***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***

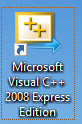
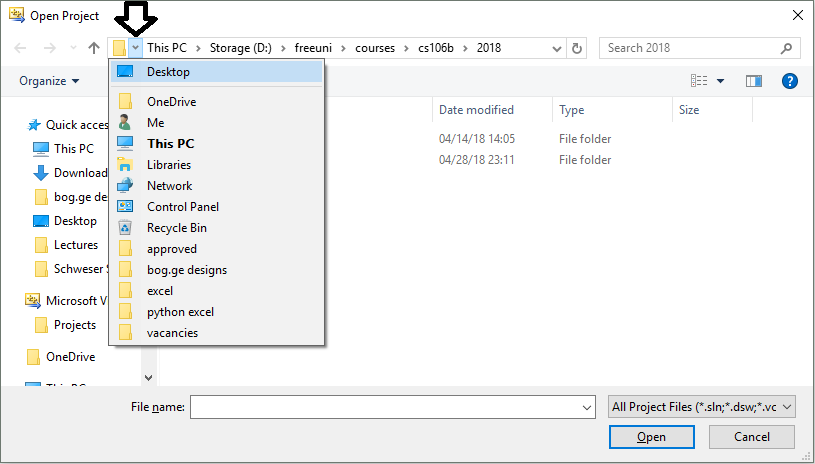
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 2 საათი | | |
| ***მაქსიმალური ქულა*** | 120 | | |
| ***სტუდენტის სახელი და გვარი:*** | | | |

სახელი: ქულა:

ფინალური გამოცდა

პროგრამირების აბსტრაქციებში

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

1. **ჩამოტვურთტეთ საგამოცდო საკითხები და თქვენი ვარიანტის შესაბამისი პროექტის არქივი 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. ფუნქციის პროტოტიპის შეცლა არ შეიძლება. შეგიძლიათ შემოიტანოთ დამხმარე ფუნქციები. არ გაქვთ გლობალური ცვლადების გამოყენების უფება.**
8. **main.cpp ფაილში წერია ამოცანების ტესტებზე შემოწმება. მისი გაშვებით ხდება ამოცანებში მოცემული ფუნქციების გამოძახება სხვადასხვა input-ით და დაბრუნებული პასუხის შემოწმება. კონსოლში გამოვა ინფორმაცია თუ რომელ ტესტებზე გაიარა\ჩაიჭრა თქვენმა ამოხსნამ.**
9. **ამოხსნის დასრულების შემდეგ დაარქივეთ „problems” ფოლდერი და არქივს დაარქვით სახელად თქვენი მეილის პრეფიქსი, მაგალითად gboch10.zip**
10. **ვებ ბრაუზერში გახსენით მისამართი** [http://192.168.210.5](http://192.168.210.5/) **და ატვირთეთ არქივი. არქივი დატოვეთ დესკტოპზე**

**საყურადღებო:**

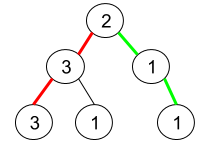
1. კოდი გაუშვით f5 ღილაკზე დაჭერით ან მწვანე Play ღილაკზე დაჭერით, წინააღმდეგ შემთხვევაში კონსოლის ფორმა არ გამოჩნდება.
2. Fatal error LNK1168 cannot open - ამ შეცდომის შემთხვევაში ctrl + shift + esc ღილაკებით გახსენით task manager და მოკალით ყველა Blank Project პროცესი
3. თუ კოდში ცვლილება შეგაქვთ მაგრამ 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.
5. თუ გაურკვეველი შეცდომა გაქვთ, ცადეთ მენიუდან Build > Clean Solution და Build > Clean Blank Project

**ამოცანა 1. (50 ქულა) ფსევდო-პალინდრომული გზა**

მოცემული გაქვთ არანულოვანი ორობითი ხე, რომელშიც წვეროების მნიშვნელობები 1 დან 9 მდე ციფრებია. გზას root წვეროდან ფოთლამდე უწოდებენ ფსევდო-პალინდრომულს, თუ ამ გზაში შემავალი წვეროების მნიშვნელობების პერმუტაციით შესაძლებელია პალინდრომის მიღება.

დააბრუნეთ ორობითი ძებნის ხეში ფსევდო-პალინდრომული გზების რაოდენობა.

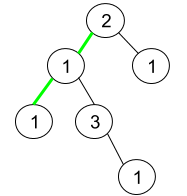
**მაგალითი 1:**



**პასუხი:**

ამ ხეში სულ სამი გზა არსებობს root წვეროდან ფოთლამდე, [2, 3, 3], [2, 3, 1] და [2, 1, 1]. ამ გზებიდან მხოლოდ ორია პალინდრომული, [2, 3, 3] და [2, 1, 1], რადგან შესაძლებელია ამ ციფრების ისეთ გადალაგება რომ პალინდრომი მივიღოთ: [3, 2, 3], [1, 2, 1]. ხოლო მესამე გზის შემთხვევაში ვერცერთი გადალაგებით ვერ მივიღებთ პალინდრომს.

**მაგალითი 2:**



**პასუხი:**

ამ შემთხვევაშიც სამი გზა არსებობს, [2, 1, 1], [2, 1, 3, 1] და [2, 1]. ამ სამი გზიდან მხოოდ [2, 1 ,1] ის გადალაგებით მიიღება პალინდრომი [1, 2, 1].

ხეში შეიძლება იყოს 10^5 წვერო და თქვენი ამოხსნა უნდა მუშაობდეს O(n) დროში. ტესტების მაგალითები შეგიძლიათ იხილოთ **project\Blank Project\palindromic** ფოლდერში.

**ამოცანა 2. (70 ქულა) ფორთოხალი**

მოცემულ გრიდში უჯრის მნიშნელობა შეიძლება იყოს ერთერთი ამ სამიდან:

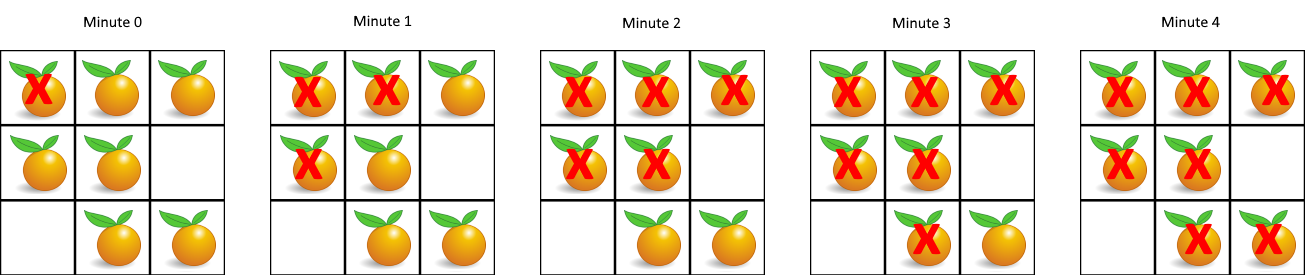
* 0 - რაც ნიშნავს რომ უჯრა ცარიელია
* 1 - რაც ნიშნავს რომ უჯრა დაკავებულია საღი ფორთოხლით
* 2 - რაც ნიშნავს რომ უჯრა დაკავებულია გაფუჭებული ფორთოხლით

ცნობილია რომ თავიდან მხოლდ ერთ უჯრაზე დევს გაფუჭებული ფორთოხალი. ყოველ წუთის გასვლის შემდეგ ფორთოხალი რომელიც მდებარეობს გაფუჭებული ფორთოხლის გვერდით(ჰორიზონტალურად ან ვერტიკალურად), ასევე ფუჭდება.

იპოვეთ მინიმუმ რამდენი წუთი უნდა გავიდეს იმისათვის რომ არცერთი საღი ფორთოხალი აღარ დარჩეს. თუ საღი ფორთოხლები ყოველთვის იარსებებს დააბრუნეთ -1. დაწერეთ orangesRotting ფუნქციის რეალიზაცია, რომელსაც გადაეცემა: მატრიცა - grid, მატრიცაში სვეტების რაოდენობა - w, სტრიქონების რაოდენობა h, გაფუჭებული ფორთოხლის კოორდინატები x,y. ფუნქცია აბრუნებს წუთების რაოდენობას.

int orangesRotting(int\*\* grid, int w, int h, int x, int y)

**მაგალითი 1:**



[[2,1,1],[1,1,0],[0,1,1]]

**პასუხი:** 4 წუთი

**მაგალითი 2:** [[2,1,1],[0,1,1],[1,0,1]]

**პასუხი:** -1

**მაგალითი 3:** [[0,2]]

**პასუხი:** 0

დაფის მაქსიმალური ზომა შეიძლება იყოს 1000x1000, თქვენი ალგორითმი უნდა მუშაობდეს O(n\*m) დროში, სადაც n და m დაფის ზომებია. დამატებითი მაგალითები შეგიძლიათ იხილოთ **project\Blank Project\orange**