***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***

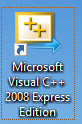
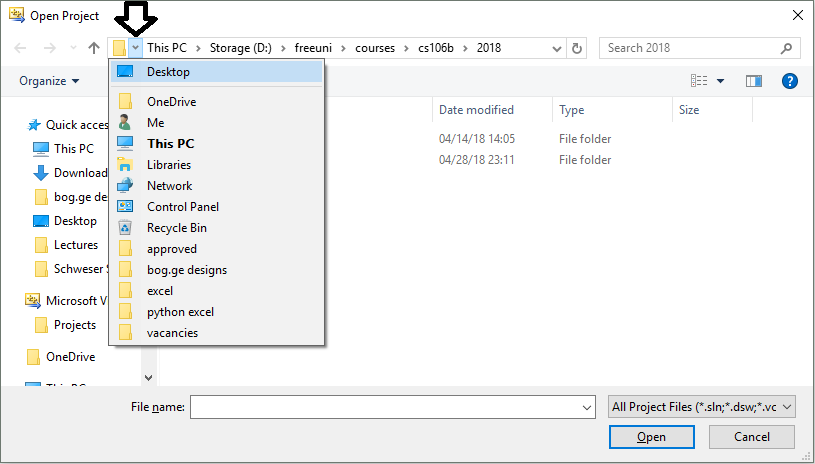
|  |  |  |  |
| --- | --- | --- | --- |
| ***სკოლა*/*საგანმანათლებლო პროგრამა*** | მათემატიკა და კომპიუტერული მეცნიერება | ***სტუდენტის მიერ მიღებული ქულა*** |  |
| ***საგანი*** | პროგრამირების აბსტრაქციები | | |
| ***ლექტორი*** | გ. ბოჭორიშვილი | | |
| ***კურსი*** | I, II | | |
| ***ჯგუფი*** |  | |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

სახელი: ქულა:

შუალედური გამოცდა

პროგრამირების აბსტრაქციებში

**შეასრულეთ შემდეგი ინსტრუქციები, წინააღმდეგ შემთხვევაში შესაძლოა თქვენი ნაშრომი არ შეფასდეს.**

1. **ჩამოტვირთეთ საგამოცდო საკითხები და თქვენი ვარიანტის შესაბამისი პროექტის არქივი midterm1.zip 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. main.cpp ფაილში წერია ამოცანების ტესტებზე შემოწმება და მისი გაშვებით კონსოლში გამოვა ინფორმაცია თუ რომელ ტესტებზე გაიარა\ჩაიჭრა თქვენმა ამოხსნამ. ფუნქციის პროტოტიპის შეცლა არ შეიძლება. შეგიძლიათ შემოიტანოთ დამხმარე ფუნქციები. არ გაქვთ გლობალური ცვლადების გამოყენების უფება.**
8. **ამოხსნის დასრულების შემდეგ დაარქივეთ „problems” ფოლდერი და არქივს დაარქვით სახელად თქვენი მეილის პრეფიქსი, მაგალითად gboch10.zip**
9. **ვებ ბრაუზერში გახსენით მისამართი** [http://192.168.210.5](http://192.168.210.5/) **და ატვირთეთ არქივი.**

**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 2. ChunkVector (110 ქულა)**

როგორც იცით ბმული სიის ერთი წევრი ინახავს მნიშვნელობას და მომდევნო წევრის მისამართს. ჩანკ ლისტი იგივე ბმული სიაა, უბრალოდ ერთი განსხვავებით, ყუთში ერთი მნიშვნელობის ნაცვლად ინახება მნიშვნელობების მასივი და შემდეგი ყუთის მისამართი, როგორც ქვემოთ დიაგრამაზეა ნაჩვენები.



თქვენი ამოცანაა დაწეროთ ვექტორ კლასის რეალიზაცია ჩანკ ლისტის საშუალებით, ანუ ვექტორის ელემენტები უნდა შეინახოთ ჩანკლისტის მეშვეობით. ჩანკებში მასივის ზომა მოცემული გაქვთ CHUNK\_SIZE კონსტანტის საშუალებით chunkvector.h ფაილში. ქვემოთ მოცემულია ვექტორ კლასის ინტერფეისის აღწერა:

ChunkVector();

კონსტრუქტორი - O(1)

~ChunkVector();

დესტრუქტორი, უნდა გაასუფთაოს დაკავებული მეხსიერება - O(n)

int size();

ფუნქცია აბრუნებს ვექტორში შენახული ელემენტების რაოდენობას - O(1)

int get(int index);

ფუნქცია აბრუნებს index ნომრის მქონე ელემენტს - O(n)

void set(int index, int value)

index ნომრის მქონე ელემენტს ანიჭებს value მნიშვნელობას - O(n)

void add(int value);

ამატებს ელემენტს სიის ბოლოში, თუ ბოლო ჩანკის მასივი სავსეა, ჩანკების სიის ბოლოში უნდა დაემატოს ახალი ცარიელი ჩანკი, სადაც უნდა გადაკოპირდეს წინა ჩანკის ელემენტების ნახევარი, შემდეგ კი ბოლოში დაემატოს value მნიშვნელობა. - O(n)

void remove(int index);

ფუნქციამ უნდა წაშალოს index ნომრის მქონე ელემენტი, თუ ელემენტის წაშლის შემდეგ დაგვრჩება ცარიელი ჩანკი, ფუქნციამ უნდა წაშალოს არსებული ჩანკი, წინააღმდეგ შემთხვევაში უნდა მოხდეს მხოლოდ ამ ჩანკის ელემენტების გადაწევა ერთით მარცხნივ. - O(n)

int getChunkNumber();

აბრუნებს ამ მომენტში არსებული ჩანკების რაოდენობას. - O(1)

void insert(int index, int value); - O(n)

ფუნქციის ზუსტი განსაზღვრისთვის შემოვიტანოთ აღვნიშნოთ k სიმბოლოთი იმ ჩანკის ნომერი სადაც არის index ნომრის მქონე ელემენტი

1. თუ k ნომრის მქონე ჩანკში არის თავისუფალი ადგილი, ფუნქციამ ყველა ელემნტი k ჩანკში რომელთა ნომერიც მეტია ან ტოლია index-ზე უნდა გადააჩოჩოს ერთი უჯრით მარჯვნივ და შემდეგ გამონთავისუფლებულ უჯრაში ჩაწეროს value მნიშვნელობა.
2. თუ k ნომრის მქონე ჩანკი გადავსებულია, ფუნქციამ ამ ჩანკის შემდეგ უნდა ჩაამატოს ახალი ჩანკი, სადაც უნდა გადაიტანოს k ჩანკის ელემენტების ნახევარი. ამის შემდეგ უნდა მოხდეს ახალი ელემენტის ჩამატება ზუსტად ისე როგორც 1. წესში არის განსაზღვრული.
3. თუ ვექტორში არ არსებობს index ნომრის მქონე ელემენტი, შესაბამისად არც ჩანკი(ასეთი რამ ხდება როცა index არის size()-ის ტოლი) ასეთ შემთხვევაში ფუქნციამ ახალი ელემენტი უნდა დაამატოს სიის ბოლოში, ზუსტად ისე როგორც add ფუნქციაში.

main.cpp ფაილში მოცემულია ტესტები, რომლებიც დაგეხმარებათ ზუსტად გაიაზროთ თუ როგორ უნდა მუშაობდეს თითოეული ფუნქცია.

გაითვალისწინეთ ვექტორის რეალიზაცია უნდა დაიწეროს ზუსტად ისე როგორც აღწერიალია ამოცანაში, წინააღმდეგ შემთხვევაში ამოცანა არ შეფასდება. თქვენ არ გაქვთ უკვე არსებული ქოლექშენ სტრუქტურების(Vector, Queue, Stack …) გამოყენების უფლება. არ გევალებათ შეცდომების დამუშავება, ჩათვალეთ რომ ტესტებში ფუნქციებს ყოველთვის ვალიდური მნიშვნელობები გადმოეცემა.

ChunkVector კლასის რეალიზაცია უდნა დაწეროთ chunkvector.h და chunkvector.cpp ფაილებში.