***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***

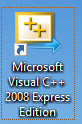
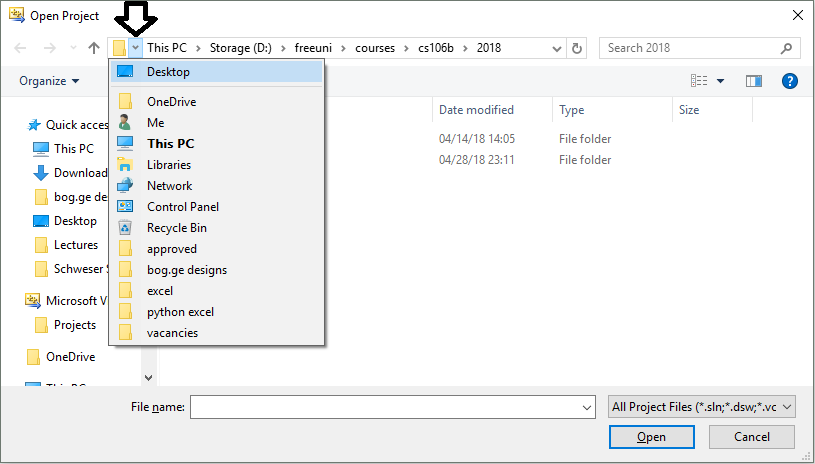
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 2.5 საათი | | |
| ***მაქსიმალური ქულა*** | 120 | | |
| ***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. მეფი შეზღუდული ზომით (120 ქულა)**

თქვენი ამოცანაა დაწეროთ CacheMap ის რეალიზაცია. სტრუქტურა თქვენთვის ნაცნობი მეფისგან განსხვავდება მხოლოდ იმით რომ მაში შენახული key – value წყვილებსი რაოდენობა შეზღუდულია.

**CacheMap(int cacheSize); - O(1)**კონსტრუქტორს გადაეცემა რაონდენობა მაქსიმუმ რამდენი წყვილის შენახვა შეუძლია სტრუქტურას.

**void put(int key, int value); - O(log n)**  
მეთოდი გადაცემულ წყვილს ინახავს სტრუქტურაში, თუ ასეთ key უკვე არსებობდა მაშინ ანახლებს მნიშვნელობას. თუ წყვილის დამატების შემდეგ, მეფში არსებული წყვილების რაოდენობა აჭარბებს კონსტრუქტორში გადაცემულ cacheSize ს, მაშინ მეფიდან უნდა მოხდეს ყველაზე ადრე დამატებული წყვილის წაშლა.

CacheMap map(2);

map.put(1, 101);

map.put(2, 102);

map.put(1, 111);

map.put(3, 103);

map.put(4, 104);

ზემოთ მოყვანილი მაგალითისთვის, პირველი სამი ხაზის შესრულების შემდეგ მეფში იქნება ორი წყვილი: 1 – 111 და 2 – 102. მესამე ელემენტის ჩამატების შემდეგ მეფიდან წაიშლება 1 – 111 წყვილი, რადგან ეს არის წყვილი რომელიც ყველაზე ადრე ჩაემატა მეფში. 4 – 104 ჩამატების შემდეგ კი, მეფიდან წაიშლება 2 – 102 წყვილი.

**int get(int key); - O(log n)**

**int size(); - O(1)**

**bool containsKey(int key); - O(log n)**

**void remove(int key); - O(log n)**

დანარჩენი მეთოდები, სტანდარტული მეფ სტრუქტურის მეთოდების ინდენტურია

ამოცანა შემოწმდება რამდენიმე ნაწილად, ქვემოთ მოცემულია მათი წონები:

1. put, get, size და containsKey მეთოდები ტესტებზე სადაც ზომის შეზღუდვა არ მონაწილეობს, ანუ casheSize უფრო დიდი იქნება ვიდრე ჯამში მეფში ჩამატებული ელემენტების რაოდენობა, წონა 15%
2. remove მეთოდი ტესტირება, წონა იქნება 45%
3. ზომითი შეზღუდვის ტესტირება, წონა 40%

გარდა ამისა თუ რომელიმე მეთოდი არღვევს დროითი სირთულის შეზღუდვას, შესაბამის კომპონენტში დაგაკლდებათ 60%. ანუ მაგალითად ზომითი შეზღუდვის ტესტირების დროს თუ put მეთოდი O(log n) ზე უფრო დიდ დროს ანდომებს, 40% ის ნაცვლად მიიღებთ 16% ს.

მზა Collection ტიპის სტრუქტურების გამოყენების უფლება არ გაქვთ, შეგიძლიათ გამოიყენოთ მასივი, ან ნოლიდან დაწეროთ თქვენთვის საჭირო სტრუქტურა და შემდეგ გამოიყენოთ. ამ პირობის დარღვევის შემთხვევაში ამოცანა განულდება.