიუთითეთ რა სახის შეცდომა მოხდება და რის გამო.

```
public void run() {  
    int arr[] = new int[100];  
  
    for (int i=0; i<10; ) {  
        for (int j=0; j<5; j++) {  
            i++;  
            arr[i+j] = i+2*j;  
        }  
    }  
  
    for (int i=0; i<10; i++) {  
        print(arr[i] + " ");  
    }  
}
```

ამოცანა 2. გექსგის გაშიფრვა (15 ქულა).

წარმოიდგინეთ შიფრაცია, სადაც თითოეული სიმბოლოს ვშიფრავთ, ანუ ვცვლით კონკრეტული, წინასწარ განსაზღვრული სიმბოლოთი. ჩავთვალოთ რომ შეხება გვაქვს მხოლოდ ლათინური ანბანის პაგარა ასოებთან 'a' -დან 'z' -მდე. ანუ დასაშიფრი გექსგიც და დაშიფრული გექსიც მხოლოდ ამ სიმბოლოებისგან შედგება. შიფრი მოცემულია 26 სიმბოლოიანი სგრინგის სახით, სადაც პირველი სიმბოლო აღნიშნავს თუ რითი უნდა შეიცვალოს 'a', მეორე აღნიშნავს თუ რითი



უნდა შეიცვალოს სიმბოლო 'b' და ა.შ. ბოლო ოცდაექვსე სიმბოლო აღნიშნავს თუ რა სიმბოლოთი უნდა შეიცვალოს სიმბოლო 'z'. ჩათვალეთ რომ შივრში ოცდაექვსივე სიმბოლო ერთმანეთისგან განსხვავდება და ასევე წარმოადგენს 'a' - 'z' ალფაბეტს, ოღონდ გადალაგებული სახით.

მოცემული გაქვთ შივრი და უკვე დაშიფრული ტექსტი. თქვენი ამოცანაა ეს ტექსტი გაშიფროთ. ამისათვის დაწერეთ მეთოდი `decode` რომელსაც გადაეცემა უკვე დაშიფრული ტექსტი (`String`) და შივრი (`String`) და რომელიც აბრუნებს გაშიფრულ ტექსტს (`String`).

```
private String decode(String encodedText, String cipher){  
}
```

ამოცანა 3. მამაშვილობა (40 ქულა)

მოცემული გაქვთ ფაილი, სადაც თითოეულ ხაზში წერია სუფისით გამოყოფილი ორი სახელი, პირველი არის შვილის სახელი, ხოლო მეორე მისი მამის სახელი. ჩათვალეთ, რომ სახელები უნიკალურია და ორი განსხვავებული ადამიანი ერთი და იგივე სახელით არ შეგხვდებათ.

დაწერეთ `getCousins` მეთოდი რომელსაც გადაეცემა, მემოთ ნახსენები ფაილის სახელი და ფაილში არსებული ერთერთი ადამიანის სახელი, ხოლო აბრუნებს გადაცემული ადამიანის ბიძაშვილების სიას.

```
private ArrayList<String> getCousins(String fileName, String name) {
```



თუ ბიძაშვილების შესახებ არ მოიძებნა ინფორმაცია, მეთოდმა უნდა დააბრუნოს ცარიელი სია.

ამოცანა 4. ჯოკერი (35 ქულა)

თქვენი ამოცანაა გაარკვიოთ ჯოკერში საშუალოდ რამდენ ღარიგებაში ერთხელ ხვდება მოთამაშის ხელში ორი ჯოკერი.

ჯოკერი ბანქოს თამაშია და მისი თამაშისას 36 ბანქო მონაწილეობს, რომელთაგან 2 ჯოკერია.

თამაშობს 4 მოთამაშე და ბანქო ამ ოთხ მოთამაშეზე რიგდება თითო-თითოდ, ისე, რომ საბოლოო ჯამში თითოეულ მოთამაშეს 9-9 ბანქო ჰყავს ხელში.

მოახდინეთ N ცალი ღარიგების სიმულაცია და დაადგინეთ საშუალოდ რამდენ ღარიგებაში ერთხელ მოხვდება პირველი მოთამაშის ხელში ორი ჯოკერი. ამისათვის უნდა დაწეროთ `simulateJoker` მეთოდი რომელსაც გადაეცემა მთელი რიცხვი N და აბრუნება ნამდვილ რიცხვს.

```
private double simulateJoker(int n){
```



ამოცანა 5. რამდენად სწორია სუდოკუ? (35 ქულა)

მოცემული გაქვთ შევსებული სუდოკუს დაფა. თქვენი ამოცანაა დაადგინოთ არის თუ არა დაფა სწორად შევსებული. უნდა დაწეროთ მეთოდი `isSudokuRight` რომელსაც გადაეცემა მთელი რიხების ორგანზომილებიანი, `9x9`-ზე, მაგრიცა და აბრუნებს `true`-ს იმ შემთხვევაში თუ სუდოკუ სწორად არის შევსებული და `false`-ს წინააღმდეგ შემთხვევაში.

```
private boolean isSudokuRight(int [][] sudoku){
```

მიუხედავად იმისა რომ საქართველოში სუდოკუ საკმაოდ გავრცელებული თამაშია მაინც გაგაცნობთ თამაშის წესებს. თქვენს წინაშე წარმოდგენილია სუდოკუს სტანდარტული სათამაშო დაფა. თამაშის მიზანია მთლიანად დაფა შეივსოს ისე რომ ერთ რიგზე შეეუღლად და თარაზულად არ განმეორდეს ერთიდაიგივე რიცხვი, ასევე არც ერთ `3x3`-ზე კვადრაგში(სულ 9 ასეთი კვადრაგია) არ უნდა განმეორდეს ერთიდაიგივე რიცხვი.



7			5	8	3			6
		6			1	4		5
	5	2			6		8	3
3			2			9	5	8
5				7	8		6	
6	4	8		1		3		
	6		8		2	5		
		3	1	5				
2	1	5	6					

შემდეგ სურათზე ლურჯად ნაჩვენებია როგორ ხდება სულოკუს შევსება.
ნებისმიერ თარაბულად მდებარე ხაზზე არ უნდა განმეორდეს ერთიდაიგივე რიცხვი.

7	9	4	5	8	3	2	1	6
		6			1	4		5
	5	2			6		8	3
3			2			9	5	8
5				7	8		6	
6	4	8		1		3		
	6		8		2	5		
		3	1	5				
2	1	5	6					

შემდეგ სურათზე ლურჯად ნაჩვენებია როგორ ხდება სულოკუს შევსება.
ნებისმიერ შევსებად მდებარე ხაზზე არ უნდა განმეორდეს ერთიდაიგივე რიცხვი.

7	9		5	8	3			6
	3	6			1	4		5
	5	2			6		8	3
3	7		2			9	5	8
5	2			7	8		6	
6	4	8		1		3		
	6		8		2	5		
	8	3	1	5				
2	1	5	6					

შემდეგ სურათზე ლურჯად ნაჩვენებია როგორ ხდება სულოკუს შევსება.
როგორც ხედავთ სულოკუს დაფა დაყოფილია 9 კვადრატად, ასევე არცერთ კვადრატში არ უნდა განმეორდეს ერთიდაიგივე რიცხვი.



7	9	4	5	8	3			6
8	3	6			1	4		5
1	5	2			6		8	3
3			2			9	5	8
5				7	8		6	
6	4	8		1		3		
	6		8		2	5		
		3	1	5		7	2	
2	1	5	6				3	

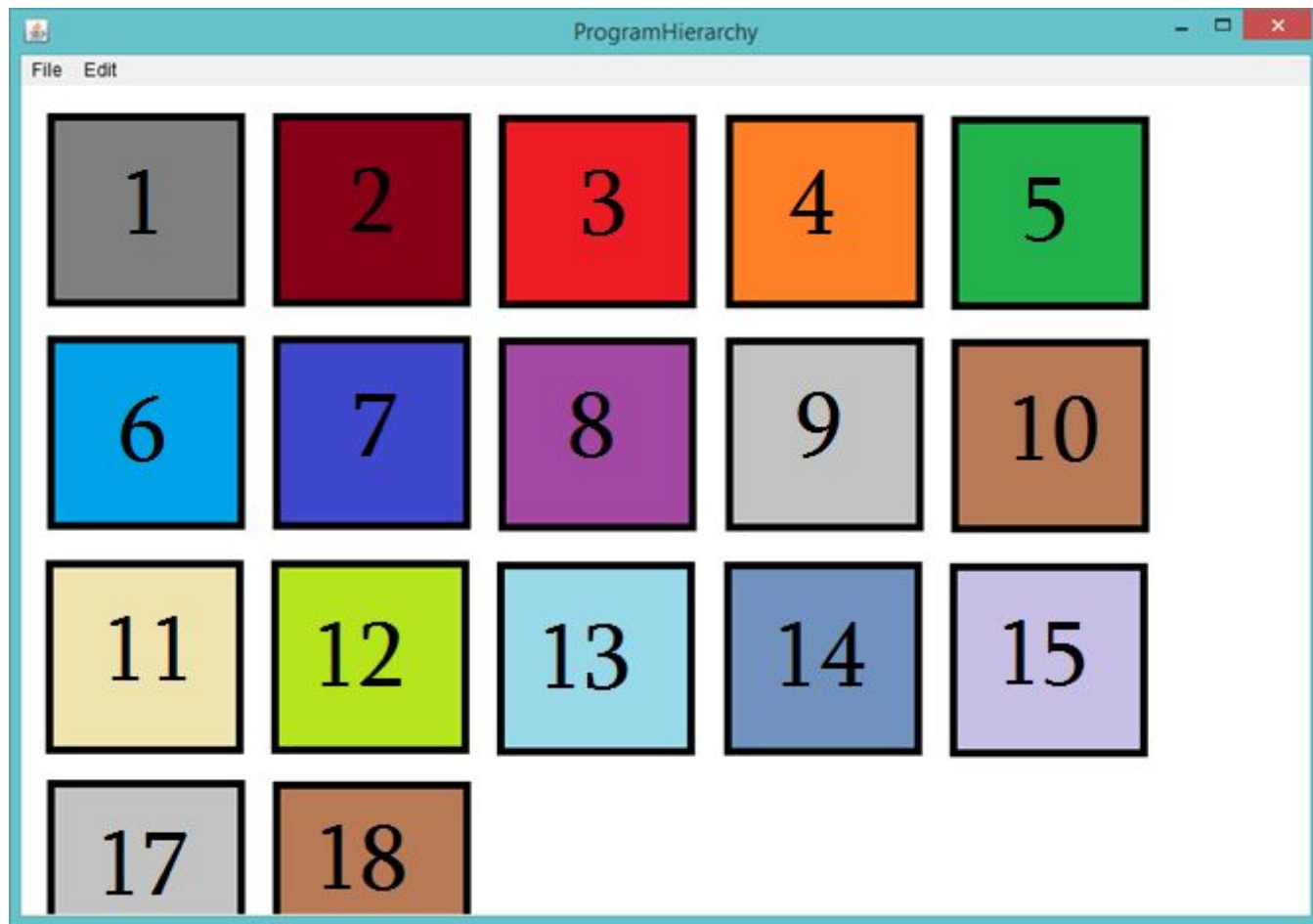
იმ შემთხვევაში თუ ყველაფერს სწორად გავაკეთებთ მივიღებთ მსგავს დაფას სადაც ყველაფერი მუსტი გათვლით არის შესრულებული.

7	9	4	5	8	3	2	1	6
8	3	6	7	2	1	4	9	5
1	5	2	4	9	6	7	8	3
3	7	1	2	6	4	9	5	8
5	2	9	3	7	8	1	6	4
6	4	8	9	1	5	3	2	7
9	6	7	8	3	2	5	4	1
4	8	3	1	5	9	6	7	2
2	1	5	6	4	7	8	3	9

თამაში სულოკუს ახსნა აღებულია crack.ge-დან.

ამოცანა 6. Grid Gallery (45 ქულა)

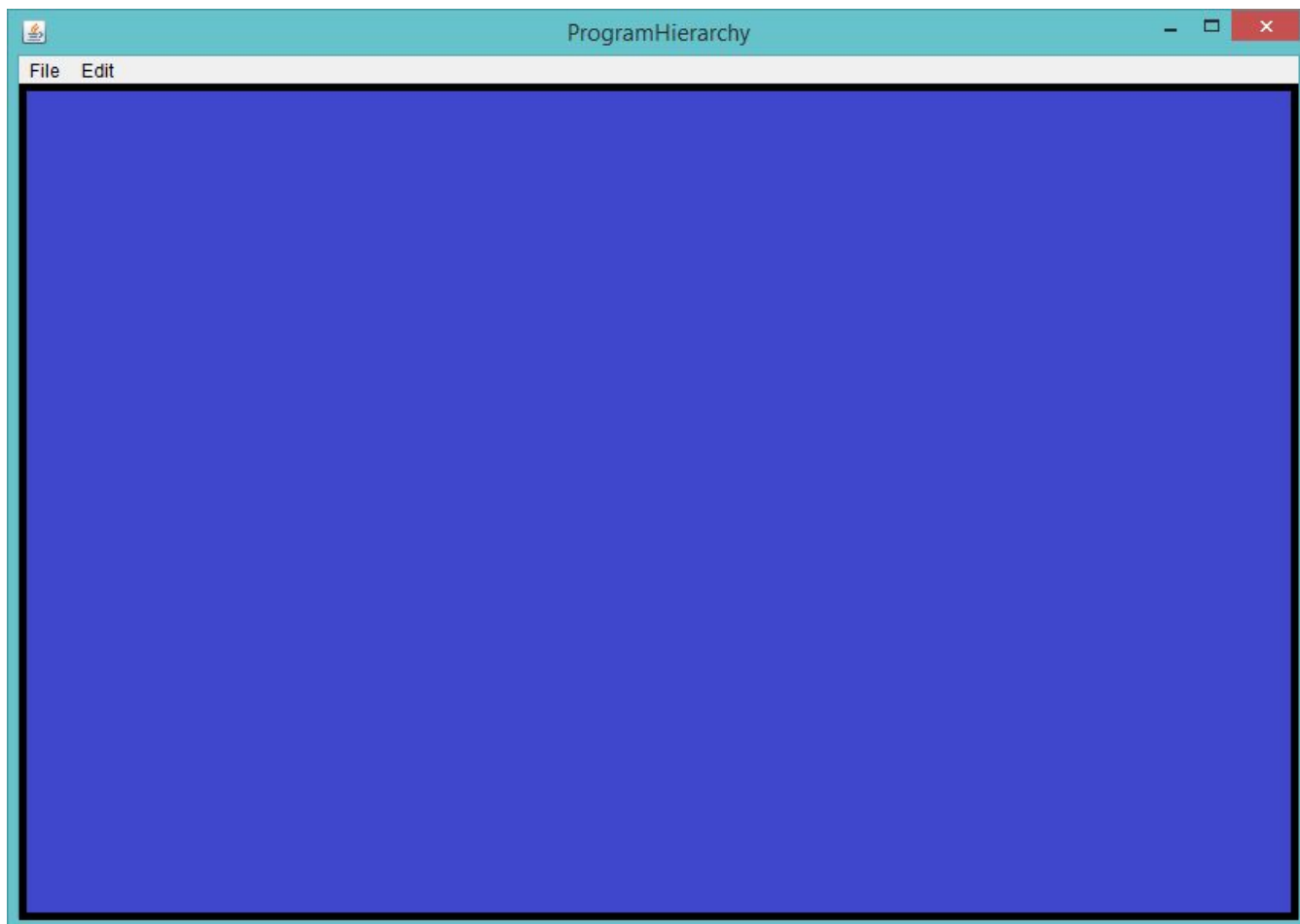
პირველ სქრინშოტზე გამოსახულია გრაფიკული პროგრამა GridGallery, რომელიც საშუალებას გვაძლევს დავათვალიეროთ სურათები. ფერადი კვადრატები შეესაბამება ჩაგვიროთულ სურათებს.



სქრინშოტი 1

რომელიმე სურათზე კილიკით, სურათი ღიდება მთლიან ფანჯარაში.

მაგალითად თუ ღაჭაქლიქებთ მეშვიდე სურათზე, პროგრამა გააღიღებს სურათს როგორც მეორე სქრინშოტზეა ნაჩვენები.



სქრინშოტი 2

გადიდებულ სურათზე კლიკს რამდენიმე დანიშნულება აქვს, თუ კლიკი მოხდება ფანჯრის მარცხენა მესამედში, ფორმაზე უნდა გამოჩნდეს გადიდებული ერთი ნაკლები ინდექსის მქონდე სურათი (ამ შემთხვევაში მეექვსე სურათი). მარჯვენა მესამედში დაკლიკების შემთხვევაში პირიქით, ერთი დიდი ინდექსის მქონდე სურათი. ხოლო შუა მესამედში კლიკზე, სურათი უნდა დაპაგარავდეს და პროგრამა დაუბრუნდეს პირველ სქრინშოტზე ნაჩვენებ მდგომარეობას.

თქვენი ამოცანაა დაწეროთ GridGallery პროგრამის რეალიზაცია, პროგრამამ უნდა გამოაჩინოს images სიაში არსებული სურათები გრიდის სახით. გრიდის ჰორიზონტალში სურათების რაოდენობა უნდა განისაზღვროს ფანჯრის სიგრძის მიხედვით.

გაითვალისწინეთ ის ფაქტი რომ რეალური სურათების ზომა შეიძლება არ იყოს კვადრატული. გრიდში ყველა სურათს კვადრატული ფორმა აქვს, ხოლო გადიდებულ სურათს კი ფანჯრის ზომების შესაბამისი ფორმა.

```
public class PhotoGallery extends GraphicsProgram {
```



```
public static final int IMG_H = 200;  
public static final int IMG_DST = 20;  
  
private ArrayList<String> images = ...;  
  
public void run() {  
  
}  
}
```

images
IMG_H
IMG_DST

მასივი უკვე შევსებულია პროგრამაში ჩასატვირთი სურათების სახელებით .
გრიდში სურათის სიმაღლე და სიგანე.
გრიდში სურათებს შორის მანძილი