***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 |  |
| ლეპტოპი/პლანშეტი |  |

**\* გამოცდის ჩატარების წესი იხილეთ ,,დესკტოპზე“ საქაღალდეში Exam materials**

***საგამოცდო საკითხების ფორმა***

***ვარიანტი # 1***

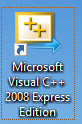
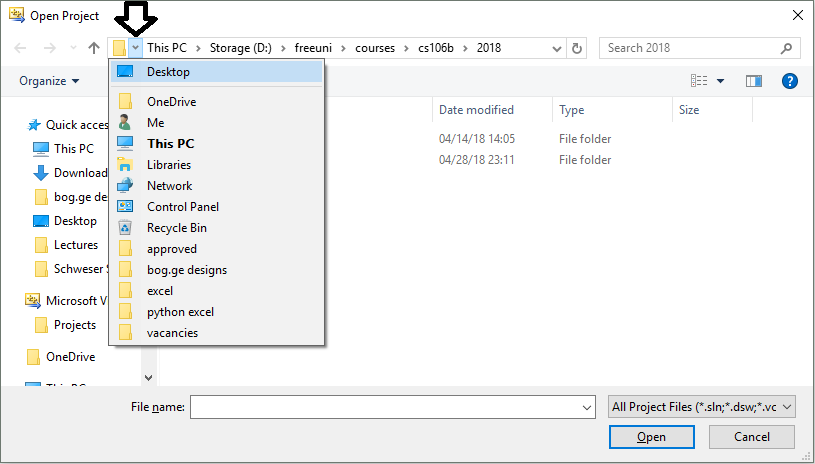
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 2.5 საათი | | |
| ***მაქსიმალური ქულა*** | 180 | | |
| ***სტუდენტის სახელი და გვარი:*** | | | |

სახელი: ქულა:

ფინალური გამოცდა

პროგრამირების აბსტრაქციებში

**შეასრულეთ შემდეგი ინსტრუქციები, წინააღმდეგ შემთხვევაში შესაძლოა თქვენი ნაშრომი არ შეფასდეს.**

1. **ჩამოტვირთეთ საგამოცდო საკითხები და თქვენი ვარიანტის შესაბამისი პროექტის არქივი midterm1.zip თქვენს დესკტოპზე.**
2. **ამოაარქივეთ პროექტი, მაუსის მენიუ ღილაკით დააჭირეთ არქივზე და აირჩიეთ Extract here.**
3. **დესკტოპზე იპოვეთ Microsoft Visual C++ 2008 ის გამშვები ფაილი და გაუშვით.  
   **
4. **File მენიუდან აირჩიეთ Open > Project/Solution, გაიხსნება ფანჯარა სადაც დააჭირეთ სურათზე აღნიშნულ ისარს და აირჩიეთ Desktop,   
   **
5. **შემდეგ აირჩიეთ ამოარქივებული ფოლდერში project\Blank Project.sln ფაილი და დააჭირეთ Open ს.**
6. **თუ Visual Studio ში არ გიჩანთ ფაილები, View მენიუში მონიშნეთ Solution Explorer პუნქტი**
7. **თითოეული ამოცანისთვის დაგხვდებათ შესაბამისი \*.cpp ფაილი, სადაც უნდა დაწეროთ თქვენი ამოხსნა. main.cpp ფაილში წერია ამოცანების ტესტებზე შემოწმება და მისი გაშვებით კონსოლში გამოვა ინფორმაცია თუ რომელ ტესტებზე გაიარა\ჩაიჭრა თქვენმა ამოხსნამ. ფუნქციის პროტოტიპის შეცლა არ შეიძლება. შეგიძლიათ შემოიტანოთ დამხმარე ფუნქციები. არ გაქვთ გლობალური ცვლადების გამოყენების უფება.**
8. **ამოხსნის დასრულების შემდეგ დაარქივეთ „Desktop\midterm 2\problems” ფოლდერი და არქივს დაარქვით სახელად თქვენი მეილის პრეფიქსი, მაგალითად gboch10.zip**
9. **ვებ ბრაუზერში გახსენით მისამართი** [http://192.168.210.5](http://192.168.210.5/) **და ატვირთეთ არქივი.**

**საყურადღებო:**

1. კოდი გაუშვით f5 ღილაკზე დაჭერით ან მწვანე Play ღილაკზე დაჭერით, წინააღმდეგ შემთხვევაში კონსოლის ფორმა არ გამოჩნდება.
2. Fatal error LNK1168 cannot open - ამ შეცდომის შემთხვევაში ctrl + shift + esc ღილაკებით გახსენით task manager და მოკალით ყველა Blank Project პროცესი
3. თუ კოდში ცვლილება შეგაქვთ მაგრამ გაშვებისას არანაირად არი ისახება, დარწმუნდით რომ წინა წარმატებული ბილდი არ ეშვება. F7 ით ცადეთ დაკომპილირება.
4. თუ გსურთ ხაზების ნუმერაციის გამოჩენა, გახსენით Tools > Options მენიუ, გადადით Text Editor > All Languages > General და მონიშნეთ Line numbers.

**ამოცანა 1. მონოტონური გზა (100 ქულა)**

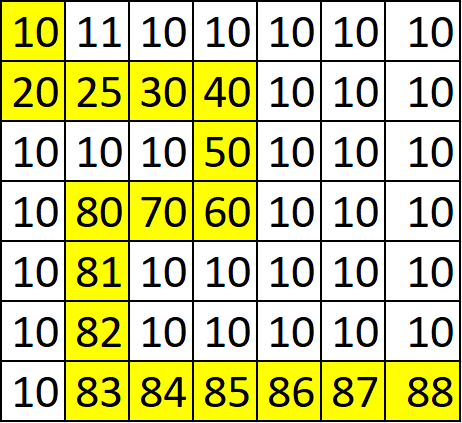
გრიდის სახით მოცემული გაქვთ თამაშის დაფა სადაც უჯრებში წერია რიცხვები. სათამაშო ფიგურა იმყოფება ზედა მარცხენა კუთხეში და თქვენი მიზანია რაც შეიძლება ნაკლები უჯრის გავლით მოხვდეთ ქვედა მარჯვენა უჯრაზე, ისე რომ არ დაარღვიოთ ორი გადაადგილების წესი:

1. გადასვლა შეიძლება მხოლოდ გვერდითა ან ქვედა მეზობელ უჯრაზე, ანუ A დან B უჯრაზე გადასვლა შესაძლებელია თუ მათ ან ქვედა გვერდი აქვთ საერთო, ან რომელიმე გვერდითა
2. საწყისი წერტლიდან საბოლოო წერილამდე განვლილი უჯრების(ზუსტად იმ რიგითობით, რა რიგითობითაც მოხდა მათი გავლა) რიცხვითი მნიშვნელობები უნდა გვაძლევდეს მკაცრად მონოტონურ მიმდერვრობას, ანუ მიმდევრობა უნდა იყოს მკაცრად კლებადი ან ზრდადი.

დაწერეთ getShortestPath ფუქნციის რეალიზაცია, რომელსაც Grid<int> ის სახით გადაეცემა სათამაშო დაფის შესახებ ინფორმაცია და აბრუნებს ნაპოვნი უმოკლესი გზის სიგრძეს, ანუ რამდენ უჯრაზეც დაჭირდა გადასვლა. თუ ასეთი გზა საერთოდ არ არსებობს, ფუქნციამ უნდა დააბრუნოს -1.

გრიდის მაქსიმალური ზომა შესაძლებელია იყოს 200x200 ზე, თქვენი ამოხსნა არ უნდა იყოს O(n\*n\*n) ზე უფრო ცუდი. ტესტების 20% ში, გრიდის ზომა იქნება 50 ზე ნაკლები.

მაგალითად ქვემოთ მოყვანილ დაფაზე უმოკლესი გზა არის 16.



ტესტები შეგიძლიათ იხილოთ Blank Project\path ფოლდერში.

**ამოცანა 2. აწონვის თამაში (80 ქულა)**

მოცემულია თეფშებიანი სასწორი, სხვადასხვა წონის გირები და ნივთები. გაქვთ ზუსტად ორი ნივთის არჩევის უფლება, თუ ახერხებთ არჩეული ნივთების წონაში დარწმუნებას, ანუ ახერხებთ მათ ცალცალკე აწონვას თეფშებიან სასწორზე, მაშინ ეს ნივთები თქვენია. თქვენი ამოცანაა შერჩეული ნივთების ჯამური წონა რაც შეიძლება მაქსიმალური იყოს. გაითალისწინეთ ისიც, რომ შერჩეული ნივთიდან რომელიმეს აწონვის შემდეგ, ამ ნივთის გამოყენებაც შეიძლება გირის როლში.

პროცესის წარმოსადგენად, ქვემოთ ჩამოწერია ბიჯები თუ როგორ უნდა განვითარდეს მოვლენები:

1. ირჩევთ თქვენთვის სასურველ ორ ნივთს
2. ცდილობთ აწონოთ ერთერთი ნივთი მხოლოდ გირების გამოყენებით
3. თუ ნივთის აწონვა მოახერხეთ, ცდილობთ აწონოთ მეორე ნივთი გირების და პირველი ნივთის გამოყენებით.
4. თუ ყველა ნივთის აწონვა მოხერხდა, ნივთები თქვენია, წინააღმდეგ შემთხვევაში ხელცარიელი რჩებით.

დაწერეთ maxWeight ფუნქციის რეალიზაცია, რომელსაც გადაეცემა გირების წონები weight ს ვექტორის სახით და საგნების წონები items ვექტორის სახით. ფუნქციამ უნდა დააბრუნოს მაქსიმუმ რა ჯამური წონის ნივთების დასაკუთრება არის შესაძლებელი.თუ ვერ ხერხდება ასეთი ორი ნივთის შერჩევა, ფუქნციამ უნდა დააბრუნოს 0.

ნივთების რაოდენობა არ აღემატება 15 ს, ხოლო გირების 10 ს. პროგრამას ერთ ტესტზე არ უნდა ჭირდებოდეს 1 წამზე მეტი დრო.