***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***

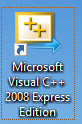
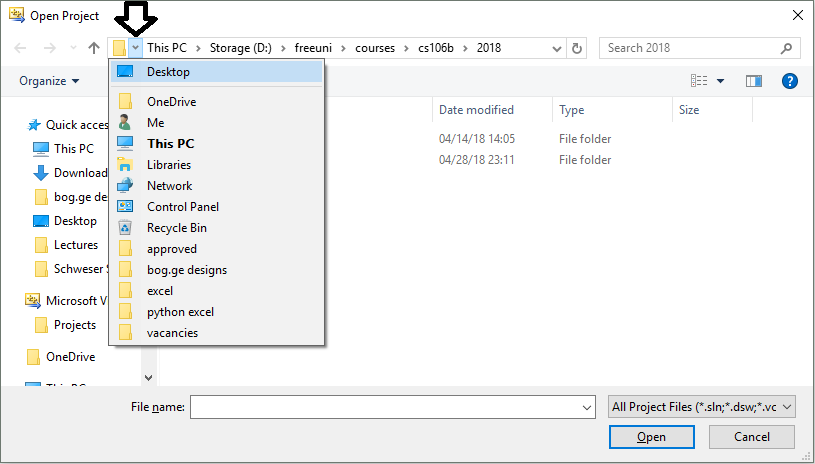
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 3 საათი | | |
| ***მაქსიმალური ქულა*** | 180 | | |
| ***სტუდენტის სახელი და გვარი:*** | | | |

სახელი: ქულა:

შუალედური 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 „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.
5. თუ გაურკვეველი შეცდომა გაქვთ, ცადეთ მენიუდან Build > Clean Solution და Build > Clean Blank Project

**ამოცანა 1. ერთი დონის ელემენტების ჯამი(60 ქულა)**

მოცემული გაქვთ ორობით ხე(უბრალოდ ორობითი და არა ორობითი ძებნის ხე) და დადებითი რიცხვი k. იპოვეთ ხის ყველა იმ წევრის ჯამი, რომლებიც იმყოფებიან ერთი და იგივე k სიმაღლეზე ხეში. ჩათვალეთ რომ ხის root ელემენტი იმყოფება 1 სიმაღლეზე, root ის შვლიები 2 ზე და ასე შემდეგ.

დაწერეთ sumLevel ფუნქციის რეალიზაცია, ფუნციას გადაეცემა ხის root ელემენტის მისამართი და დადებითი რიცხვი k, ფუნქციამ უნდა დააბრუნოს k სიმაღლეზე არსებული ყველა ელემენტის ჯამი.

int sumLevel (nodeT \* root, int k)

მაგალითი:

1  
 / \  
 2 3  
 / / \  
 4 2 4  
 /  
 4

მოცემულ ხეში 3 სიმაღლეზე არსებული ელემენტების ჯამი არის 4+2+4 = 10.

თქვენი ალგორითმი უნდა მუშაობდეს O(n) დროში. ტესტები შეგიძლიათ ნახოთ midterm2\_1\project\Blank Project\sumlevel ფოლდერში.

**ამოცანა 2. ბმული სიმრავლე (120 ქულა)**

თქვენი ამოცანაა დაწეროთ სიმრავლის რეალიზაცია ბმული სიის საშუალებით. ქვემოთ მოცემულია ფუქნციების ჩამონათვალი რაც უნდა ქონდეს სიმრავლეს.

სიმრავლეში ამატებს k-ს, თუ უკვე არის ასეთი ელემენტი მაშინ არაფერს არ ცვლის

void add(int k); - O(n)

ამოწმებს სიმრავლეში არის თუ არა გადაცემული ელემენტი

bool contains(int k); - O(n)

აბრუნებს სიმრავლეში არსებული ელემენტების რაოდენობას

int size(); - O(1)

შლის სიმრავლიდან k-ს. თუ ასეთი ელემენტი არ მოიძებნება სიმრავლეში, არაფერს არ ცვლის.

void remove(int k); - O(n)

იტერატორის დარესეტება, ანუ იძ₾ევა საშუალებას ისევ თავიდან გადავუაროთ ყველა ელემენტს.

void resetIterator(); - O(1)

აბრუნებს მიმდინარე იტერაციის დროს დარჩა თუ არა ელემენტი

bool hasNext(); - O(1)

აბრუნებს სიმრავლიდან მომდევნო ელემენტს, მიმდევრობას არ აქვს მნიშვნელობა

int next(); - O(1)

იტერატორის გამოყენებისთვის პირველ რიგში უნდა გამოიძახოთ resetIterator, მხოლოდ ამის შემდეგ შეიძლება hasNext და next ფუნქციების საშუალებით გადავუაროთ რიგში არსებულ ელემენტებს. მაგალითი ქვემოთ მოცემული კოდი დაბეჭდავს სიმრავლეში არსებულ ყველა ელემენტებს.

s.resetIterator();

while(s.hasNext()) {

cout << s.next() << endl;

}

ჩათვალეთ, რომ სანამ იტერატორით ელემენტებზე გადავლა ხდება, არ მოხდება add-ის ან remove-ის გამოძახება.

ამოცანის ამოხსნისას არ გაქვთ უფლება გამოიყენოთ მასივი, ან რომელიმე მზა ქოლექშენ ტიპის სტრუქტურა. ტესტების გენერაცია ხდება main.c ფაილში.