***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***

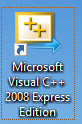
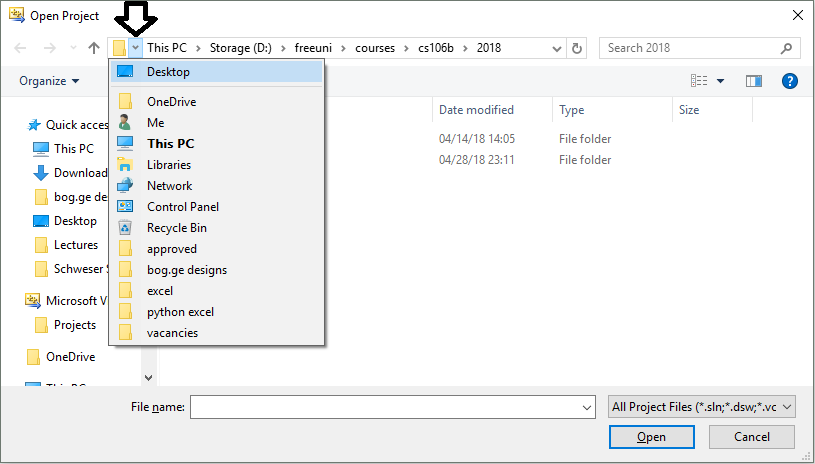
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | | |
| ***გამოცდის ფორმა*** | ღია წიგნი | | |
| ***გამოცდის***  ***ხანგრძლივობა*** | 2 საათი | | |
| ***მაქსიმალური ქულა*** | 120 | | |
| ***სტუდენტის სახელი და გვარი:*** | | | |

სახელი: ქულა:

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 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. 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 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. (120 ქულა) ბრაუზერი**

თქვენი ამოცანაა დაწეროთ სტრუქტურა რომელსაც აქვს ბრაუზერის ანალოგიური ფუნქციონალი. ქვემოთ შეგიძლიათ იხილოთ კლასის ინტერფეისი.

Browser()

კონსტრუქტორი, ქმნის ცარიელ სტრუქტირას. უნდა მუშაობდეს O(1)-ში.

void NewTab()  
ქმნის ახალ ტაბს, რომელშიც ჩატვირთულია ცარიელი გვერდი (Blank Page). ახლად შექმნილი ტაბი ემატება სიის ბოლოში, ტაბები გადანომრილია 0 დან size-1 მდე, სადაც size არის ტაბების რაოდენობა. უნდა მუშაობდეს ამორტიზებულ O(1)-ში.

void CloseTab(int tabIndex)

ხურავს tabIndex ნომრის მქონე ტაბს, ანუ შლის მას ტაბების სიიდან. უნდა მუშაობდეს O(n)-ში, სადაც n ტაბების რაოდენობაა.

int Size()

აბრუნებს ბრაუზერში ტაბების რაოდენობას, უნდა მუშაობდეს O(1)-ში.

GoToPage(int tabIndex, string url)

tabIndex ნომრის მქონე ტაბში გადადის ახალ გვერდზე რომლის მისამართია url. უნდა მუშაობდეს ამორტიზებულ O(1)-ში.

string GetUrl(int tabIndex)

აბრუნებს tabIndex ნომრის მქონე ტაბში ჩატვირთულ გვერდის მისამართს. თუ ტაბი სუფთაა მაშინ აბრუნებს ცარიელ სტრინგს. უნდა მუშაობდეს O(1)-ში.

void GoBack(int tabIndex)

tabIndex ნომრის მქონე ტაბზე ასრულებს back მოქმედებას, ანუ ბრუნდება წინა გვერდზე. თუ Tab ზე ჩატვირთულია ცარიელი გვერდი, GoBack მეთოდი არაფერს არ ცვლის. უნდა მუშაობდეს ამორტიზებულ O(1)-ში.

void GoForward(int tabIndex)

tabIndex ნომრის მქონე ტაბზე ასრულებს forward ოპერაციას, ანუ თუ მიმდინარე გვერდზე მოვხვდით GoBack მოქმედების საშუალებით, მაშინ GoForward მა უნდა დაგვაბრუნოს გვერდზე რომელიც ჩატვურთული იყო ტაბზე GoBack ის გამოძახებამდე. თუ გვერდზე მოვხვდით GoToPage ის საშუალებით, მაშინ GoForward მეთოდი არაფერს არ ცვლის. უნდა მუშაობდეს ამორტიზებულ O(1)-ში.

გაითვალისწინეთ, იმისათვის რომ სტრუქტურამ გამართულად იმუშაოს, არ გევალებათ რაიმე ვებგვერდის გახსნა ან ჩატვრთვა, თქვენ უბრალოდ უნდა შეინახოთ თუ რა გვეერდებიზე და რა რიგითობით მოხდა ნავიგაცია.

**მაგალითი 1:**

Browser b;

cout << b.size() << endl; //დაბეჭდავს 0 ს

b.NewTab();

b.NewTab();

b.NewTab();

cout << b.size() << endl; //დაბეჭდავს 3 ს

b.GoToPage(0, “urlA”);

b.GoToPage(1, “urlB”);

b.GoToPage(2, “urlC”);

cout << b.GetUrl(1) << endl; // დაბეჭდავს urlB

b.CloseTab(1);

cout << b.GetUrl(1) << endl; // დაბეჭდავს urlC

b.GoBack(0);// დაბრუნდება ცარიელ გვერდზე

cout << b.GetUrl(0) << endl; // დაბეჭდავს ცარიელ სტრინგს

b.GoToPage(1, “urlD”);

b.GoBack(1);

b.GoBack(1);

b.GoForward(1);

cout << b.GetUrl(1) << endl; // დაბეჭდავს urlC

b.GoForward(1);

cout << b.GetUrl(1) << endl; // დაბეჭდავს urlD

**მაგალითი 2:**

Browser b;

b.NewTab();

b.GoToPage(0, “urlA”);

b.GoToPage(0, “urlB”);

b.GoBack(0);

cout << b.GetUrl(0) << endl; // დაბეჭდავს urlA

b.GoToPage(0, “urlD”);

cout << b.GetUrl(0) << endl; // დაბეჭდავს urlD

b.GoBack(0);

cout << b.GetUrl(0) << endl; // დაბეჭდავს urlA

b.GoForward(0);

cout << b.GetUrl(0) << endl; // დაბეჭდავს urlD

დამატებითი მაგალითებისთვის იხილეთ main.cpp ფაილში ტესტები. ჩათვალეთ რომ მეთოდებს არ გადაეცემათ არავალიდური ინდექსი.

**დამატებითი შეზღუდბა:**

* **არ გაქვთ უფლება გამოიყენოთ რომლიმე მზა ქოლექშენ სტრუქტურა, როგორიხაა Vector, Stack, Queue, Map, Set …**
* **თუ მაინც გამოიყენებთ რაიმე მზა ქოლექშენ სტრუქტურას, მიიღებთ დაგროვებული ქულის 40% ს**