***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***

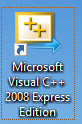
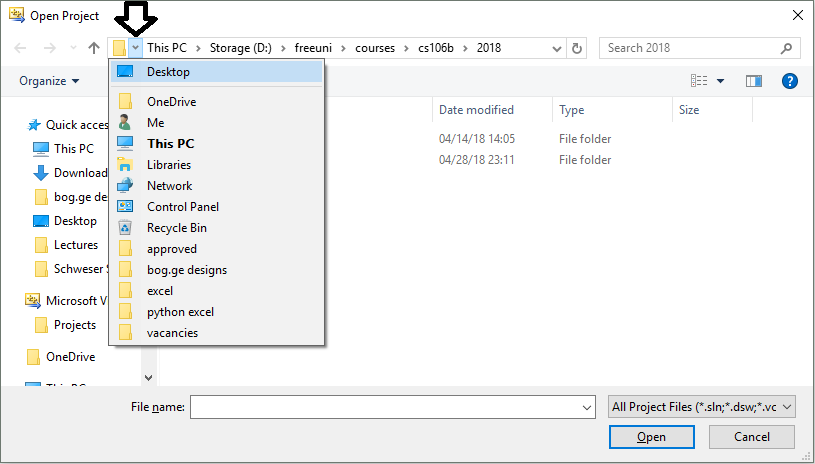
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***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. მეორე მინიმალური დამფარავი ხე (120 ქულა)**

მოცემული გაქვთ არამიმართული გრაფი, ანუ თუ არსებობს გზა A დან B ქალაქში, ეს იმას ნიშნავს რომ B დან A ქალაქშიც არსებობს გზა. თქვენი ამოცანაა გრაფში იპოვოთ რიგით მეორე მინიმალური დამფარავი ხე. ამისათვის უნად დაწეროთ findSecondMinSpanningTree ფუნცქიის რეალიზაცია.

int findSecondMinSpanningTree(Vector<Arc\*> graph, const int n)

ფუნქციას გადაეცემა გრაფი, წიბოების სიის სახით და გრაფში წვეროების რაოდენობა. თითოეული წიბო შეიცავს ინფორმაციას თუ რომელ წვეროებს აკავშირებს იგი და რა არის წიბოს წონა(იხილეთ common.h) გრაფის წვეროები გადანომრილია 0 დან n-1 მდე ჩათვლით.

ფუნქცია უნდა აბრუნებდეს რიგით მეორე მინიმალური დამფარავი ხის წიბოების წონების ჯამს, თუ ასეთი იპოვა. თუ გრაფში ვერ იპოვა ალტერნატიული დამფარავი ხე, მაშინ ფუნქციამ უნდა დააბრუნოს -1.

მეთოდი არ უნდა მუშაობდეს O(n\*n\*n\*n) ზე უარეს დროში.

**ამოცანა 2. (60 ქულა)**

სავაჭრო ცენტრში იყიდება ჩანთები და სხვადასხვა ნივთები. თითოეული ჩანთისთვის მოცემული გაქვთ ფასი და ტევადებოა კილოგრამებში, ხოლო ნივთებისთვის მოცემულია ფასი და წონა კილოგრამებში. თქვენი ამოცანაა მოცემული თანხით შეიძინოთ ზუსტად ორი ჩანთა და ნივთების ისეთი კომბინაცია რომ ნივთების ჯამური წონა იყოს მაქსიმალური, ამასთან შესაძლებელი უნდა იყოს ნივთების შეძენილი ჩანთებით გადატანა. ერთი ნივთის ან ჩანთის ორჯერ ყიდვა არ არის დაშვებული.

ფუნქციამ უნდა დააბრუნოს შეძენილი ნივთების ჯამური წონა. თუ თანხა არ გყოფნით ორი ჩანთის შესაძენად, მეთოდმა უნდა დააბრუნოს 0.

struct item {

int price;

int weight;

};

struct bag {

int price;

int cap;

}

int findBestComb(Vector<item> &items, Vector<bag> &bags,

int money) {

}

ჩანთების რაოდენობა არ აღემატება 10 ს ხოლო, ნივთების რაოდენობა 15 ს. ერთ ტესტზე ამოხსნა არ უნდა ანდომებდეს 1 წამზე მეტს.