

ქვიზთან დაკავშირებით:

ჩატარდება ალბორატორიულ მეცადინეობაზე. როგორც ამოცანა, ასევე პროგრამის კოდი, თქვენს მიერ შეტანილი სიდიდეები და ეკრანზე გამოტანილი შედეგები უნდა გადმოიტანოთ ფურცელზე. ფურცლებს და ამოცანის პირობებს დაგირიგებთ ინსტრუქტორი. ტიპური ამოცანის პირობა ასეთ სტილშია:

შექმენით ფუნქცია, რომელიც `[first, last)` დიაპაზონში აკეთებს გარკვეულ საქმეს (დიაპაზონში მოთავსებული ობიექტების ჯამს, ან ნამრავლს, ან საშუალოს, ან ბეჭდავს მათ, ან კიდევ რაიმე ამნაირი). შინაარსის მიხედვით, მითითებული იქნება თუ რა უნდა დააბრუნოს ფუნქციამ.

1. შექმენით ვექტორი და შექმნის მომენტში მოათავსეთ მასში 6 რიცხვი. გამოიძახეთ თქვენი ფუნქცია მთლიანი კონტეინერისთვის. ამოწერეთ ეკრანზე გამოტანილი შედეგი.
2. STL ბიბლიოთეკის შესაბამისი ალგორითმის გამოყენებით მოძებნეთ კონტეინერში ... (სიდიდე, ან რაიმე თვისების მქონე/არმქონე ელემენტი). თუ ასეთი არსებობს, გამობეჭდეთ.
3. ჩაატარეთ გარკვეული მანიპულაცია მთლიან კონტეინერზე (მაქსიმალურის იტერატორის განსაზღვრა, ან ორი ტოლი მეზობლების მოძებნა, ან გადანაწილება) STL ბიბლიოთეკის შესაბამისი ალგორითმის გამოყენებით და გამობეჭდეთ შედეგები.

დავალების დასრულების შემდეგ, ფურცელზე ამოწერეთ თქვენს მიერ გამოყენებული ალგორითმების კოდები და უპასუხეთ დამატებით კითხვას: დავალების რა ნაწილი შესრულდებოდა ვექტორის ნაცვლად სია რომ შეგვექმნა?

მაგალითად, თუ თქვენ გინდათ მოძებნოთ გარკვეული თვისების არმქონე ელემენტის იტერატორი, გამოიძახებთ უშუალოდ `find_if_not` ალგორითმს და გადააწვდით მას კონტეინერის თავს, ბოლოს და პრედიკატს. მხოლოდ პროგრამის გამართვის შემდეგ მოგეთხოვებათ ამ ალგორითმის კოდის ამოწერა. სხვა სიტყვებით, თუ ალგორითმების დანიშნულება და გამოყენება იცით, მაგრამ რაიმე მიზეზის გამო ვერ წერთ მის კოდს, თქვენ მიიღებთ ქულებს შესრულებული სამუშაოსთვის.

საშუალოდ გამოცდასთან, ანუ კოლოკვიუმთან დაკავშირებით:

ყოველ ბილეთში იქნება რამდენიმე კითხვა ქვემოთ მოყვანილებიდან. კოლოკვიუმის მასალაში არ შედის გროვის აგების სისწრაფე, - გამოცდისთვის ვაპირებთ მის გამოყენებას.

1. რას ნიშნავს მონაცემთა აბსტრაგირება, ანუ მონაცემთა აბსტრაქტული ტიპი(ADT)?
2. რას ნიშნავს მონაცემთა სტრუქტურების განზოგადებათა ჯგუფები (მონათესავე ჯგუფები, families of abstractions). STL -ში ასეთი ჯგუფის მაგალითები.
3. ახსენით, კონტეინერის ელემენტზე წვდომის დრო როგორ არის დამოკიდებული მუხსიერებაში კონტეინერის განთავსების მეთოდზე. მოიყვანეთ მაგალითები.

4. რატომ ვუწოდებთ `bind2nd` -ს ფუნქციის ადაპტერს? მოიყვანეთ მისი გამოყენების მაგალითები.
5. რატომ ვუწოდებთ `not1`-ს ფუნქციის ადაპტერს? მოიყვანეთ მისი გამოყენების მაგალითები.
6. რა არის იტერატორი? რაში მდგომარეობს მისი გამოყენების აუცილებლობა და მოსახერხებლობა?
7. ვთქვათ, რომელიღაც ალგორითმის იმპლემენტაციაში არგუმენტად გადაცემულ `it` იტერატორზე სრულდება შემდეგი მოქმედებები: `it++; *it != a; it == b;`
დაასახელეთ იტერატორების ყველაზე ვრცელი კატეგორია, რომელსაც ეკუთვნის `it`.
8. ვთქვათ, რომელიღაც ალგორითმის იმპლემენტაციაში არგუმენტად გადაცემულ იტერატორზე სრულდება შემდეგი მოქმედებები: `it++; *it = a;` დაასახელეთ იტერატორების ყველაზე ვრცელი კატეგორია, რომელსაც ეკუთვნის `it`.
9. ვთქვათ, რომელიღაც ალგორითმის იმპლემენტაციაში არგუმენტად გადაცემულ იტერატორზე სრულდება შემდეგი მოქმედებები: `it++; *it = a; it == b;`
დაასახელეთ იტერატორების ყველაზე ვრცელი კატეგორია, რომელსაც ეკუთვნის `it`.
10. ვთქვათ, რომელიღაც ალგორითმის იმპლემენტაციაში არგუმენტად გადაცემულ იტერატორზე სრულდება შემდეგი მოქმედებები: `it++; it--; *it = a; it == b;`
დაასახელეთ იტერატორების ყველაზე ვრცელი კატეგორია, რომელსაც ეკუთვნის `it`.
11. ვთქვათ, რომელიღაც ალგორითმის იმპლემენტაციაში არგუმენტად გადაცემულ იტერატორზე სრულდება შემდეგი მოქმედებები: `it++; *(it-11); it < b;`
დაასახელეთ იტერატორების ყველაზე ვრცელი კატეგორია, რომელსაც ეკუთვნის `it`.
12. როგორ განიმარტება იტერატორების `[first,last)` დიაპაზონი (`range`)? რა თვისებები უნდა ჰქონდეს ამ წყვილს, რომ დიაპაზონი კორექტულად იყოს განმარტებული?
13. შეგვიძლია თუ არა C++ ენის ბიბლიოთეკის ალგორითმის გამოყენებით მონაცემების კოპირება დიაპაზონიდან ფაილში? ეკრანზე? მოიყვანეთ მაგალითი.
14. რა შემთხვევაში ხდება იტერატორების გაუქმება? მოიყვანეთ მაგალითი. რითი განსხვავდება იტერატორის გაუქმება სიაში და ვექტორში?
15. რატომ ეწოდება სტეკს კონტეინერის ადაპტერი? რომელი კონტეინერის ადაპტირება შეუძლია მას? მოიყვანეთ მაგალითები.
16. რატომ ეწოდება რიგს კონტეინერის ადაპტერი? რომელი კონტეინერის ადაპტირება შეუძლია მას? მოიყვანეთ მაგალითები.
17. რა არის პრიორიტეტების რიგი? რომელი კონტეინერის ადაპტირება შეუძლია მას? მოიყვანეთ მაგალითები.
18. აღწერეთ სტეკის რამდენიმე მეთოდი. მოიყვანეთ მაგალითები.
19. აღწერეთ რიგის რამდენიმე მეთოდი. მოიყვანეთ მაგალითები.

20. აღწერეთ პრიორიტეტების რიგის რამდენიმე მეთოდი. მოიყვანეთ მაგალითები.
21. ზოგადად, რა შემთხვევაში ვამბობთ, რომ იტერატორების $[first, last)$ დიაპაზონი წარმოადგენს გროვას?
22. რა შემთხვევაში ვამბობთ, რომ იტერატორების $[first, last)$ დიაპაზონი წარმოადგენს ორობით გროვას? განმარტეთ PARENT, LEFT და RIGHT ფუნქციები და ახსენით მათი გეომეტრიული აზრი.
23. რას აკეთებს HEAPIFY(a, b, i) ალგორითმი, როგორია მისი სისწრაფე და რა პირობებში მუშაობს კორექტულად?
24. რას აკეთებს MAKE_HEAP(a, b) ალგორითმი და როგორია მისი სისწრაფე (დამტკიცების გარეშე)? მოიყვანეთ ალგორითმის ფსევდოკოდი ან კოდი.
25. რას აკეთებს POP_HEAP(a, b) ალგორითმი და როგორია მისი სისწრაფე (დამტკიცების გარეშე)? მოიყვანეთ ალგორითმის ფსევდოკოდი ან კოდი.
26. რას აკეთებს PUSH_HEAP(a, b) ალგორითმი და როგორია მისი სისწრაფე (დამტკიცების გარეშე)? მოიყვანეთ ალგორითმის ფსევდოკოდი ან კოდი.
27. რას აკეთებს SORT_HEAP(a, b) ალგორითმი და როგორია მისი სისწრაფე (დამტკიცების გარეშე)? მოიყვანეთ ალგორითმის ფსევდოკოდი ან კოდი.