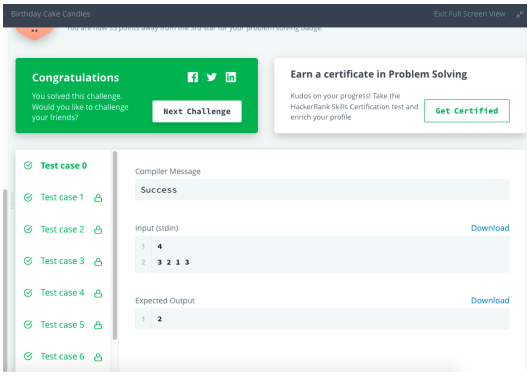
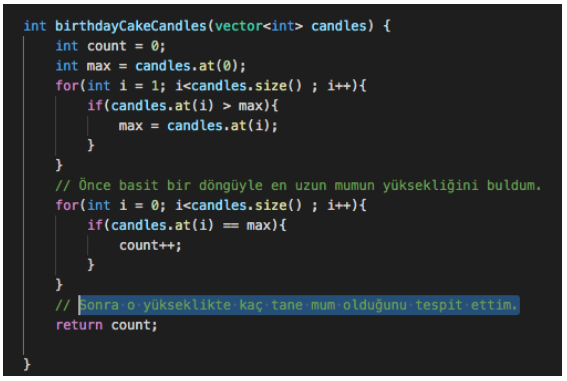
Unicrow Stajyerlik Başvuru Soruları

Seyfullah Sait ŞAHİN

Birthday Cake Candles(Temel Seviye): Test Cases 9/9

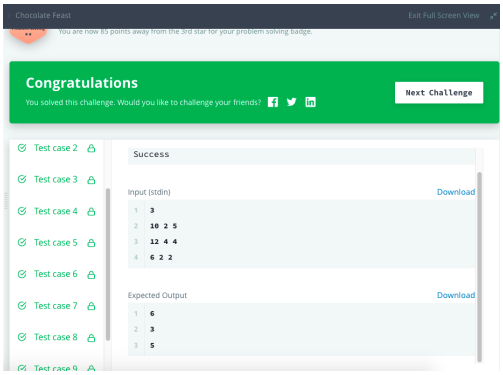


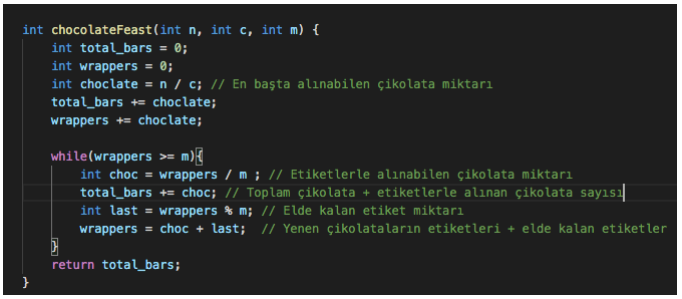


```cpp
int birthdayCakeCandles(vector<int> candles) {
    int count = 0;
    int max = candles.at(0);
    for(int i = 1; i<candles.size() ; i++){
        if(candles.at(i) > max){
            max = candles.at(i);
        }
    }
    // Önce basit bir döngüyle en uzun mumun yüksekliğini buldum.
    for(int i = 0; i<candles.size() ; i++){
        if(candles.at(i) == max){
            count++;
        }
    }
    // Sonra o yükseklikte kaç tane mum olduğunu tespit ettim.
    return count;
}
```

Önce basit bir döngüyle en uzun mumun yüksekliğini buldum. Sonra o yükseklikte kaç tane mum olduğunu tespit ettim.

Chocolate Feast (Temel Seviye): Test Cases 11/11

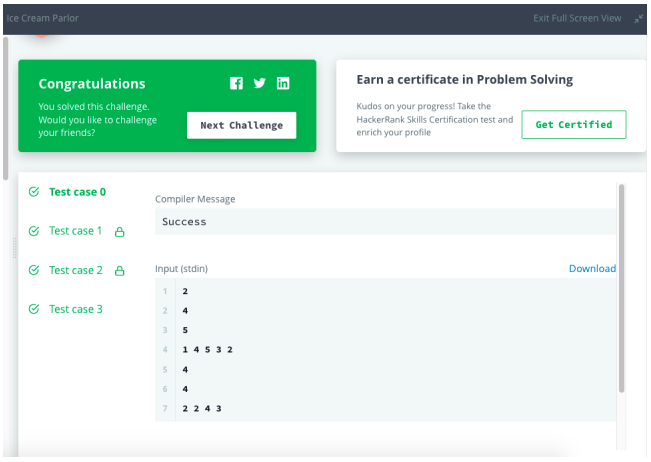


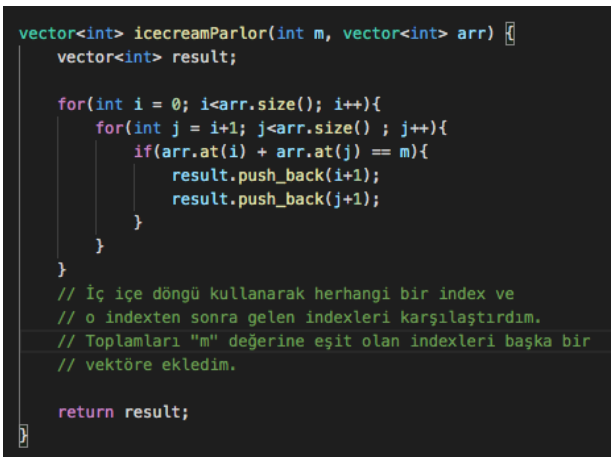


```cpp
int chocolateFeast(int n, int c, int m) {
    int total_bars = 0;
    int wrappers = 0;
    int choclate = n / c; // En başta alınabilen çikolata miktarı
    total_bars += choclate;
    wrappers += choclate;

    while(wrappers >= m){
        int choc = wrappers / m ; // Etiketlerle alınabilen çikolata miktarı
        total_bars += choc; // Toplam çikolata + etiketlerle alınan çikolata sayısı
        int last = wrappers % m; // Elde kalan etiket miktarı
        wrappers = choc + last;  // Yenen çikolataların etiketleri + elde kalan etiketler
    }
    return total_bars;
}
```

Öncelikle ilk parayla ne kadar çikolata alınabildiğini ve ne kadar etiket kaldığını buldum. Etiket sayısının, m değerinden küçük olmadığı durumları kontrol ettim ve her seferinde yeni çikolata sayısı ile etiket sayısını arttırdım.

Ice Cream Parlor (Temel Seviye): Test Cases 4/4

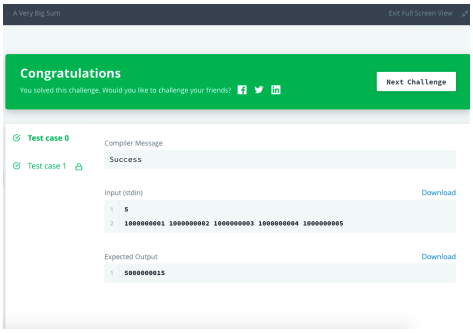


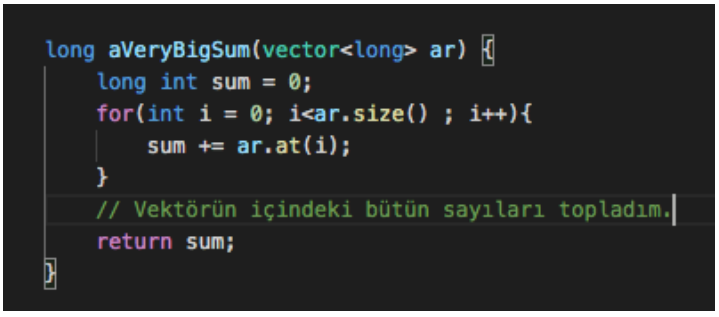


```cpp
vector<int> icecreamParlor(int m, vector<int> arr) {
    vector<int> result;

    for(int i = 0; i<arr.size(); i++){
        for(int j = i+1; j<arr.size() ; j++){
            if(arr.at(i) + arr.at(j) == m){
                result.push_back(i+1);
                result.push_back(j+1);
            }
        }
    }
    // İç içe döngü kullanarak herhangi bir index ve
    // o indexten sonra gelen indexleri karşılaştırdım.
    // Toplamları "m" değerine eşit olan indexleri başka bir
    // vektöre ekledim.

    return result;
}
```

2 döngü kullanarak bir indexteki sayıyı, ondan sonra gelen indexteki sayılarla karşılaştırdım ve koşulu sağlayanları yeni bir vektöre ekledim.

A Very Big Sum (Temel Seviye): Test Cases 2/2

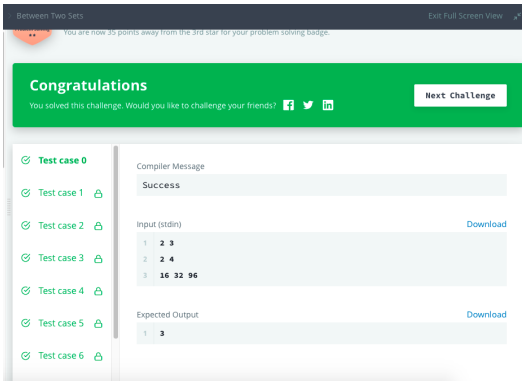




```cpp
long aVeryBigSum(vector<long> ar) {
    long int sum = 0;
    for(int i = 0; i<ar.size() ; i++){
        sum += ar.at(i);
    }
    // Vektörün içindeki bütün sayıları topladım.
    return sum;
}
```

Basit bir döngüyle vektör içindeki sayılar toplandı.

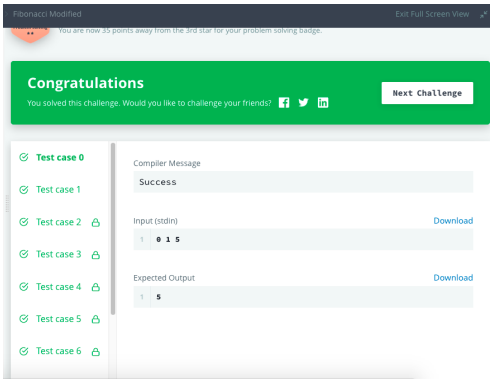
Between Two Sets (Temel Seviye) : Test Cases 9/9




```cpp
int getTotalX(vector<int> a, vector<int> b) {
    int count = 0;
    for(int k = 1; k <= 100 ; k++){
        // Bulmamız gereken sayılar 1 ile 100 arasında olduğu için hepsini deneyebiliriz
        bool factor = true;
        for(int i = 0 ; i < a.size(); i++){
            if(x % a.at(i) != 0)
                factor = false;
            // Bizden istenen sayının, array içindeki sayıya tam bölünebilmesinin kontrolü
        }
        for(int j = 0 ; j < b.size(); j++){
            if(b.at(j) % x != 0)
                factor = false;
            //  Array içindeki sayının, bizden istenen sayıya tam bölünebilmesinin kontrolü
        }
        if(factor) // Eğer "factor" değişkeni hala "true" ise count değeri 1 arttırılır.
            count++;
    }
    return count;
}
```

Aralık 1-100 arasında olduğu için bu aralıktaki her sayının ilk dizideki sayılara tam bölünebildiği ve ikinci dizideki sayıları tam böldüğü durumlar kontrol edildi.

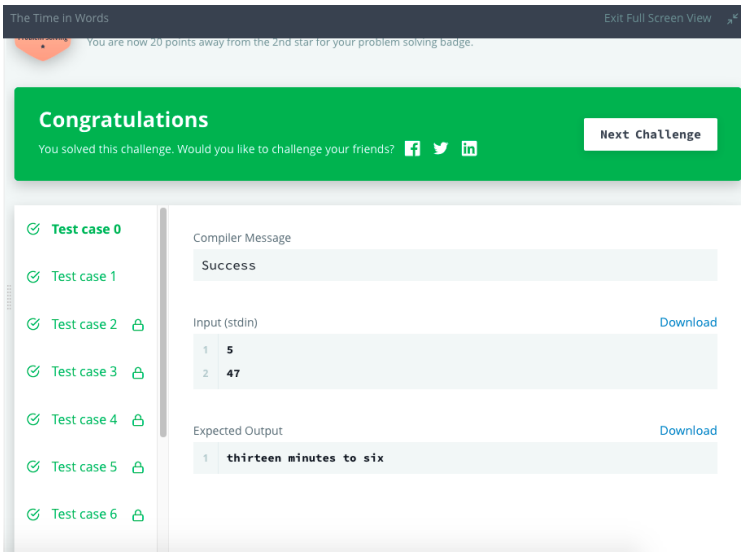
Fibonacci Modified (Orta Seviye): Test Cases 10/10




```python
def fibonacciModified(t1, t2, n):
    for i in range(2, n):
        Result = t1 + t2*t2
        t1 = t2
        t2 = Result
    # Basit bir döngü kullandım.
    # Büyük sayılarla C++ kodunda sıkıntı çıkınca
    # bu kodu Python ile yazmak durumunda kaldım.
    return Result
```

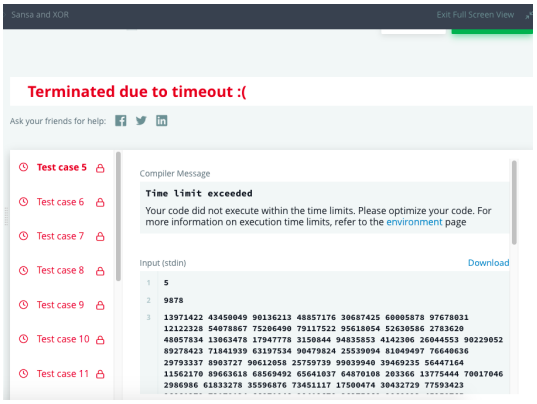
C++'ta büyük sayılarla çalışmak zor olduğu için bu sorunun çözümünü Python ile yaptım.

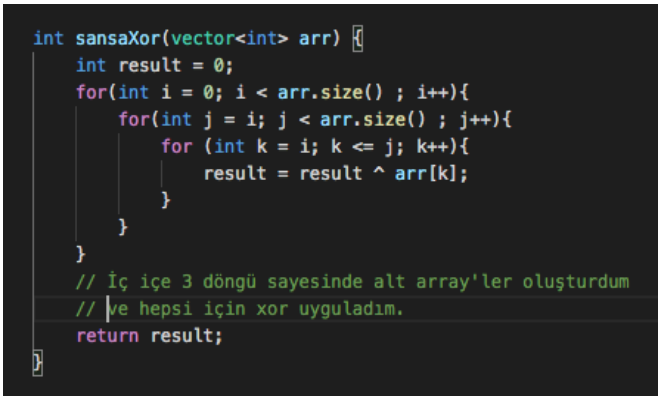
The Time In Words (Orta Seviye): Test Cases 11/11



Kullanmam gereken stringler için bir array oluşturdum ve kontrolleri sağlayarak bu stringleri kullandım.

Sansa And XOR (Orta Seviye): Test Cases 6/14





```
int sansaXor(vector<int> arr) {
    int result = 0;
    for(int i = 0; i < arr.size() ; i++){
        for(int j = i; j < arr.size() ; j++){
            for (int k = i; k <= j; k++){
                result = result ^ arr[k];
            }
        }
    }
    // İç içe 3 döngü sayesinde alt array'ler oluşturdum
    // ve hepsi için xor uyguladım.
    return result;
}
```

İç içe 3 döngü kullanarak bütün alt array'leri oluşturdum ve hepsine XOR uyguladım. Time Complexity'si fazla olduğu için 8 tane case'i geçemedim.