***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*** | შ. ღვინეფაძე | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 3 საათი | | |
| ***მაქსიმალური ქულა*** | 180 | | |
| ***სტუდენტის სახელი და გვარი:*** | | | |

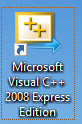
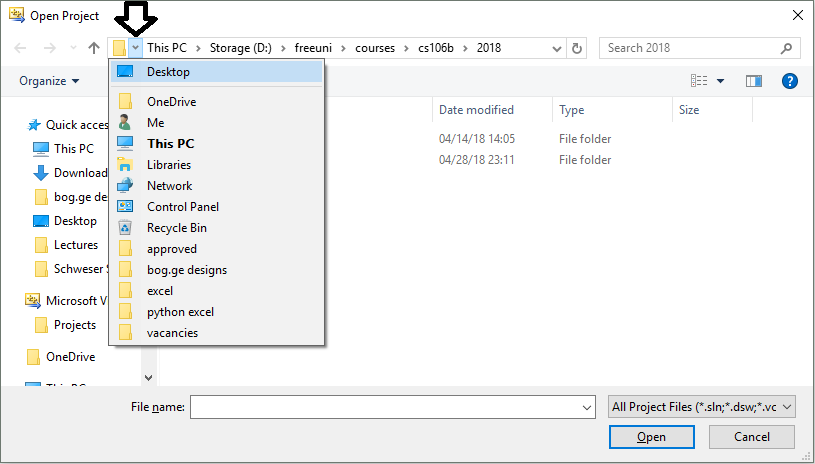
სახელი: ქულა:

შუალედური გამოცდა

პროგრამირების აბსტრაქციებში

2018, 5 მაისი 17:00 - 20:00

**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 „Deskto\midterm test\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. ხურდა (80 ქულა)**

თქვენი ამოცანაა მოლარეს დაეხმაროთ მომხმარებლისთვის ხურდის დაბრუნებაში. დაწერეთ giveChange ფუნქციის რეალიზაცია რომელსაც გადაეცემა სალაროში არსებული კუპიურების სია(cshBills), კლიენტის ხელთ არსებული კუპიურების სია( cstBills) და სერვისის ღირებულება რომლის საფასურსაც იხდის მომხმარებელი. ფუნქციამ უნდა დააბრუნოს true, თუ არსებობს ზუსტი ხურდის დაბრუნების საშუალება და false წინააღმდეგ შემთხვევაში. გაითვალისწინეთ ერთი და იგივე კუპიურა შესაძლოა რამდენიმე ცალი იყოს.

bool giveChange(Vector<int> &cshBills, Vector<int> &cstBills, int price);

ჯამურად სალაროში და მომხმარებლის ხელთ არსებული კუპიურების რაოდენობა არ აღემატება 24 ს.

პროგრამა შემოწმდება კომპიუტერზე რომელიც წამში ასრულებს დაახლოებით 109 ოპერაციას. ფუნქციას ერთ ტესტზე არ უნდა ჭირდებოდეს 1 წამზე მეტი დრო, წინააღმდეგ შემთხვევაში ტესტი არ ჩაითვლება.

ტესტების 50% - n < 15

ტესტების 25% - 15 <= n < 20

ტესტების 25% - 20 <= n < 24

main.c ფაილში შეგიძლიათ იხილოთ ტესტები და პასუხები.

**ამოცანა 2. წყვილები(100 ქულა)**

ტინდერის აპლიკაციში გეგმავენ დაწყვილების ფუნქციონალის დამატება. აპლიკაცია ლოკაციის მიხედვით შეარჩევს n(1 < n < 16) ადამიანს, თითოული მონაწილე მიუთითებს თუ რომელი მონაწილეები მოსწონს და ამის შემდეგ ავტომატურად უნდა მოხდეს მონაწილეების ისეთნაირი დაწყვილება რომლის დროსაც:

1. ნებისმიერი წყვილში ორივე ადამიანს მოსწონს ერთმანეთი
2. წყვილების რაოდენობა არის მაქსიმალური

თქვენ გევალებათ დაეხმაროთ ტინდერს მაქსიმალური დაწყვილების პოვნაში, ამისათვის უნდა დაწეროთ int maxPairs(Grid<bool> &likes)

ფუნქციის რეალიზაცია. ფუნქციას nxn-ზე გრიდის სახით გადაეცემა ინფორმაცია იმის სესახებ, თუ ვის ვინ მოსწონს. მონაწილეები გადანომრილები არიან 0 დან n-1 მდე, likes[i][j] არის true თუ i-ნომრის მქონე მონაწილეს მოსწონს j-ნომრის მქონე მონაწილე, წინააღმდეგ შემთხვევაში likes[i][j] იქნება false. ფუნქციამ უნდა დააბრუნოს მაქსიმალური წყვილების რაოდენობა. ამოცანა შეგიძლიათ ამოხსნათ ყველა შესაძლო ვარიანტის გადარჩევით, ეცადეთ ადრეულ ეტაპზევე გამორიცხოთ არავალიდური კომბინაციები, წინააღმდეგ შემთხვევაში თქვენი ამოხსნა შესაძლოა დროით შეზღუდვაში ჩაიჭრეს.

პროგრამა შემოწმდება კომპიუტერზე რომელიც წამში ასრულებს დაახლოებით 109 ოპერაციას. ფუნქციას ერთ ტესტზე არ უნდა ჭირდებოდეს 1 წამზე მეტი დრო, წინააღმდეგ შემთხვევაში ტესტი არ ჩაითვლება.

ტესტების 50% - n < 10

ტესტების 25% - 10 <= n < 14

ტესტების 25% - 14 <= n <= 16

პ.ს. თუ თქვენი ამოხსნა დროში არ იჭრება გამოცდის ტესტებზე, ჩათვალეთ რომ არც ჩემს ტესტებზე ჩაიჭრება დროში.