| 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 პროექტის არქივი თქვენს დესკტოპზე.**
2. **ამოაარქივეთ პროექტი. მაუსის მენიუ ღილაკით დააჭირეთ არქივზე და აირჩიეთ Extract here.**
3. **გახსენით ფოლდერი და გაუშვით Exam.sln ფაილი **
4. **თუ შემოგთავაზათ Visual Studio ს ვერსიებს შორის არჩევანი, აირჩიეთ 2022**
5. **თუ შემოგთავაზათ Sign In ოფციები, აირჩიეთ Skip this for now. შემდეგ ფორმაზე აირჩიეთ სასურველი სტილი და დააჭირეთ Start Visual Studiuo ს**
6. **შესაძლოა რამდენიმე წუთი დაჭირდეს გაშვებას. თუ 5 წუთის შემდეგ არ გაეშვა, ხელახლა გახსენით Exam.sln ფაილი.**
7. **თუ გახსნისას გამოიტანა შეტყობინება: “One or more projects in the solution where not loaded”, დახურეთ სტუდიო და ხელახლა გახსენით Exam.sln ფაილი**
8. **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.**
9. **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.**
10. **ამოხსნის დასრულების შემდეგ დაარქივეთ „src” ფოლდერი და არქივს დაარქვით სახელად თქვენი მეილის პრეფიქსი, მაგალითად gboch10.zip**
11. **არქივი დატოვეთ დესკტოპზე თქვენს ფოლდერში**

**საყურადღებო:**

1. კოდი გაუშვით f5 ღილაკზე დაჭერით ან მწვანე Play ღილაკზე დაჭერით:



წინააღმდეგ შემთხვევაში კონსოლის ფორმა არ გამოჩნდება.

1. Fatal error LNK1168 cannot open - ამ შეცდომის შემთხვევაში მასალების ფოლდერიდან ჩამოტვირთეთ დესკტოპზე ExamKiller.exe ფაილი და გაუშვით. თუ არ მოგვარდა პრობლემა დამიძახეთ მე, ან საგამოცდოს რომელიმე თანამშრომელს. თქვნს კომპოიყერზე საჭირო იქნება Blank Project პროცესის მოკვლა რაშიც IT ის თანამშრომელი დაგეხმარებათ.
2. თუ კოდში ცვლილება შეგაქვთ მაგრამ გაშვებისას არანაირად არი ისახება, დარწმუნდით რომ წინა წარმატებული ბილდი არ ეშვება. F7 ით ცადეთ დაკომპილირება.
3. თუ გსურთ ხაზების ნუმერაციის გამოჩენა, გახსენით Tools > Options მენიუ, გადადით Text Editor > All Languages > General და მონიშნეთ Line numbers.
4. თუ გაურკვეველი შეცდომა გაქვთ, ცადეთ მენიუდან Build > Clean Solution და Build > Clean Blank Project

**ამოცანა 1. (50 ქულა) ჯავშნები**

თქვენ გაქვთ ბინა რომელსაც აქირავებთ სხვადასხვა პლატფორმების საშუალებით. წლის დასაწყისში მიიღეთ მთელი წლის ჯავშნების სია. თითოეილი ჯავშანი შედგება შემდეგი ინფორმაციისგან:

* ჯავშნის დაწყების თარიღი, ჩათვლით
* ჯავშნის დასრულების თარიღი, ჩათვლით
* ჯამური ღირებულება

თარიღი მოცემულია 1 დან 365 მდე რიცხვის საშუალებით. ანუ პირველ იანვარს შეესაბამება ერთიანი, პირველ თებერგვალს შეესაბამება 32 და ასე შმედეგ.

თქვენი ამოცანაა აარჩიოთ ჯავშნების საკუთესო კომბინაცია, ისე რომ ჯამური შემოსავალი იყოს მაქსიმალური. რათქმაუნდა არ შეიძლება ერთი და იგივე თარიღზე რამდენიმე ჯავშანი.

დაწერეთ bestBookings ფუნქციის რეალიზაცია.

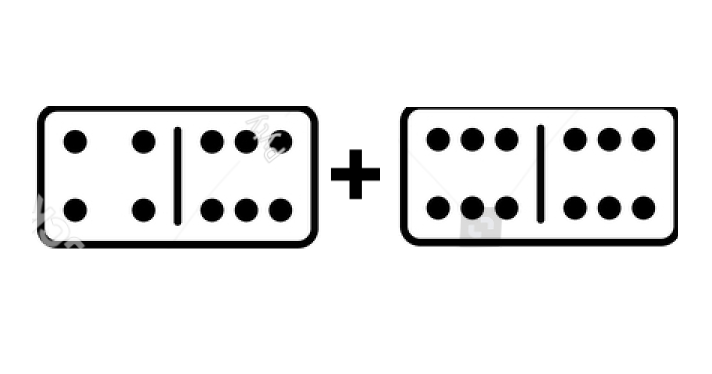
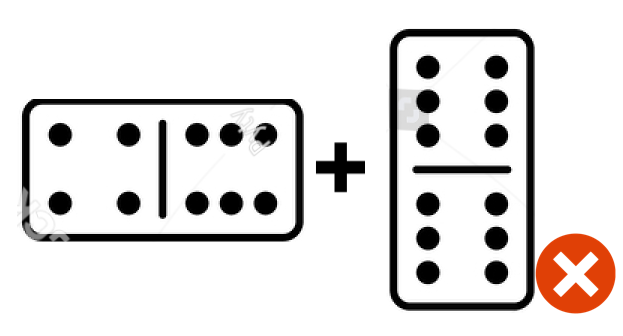
int bestBookings(Vector<bookingT> &bookings)

ფუნქციას გადაეცემა ჯავშნების სია და აბრუნებს მაქსიმალურ შემოსავალს. ჯავშნების რაოდენობა არ აღემატება 30-ს.

main.cpp ფაილში შეგიძლიათ იხილოთ ტესტების მაგალითები. თუ თქვენი ამოხსნა main.cpp ფაილში მოცემული თითოეული ტესტისთვის, 4 წამზე ნაკლებ დროში აბრუნებს პასუხს, ჩათვალეთ რომ საკმარისად სწრაფი ახმოსნა გაქვთ. ტესტი რომელზეც პასუხის დაბრუნებას 4 წამზე მეტი დრო დაჭირდება არ ჩაითვლება.

**ამოცანა 1. (70 ქულა) დომინო**

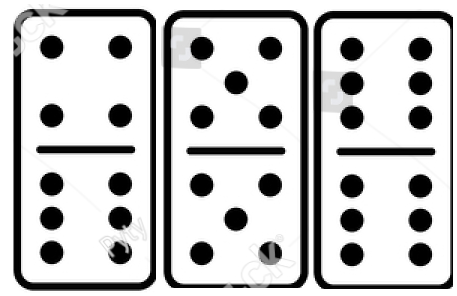
მორტი სახლში მარტო დარჩა და ერთკაცა დომინოს თამაშს სწავლობს. თამაშის წესები ასეთია:

* დომინო შედგება 28 ქვისგან. ქვა გაყოფილია ორ ნაწილად და თოთოეულ ნაწილზე გამოსახულია 0-6 მდე ცალი შავი წერტილი.
* მოთამაშე იღებს n ცალ შემთხვევით ქვას და ყოველ სვლაზე თითო ქვას დებს. ქვის დადება შესაძლებელია მხოლოდ ჰორიზონტალურად. მარცხენა სურათზე ნაჩვენები სვლა დასაშვებია, ხოლო მარჯვენაზე არა:  
   
* პირველ სვლაზე შეუძლია დადოს ნებისმიერი ქვა.
* ყოველ მომდევნო სვლაზე შეუძლია ქვა მიადოს ორი ბოლოდან რომელიმეს. მიდება ხდება მხოლოდ ერთი და იმავე მნიშვნელობის ციფრის მქონე ქვების.
* ყოველთვის როცა დადებული ქვების ორ ბოლოში მყოფი ციფრების ჯამი არის 5 ის ჯერადი, მოთამაშე ემატება ამ ორი ციფრის ჯამის რაოდენობა ქულა.
* თამაში მთავრდება როცა მოთამაშე ყველა ქვას დადებს, ან როცა სვლა აღარ აქვს
* მოთამაშის მიზანია რაც შეიძლება მეტი ქულა მოაგროვოს

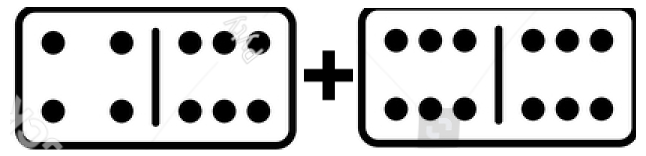
თქვენი ამოცანაა დაეხმაროთ მორტის თამაშში. ამისათვის, მოცემული n ცალი ქვისთვის უნდა იპოვოთ საუკეთესო თამაშის სტრატეგია, რომლითაც შესაძლებელი იქნება მაქსიმალური ქულის დაგროვება. დაწერეთ maxDominoScore ფუნქციის იმპლემენტაცია. რომელსაც გადაეცემა, n ცალი დომინოს ქვა და აბრუნებს მაქსუმალური ქულის რაოდენობას.

**მაგალითი 1:**

მოცემული გვაქვს სამი ქვა



საუკეთესო სტრატეგია:



პირველ სვლაზე დავიწერთ 10 ს, მეორე სვლაზეც 10 ს, ჯამში 20 ქულა. მესამე ქვა ხელში დარჩება.

**მაგალითი 2:**

მოცემული გვაქვს ოთხი ქვა:   
4|3; 3|5; 5|2; 2|6;

საუკეთესო სტრატეგია:   
პირველი ორი სვლა: 3|5 + 5|2 - შედეგად დავიწერთ 5 ქულას.   
მომდეოვნო ორი სვლა: 4|3 + 3|5 + 5|2 + 2|6 დავიწერთ კიდევ 10 ქულას, სულ ჯამში 15 ქულა.

main.cpp ფაილში შეგიძლიათ იხილოთ ტესტების მაგალითები. თუ თქვენი ამოხსნა main.cpp ფაილში მოცემული თითოეული ტესტისთვის, 4 წამზე ნაკლებ დროში აბრუნებს პასუხს, ჩათვალეთ რომ საკმარისად სწრაფი ახმოსნა გაქვთ. ტესტი რომელზეც პასუხის დაბრუნებას 4 წამზე მეტი დრო დაჭირდება არ ჩაითვლება.