

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #8

1 მინიმუმის პოვნა ციკლის გარეშე

დაწერეთ ფუნქცია, ციკლების გამოყენების გარეშე, რომელიც იპოვოს და დააბრუნებს მოცემული ვექტორში მინიმალურ ელემენტს. ამ ამოცანაში ნებისმიერი ტიპის ციკლის მაგივრად რეკურსია უნდა გამოიყენოთ.

```
int findMin(Vector<int>& vec, int startIndex = 0) {  
    if (startIndex == vec.size() - 1)  
        return vec[startIndex];  
  
    return min(vec[startIndex],  
               findMin(vec, startIndex + 1));  
}
```

2 თანხის შედგენა კუპიურებით

დაწერეთ ფუნქცია, რომელსაც გადაეცემა თანხა, რომლის შედგენაც გვინდა, და კუპიურების ვექტორი, რომლებითაც უნდა შევადგინოთ ეს თანხა. ფუნქციამ უნდა გვიპასუხოს შესაძლებელია თუ არა ამ თანხის ამ კუპიურებით შედგენა. თითოეული კუპიურა შეგვიძლია ბევრჯერ გამოვიყენოთ.

```
bool canComposeWith(int value, Vector<int>& units) {  
    if (0 == value)  
        return true;  
    if (0 > value)  
        return false;  
  
    bool result = false;  
    for (int i = 0; i < units.size(); i++) {  
        result = result || canComposeWith(value - units[i], units)  
    }  
  
    return result;  
}
```

3 ქვესიმრავლის ჯამის ამოცანა

დაწერეთ ფუნქცია, რომელიც მიიღებს მთელი რიცხვების (int-ების) ვექტორს და სამიზნე რიცხვს დაადგენს ამ ვექტორიდან რომელიმე ქვესიმრავლე თუ შეგვიძლია რომ ამოვირჩიოთ, ისე რომ ამ ქვესიმრავლეში შემავალი რიცხვების ჯამი ზუსტად სამიზნე რიცხვის ტოლი იყოს.

```
bool subsetSum(int target, Vector<int>& set, int startIndex = 0) {  
    if (0 == target)  
        return true;  
  
    if (startIndex == set.size())  
        return false;  
  
    return (subsetSum(target - set[startIndex], set, startIndex + 1)  
            || subsetSum(target, set, startIndex + 1));  
}
```