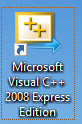
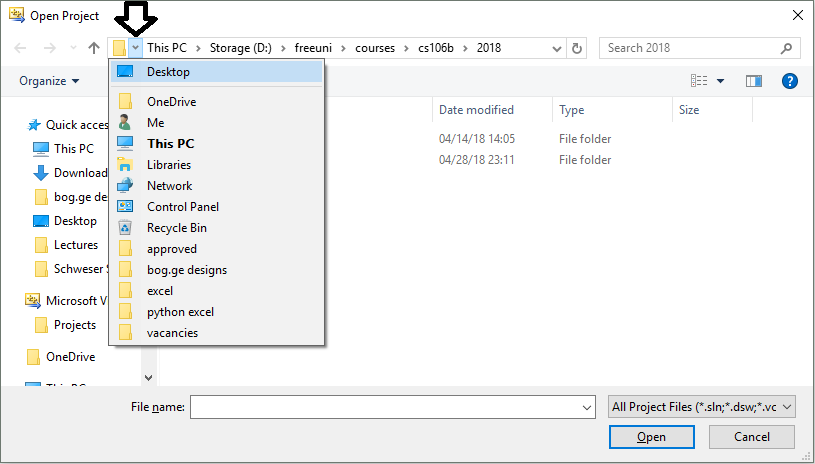
|  |  |
| --- | --- |
| საგანი | პროგრემირების აბსტრქციები |
| პროგრამა | MACS |
| კურსი | I, II |
| ჯგუფი/ჯგუფები |  |
| ლექტორი | გიორგი ბოჭორიშვილი |
| **გამოცდის ფორმატი (მონიშნეთ)** | |
| ☐ დახურული წიგნი | |
| ღია წიგნის შემთხვევაში მონიშნეთ | ☒ ელექტრონული მასალა  ☒ წიგნები |
| გამოცდის კომპიუტერში წერის შემთხვევაში,მონიშნეთ მხოლოდ ის კონკრეტული პროგრამული უზრუნველყოფა (ერთი ან რამდენიმე),რომელიც უნდა გამოიყენოს სტუდენტმა.  ***ნაშრომების შეგროვება და მოწოდება მოხდება , თქვენ მიერ მითითებული პროგრამის გაფართოების მიხედვით.*** | ☒ **Word** (\*.docx; \*.doc)  ☐ **Excel** (\*.xlsx; \*.xls)  ☐ **PowerPoint** (\*.pptx; \*.ppt)  ☐ **Oracle Java** (\*.java)  ☐ **IBM SPSS Statistics 20** (\*.sav)  ☐ **SPSS Statistics 17.0** (\*.spv; \*.sav)  ☐ **ISIS 7 Professional** (\*.DSN)  ☐ **Multisim 12.0** (\*.ms12)  ☐ **ARES 7 Professional** (\*.LYT)  ☐ **Flowcode V4 for AVRs** (\*.fcf\_avr)  ☐ **logisim-generic-2.7.1** (\*.circ)  ☐ **GeoGebra** (\*.ggb)  ☐ **Matlab 2016a** (\*.m; \*.mat)  ☐ **EpiInfo** (\*.prj)  ☐ **LCD Vision**  ☒ **Microsoft Visual C++ 2008 Express Edition** (\*.vcproj)  ☐ **Codevision AVR** (\*.c)  ☐ **PHStat2** (\*.xlsx)  ☐ **Kawa** (\*.\*)  ☒ **Dev-C++** (\*.cpp;\*.cc;\*.cxx;\*.c++;\*.cp; \*.c)  ☒ **Visual Studio Code** (\*.cpp; \*.c; \*.java; \*.sql; \*.jsx; \*.js)  ☒ **Visual Studio 2015**  ☐ **Gcc** (\*.c; \*.cpp; \*.java)  ☐ **Eclipse IDE 9** (\*.txt; \*.cpp; \*.c; \*.java; \*.sql; \*.jsx; \*.js)  ☒ **Notepad ++** (\*.txt; \*.cpp; \*.c; \*.java; \*.sql; \*.jsx; \*.js)  ☐ **სხვა \_** (გაფართოება) |
| თუ სტუდენტი ერთდროულად მუშაობს რამოდენიმე პროგრამაში და საბოლოო სახით, საჭიროა ნაშრომების დაარქივება, გთხოვთ მონიშნოთ შესაბამისი გრაფა. | ☒\*.zip;  ☒\*.rar; |
| ფურცელზე წერის შემთხვევაში მონიშნეთ შესაბამისი | ☐ სრულად ფურცელზე  ☐ ფურცელზე და კომპიუტერში |

**\*მონიშნეთ გამოცდის ფორმატი (მიუთითეთ √)**

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 მარცხნივ ჩამონათვალში აირჩიეთ Desktop. თუ ჩამონათვალში არ არის Desktop დააჭირეთ სურათზე აღნიშნულ ისარს და აირჩიეთ.   
   **
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. **არქივი დატოვეთ დესკტოპზე თქვენს ფოლდერში**

**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 დამიძახეთ მე, ან საგამოცდოს რომელიმე თანამშრომელს. თქვენს კომპიუტერზე საჭირო იქნება Blank Project პროცესის მოკვლა რაშიც IT ის თანამშრომელი დაგეხმარებათ.
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. (40 ქულა) მარცხენა ფოთლების ჯამი**

დაწერეთ sumOfLeftLeaves ფუნქციის რეალიზაცია, რომელსაც გადაეცემა ორობითი ხე და აბრუნებს მარცხენა ფოთლების მნიშვნელობების ჯამს. შეგახსენებთ ფოთოლი არის ხის ისეთი ელემენტი, რომელსაც არცერთი შვილი არ ყავს. ხოლო მარცხენა ფოთოლი არის ისეთი ფოთოლი, რომელიც მშობელი ელემენტის მარცხენა შვილია. თუ მოცემული გაქვთ მხოლოდ ერთი, root node ისგან შედგენილი ხე, იგი არ ითვლება მარცხენა შვილად და ასეთ შემთხვევაში პასუხი არის 0.

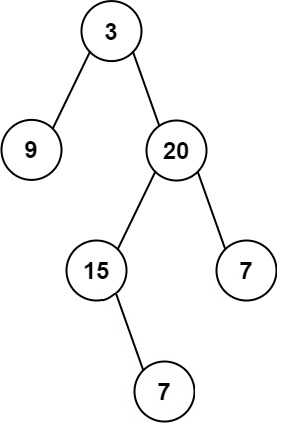
int sumOfLeftLeaves(TreeNode \* root)

მაგალითი 1:



პასუხი: 24 (9 + 15)

მაგალითი 2:



პასუხი: 9 (15 უკვე აღარ არის ფოთოლი და ამიტომ აღარ მონაწილეობს პასუხში)

final2\project\Blank Project\leftleaves ფოლდერში შეგიძლიათ იხილოთ სხვა მაგალითები.

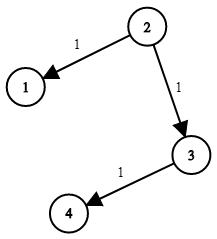
თქვენი ამოხსნა უნდა მუშაობდეს O(n) დროში

**ამოცანა 1. (80 ქულა) ქსელში სიგნალის დაგვიანება**

მოცემული გაქვთ n ცალი წერტილისგან(node) შედგენილი ქსელი, რომლებიც გადანომრილია 1 დან n მდე. წერტილები ერთმანეთთან დაკავშირებულია მიმართული წიბოებით(edge). თითოეული წიბოსთვის მოცემული გაქვთ, თუ რა დრო ჭირდება საწყისი წერტილიდან საბოლოო წერტილამდე სიგნალის გადაცემას.

ჩვენ გავუშვებთ სიგნალს k წერტილიდან, თქვენ უნდა იპოვოთ დრო რაც საჭიროა იმისათვის რომ სიგნალი მივიდეს ყველა წერტილამდე. ამისათვის უნდა დაწეროთ networkDelayTime ფუნქციის რეალიზაცია, რომელიც აბრუნებს მინიმალურ დროს რაც საჭიროა იმისთვის რომ სიგნალმა მიაღწიოს ყველა წერტილამდე, თუ ეს უკანასკნელი შეუძლებელია ფუნქციამ უნდა დააბრუნოს -1.

მაგალითი:



n = 4, k = 2

პასუხი: 2

int networkDelayTime(HashMap<int, Vector<Edge> > &graph, int n, int k)

graph მეფის საშუალებით მოცემული გაქვთ თუ როგორ არიან დაკავშირებული წერტილები ერთმანეთთან. Edge სტრუქტურაში from ველი არის საწყისი წერტილი, to ველი საბოლოო და time ველი დრო რაც ჭირდება სიგნალს საწყისი წერტილიდან საბოლოომდე მისაღწევად.

თქვენი ალგორითმი უნდა მუშაობდეს O(n\*n) დროში.