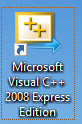
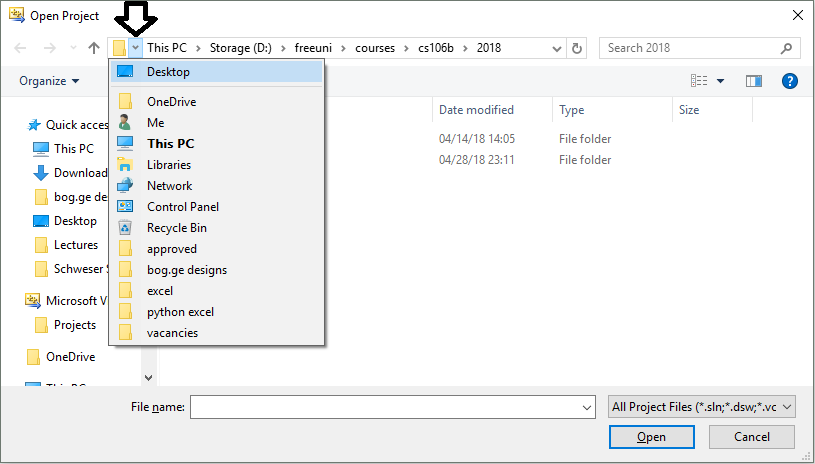
|  |  |
| --- | --- |
| საგანი | პროგრემირების აბსტრქციები |
| პროგრამა | MACS |
| კურსი | I, II |
| ჯგუფი/ჯგუფები |  |
| ლექტორი | გიორგი ბოჭორიშვილი |
| **გამოცდის ფორმატი (მონიშნეთ)** | |
| ☐ დახურული წიგნი | |
| ღია წიგნის შემთხვევაში მონიშნეთ | ☒ ელექტრონული მასალა  ☒ წიგნები |
| გამოცდის კომპიუტერში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

შუალედური 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. **ვებ ბრაუზერში გახსენით მისამართი** [http://192.168.210.5](http://192.168.210.5/) **და ატვირთეთ არქივი. არქივი დატოვეთ დესკტოპზე.**

**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) კოდი**

ხანგრძლივი მოგზაურობიდან დაბრუნებულ მორტის სადარბაზოს კარზე კოდიანი საკეტი დახვდა დამონტაჟებული. საკეტის დეტალურად შესწავლისას მორტიმ ორი რამ აღმოაჩინა:

1. კოდი შედგება n ცალი სიმბოლოსგან
2. ღიალკების ნაწილი გაცვეთილია, საიდანაც დგინდება რომელი სიმბოლოები მონაწილეობენ კოდში

დაეხმარეთ მორტის მიღებული ინფორმაციის საფუძველზე კოდის ყველა შესაძლო ვარიანტის პოვნაში. ამისათვის თქვენ უნდა დაწეროთ **generateCodes** ფუნქციის რეალიზაცია, რომელსაც გადაეცემა სიმბოლოების რაოდენობა და კოდში მონაწილე სიმბოლოების სია. ფუნქცია უნდა აბრუნებდეს კოდის ყველა შესაძლო ვარიანტს, ზრდადობით დალაგებულს. **symbols** სტრინგში სიმბოლოები არ მეორდება და დალაგებულია ზრდადობით.

**Vector<string> generateCodes(int n, string symbols)**

მაგალითი 1:

Input: n = 2, symbols = "ac"

Output: ["aa", "ac", "ca", "cc"]

მაგალითი 2:

Input: n = 3, symbols = "bd"

Output: ["bbb", "bbd", "bdb", "bdd", "dbb", "dbd", "ddb", "ddd"]

მაგალითი 3:

Input: n = 7, symbols = "1"

Output: ["1111111"]

ტესტები შეგიძლიათ იხილოთ “project\Blank Project\code” ფოლდერში არსებულ ფაილებში. ფაილში პირველი ხაზში მოცემულია n და სიმბოლოების სია. ხოლო მეორე ხაზში მოცემულია სწორი პასუხი, ანუ ყველა შესაძლო კოდი. თქვენი ამოხსნა მაქსუმუმ 10 წამში უნდა აბრუნებდეს სწორ პასუხს, წინააღმდეგ შემთხვევაში ტესტი განულდება. თუ პროექტში არსეუბულ ტესტებზე ამოხსნამ დროულად დაარბრუნა პასუხი, ესეგი თქვენი ამოხსნის ეფექტურობა მისაღებია.

**ამოცანა 2. (60 ქულა) რიცხვები**

თქვენი ამოცანაა იპოვოთ n ცალი ციფრისგან შედგენილი ყველა დადებითი რიცხვი, რომელშიც ნებისმიერი ორი მეზობელი ციფრის სხვაობის მოდული არის k-ს ტოლი. გაითვალისწინეთ რომ პასუხში არცერთი რიცხვი არ უნდა იწყებოდეს 0 ზე.

დაწერეთ **numsSameConsecDiff** ფუნქციის იმპლემენტაცია, რომელსაც გადაეცემა ციფრების რაოდენობა და მეზობელ ციფრებს შორის სხვაობის მოდული. დაბრუნებული რიცხვები უნდა იყოს ზრდადობით დალაგებული.

**Vector<int> numsSameConsecDiff(int n, int k)**

მაგალითი 1:

Input: n = 3, k = 7

Output: [181,292,707,818,929]

მაგალითი 2:

Input: n = 2, k = 1

Output: [10,12,21,23,32,34,43,45,54,56,65,67,76,78,87,89,98]

მაგალითი 3:

Input: n = 2, k = 0

Output: [11,22,33,44,55,66,77,88,99]

2 <= n <= 9

0 <= k <= 9

ტესტები შეგიძლიათ იხილოთ “project\Blank Project\numbers” ფოლდერში არსებულ ფაილებში. ფაილში პირველი ხაზში მოცემულია n და k. ხოლო მეორე ხაზში მოცემულია სწორი პასუხი. თქვენი ამოხსნა მაქსუმუმ 10 წამში უნდა აბრუნებდეს სწორ პასუხს, წინააღმდეგ შემთხვევაში ტესტი განულდება. თუ პროექტში არსეუბულ ტესტებზე ამოხსნამ დროულად დაარბრუნა პასუხი, ესეგი თქვენი ამოხსნის ეფექტურობა მისაღებია.