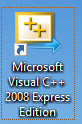
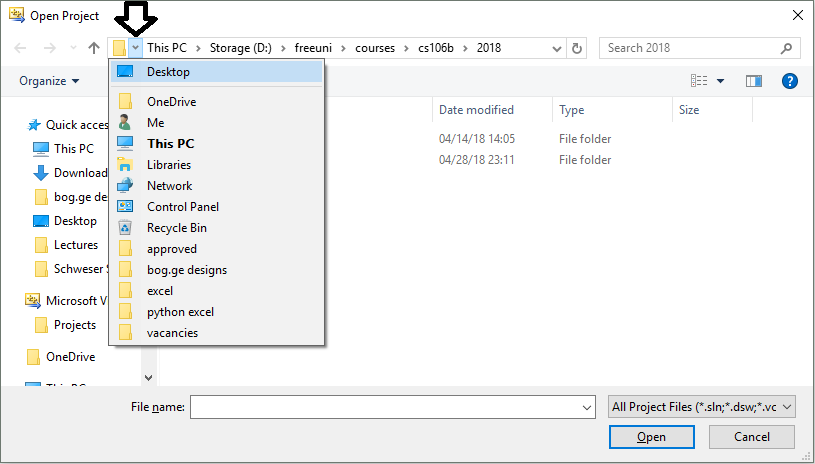
|  |  |
| --- | --- |
| საგანი | პროგრემირების აბსტრქციები |
| პროგრამა | MACS |
| კურსი | I, II |
| ჯგუფი/ჯგუფები |  |
| ლექტორი | გიორგი ბოჭორიშვილი |
| **გამოცდის ფორმატი (მონიშნეთ)** | |
| ☐ დახურული წიგნი | |
| ღია წიგნის შემთხვევაში მონიშნეთ | ☒ ელექტრონული მასალა  ☒ წიგნები |
| გამოცდის კომპიუტერში 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. თქვნს კომპოიყერზე 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. (60 ქულა) Split**

მოცემული გაქვთ მხოლოდ ციფრებისგან შედგენილი ტექსტი. დაადგინეთ შესაძლებელია თუ არა მოცემული ტექსტის დაყოფა ორ ან მეტ არა-ცარიელ ქვეტექსტად ისე რომ ქვეტექსტების რიცხვითი მნიშვნელობები დალაგებული იყოს კლებადობით და ყოველ ორ მეზობელ ქვეტექსტს შორის სხვაობა იყოს ერთის ტოლი.

მაგალითად ტექსტი "0090089" შეიძლება დაიყოს ["0090", "089"], რომელთა შესაბამიი რიცხვითი მნიშვნელობები არის [90,89].

კიდევ ერთი მაგალითი "001" შეიძლება დაიყოს ["0", "01"], ["00", "1"], ან ["0", "0", "1"] ქვეტექსტებად. მაგრამ როგორც ვხედავთ არცერთი მათგანი არ გვაძლევს კლებადობით დალაგებული რიცხვებს.

დაწერეთ splitString(string s) ფუნქციის იმპლემენტაცია, რომელიც აბრუნებს true თუ გადაცემული ტექსტის დაყოფა შესაძლებელია, თუ არადა აბრუნებს false-ს.

მაგალითი 1:

Input: s = "1234"

Output: false

მაგალითი 2:

Input: s = "050043"

Output: true

შეიძლება დაიყოს ["05", "004", "3"], რასაც შეესაბამება რიცხვები [5,4,3]

მაგალითი 3:

Input: s = "9080701"

Output: false

ტექსტი შეიძლება შედგებოდეს მაქსიმუმ 15 ციფრისგან.

ტექსტის რიცხვში გადაქცევისთვის გამოიყენეთ stringToLong ფუნქცია, რომელიც აბრუნებს long ტიპის მნიშვნელობას. გამოყენების მაგალითი იხილეთ split.cpp ფაილში.

main.cpp ფაილში შეგიძლიათ იხილოთ ტესტების მაგალითები.

**ამოცანა 2. (60 ქულა) Closest**

მოცემული გაქვთ დადებითი რიცხვი target და დადებითი რიცხვების სია list. თქვენი ამოცანაა იპოვოთ target მნიშვნელობასთან ყველაზე ახლო რიცხვი, რომელიც მიიღება list სიაში არსებული რიცხვების შეკრება-გამოკლებით. თუ ასეთი რიცხვი ორია, მაშინ დააბრუნეთ მათ შორის მინიმალური.

თითოეული რიცხვი list სიიდან შეგიძლიათ გამოიყენოთ მხოლოდ ერთხელ, ან საერთოდ არ შეიყვანოთ გამოსახულებაში. დაწერეთ findClosest(Vector<int> &list, int target) ფუნქციის იმპლემენტაცია.

მაგალითი 1:

Input: target = 10, list = [1,2,2,17,19]

Output: 12 (17-1-2-2 = 12)

მაგალითი 2:

Input: target = 14, list = [10,2,21]

Output: 13 (21-10+2 = 13)

მაგალითი 3:

Input: target = 1, list = [5,29,2,21]

Output: 1 (29-21-5-2 = 1)

მაგალითი 3:

Input: target = 9, list = [5,29,2,21]

Output: 8 (29-21 = 9, ასევე შესაძლებელია მივიღოთ 10, 29-21+2=10, რომელიც ასევე 1 ით არის დაშორებული target დან, მაგრამ ამ შემთხვევაში პასუხი არის 8 ს და 10 ს შორის მინიმალური)

list შეიძლება შედგებოდეს მაქსიმუმ 15 რიცხვისგან. ორი რიცხვის სხვაობის მოდულის დასათლელად შეგიძლიათ გამოიყენოთ abs ფუნქცია: abs(a-b).

\project\Blank Project\closest ფოლდერში შეგიძლიათ იხილოთ ტესტების მაგალითები. ტესტში პირველ ხაზზე წერია რიცხვების რაოდენობა, target ის მნიშვნელობა და პასუხი. ხოლო მეორე ხაზზე წერია მოცემული რიცხვების სია.